Tempo2

Generated by Doxygen 1.8.9.1

Thu Sep 17 2015 09:22:16

Contents

1 Main Page 1

2 Git INSTALLATION README 3

3 User Guide 7

4 Core Developers 9

5 Developer Guide 11

5.1 Tempo2 Developer Guide . . . 11

5.1.1 About this guide . . . 11

5.1.2 General code guidelines . . . 11

5.1.3 Development workflow . . . 11

5.1.4 Coding style . . . 12

6 Directory structure 15

7 Todo List 17

8 Module Index 19

8.1 Modules . . . 19

9 Class Index 21

9.1 Class List . . . 21

10 File Index 23

10.1 File List . . . 23

11 Module Documentation 25

11.1 libt2toolkit API . . . 25

11.1.1 Detailed Description . . . 25

11.2 libtempo2 External API . . . 26

11.2.1 Detailed Description . . . 26

12 Class Documentation 27

12.1 Cheby2D Struct Reference . . . 27

12.1.1 Member Data Documentation 27
12.1.1.1 coeff 27
12.1.1.2 nx 27
12.1.1.3 ny 27
12.2 ChebyModel Struct Reference 27
12.2.1 Member Data Documentation 28
12.2.1.1 cheby 28
12.2.1.2 dispersion_constant 28
12.2.1.3 freq_end 28
12.2.1.4 freq_start 28
12.2.1.5 frequency_cheby 28
12.2.1.6 mjd_end 28
12.2.1.7 mjd_start 28
12.2.1.8 psrname 28
12.2.1.9 sitename 28
12.3 ChebyModelSet Struct Reference 29
12.3.1 Member Data Documentation 29
12.3.1.1 nsegments 29
12.3.1.2 segments 29
12.4 clock_correction Struct Reference 29
12.4.1 Detailed Description 30
12.4.2 Member Data Documentation 30
12.4.2.1 correction 30
12.4.2.2 corrects_to 30
12.5 complexVal Struct Reference 30
12.5.1 Member Data Documentation 30
12.5.1.1 imag 30
12.5.1.2 real 30
12.6 DynamicArray Struct Reference 30
12.6.1 Member Data Documentation 30
12.6.1.1 data 30
12.6.1.2 elem_size 30
12.6.1.3 nalloced 31
12.6.1.4 nelem 31
12.7 FitInfo Struct Reference 31
12.7.1 Detailed Description 31
12.7.2 Member Data Documentation 32
12.7.2.1 constraintCounters 32
12.7.2.2 constraintDerivs 32
12.7.2.3 constraintIndex 32

12.7.2.4 nConstraints 32
12.7.2.5 nParams 32
12.7.2.6 paramCounters 32
12.7.2.7 paramDerivs 32
12.7.2.8 paramIndex 32
12.7.2.9 updateFunctions 32
12.8 gwgeneralSrc Struct Reference 32
12.8.1 Member Data Documentation 33
12.8.1.1 across_g 33
12.8.1.2 across_im_g 33
12.8.1.3 aplus_g 33
12.8.1.4 aplus_im_g 33
12.8.1.5 asl_g 33
12.8.1.6 asl_im_g 33
12.8.1.7 ast_g 33
12.8.1.8 ast_im_g 33
12.8.1.9 avx_g 33
12.8.1.10 avx_im_g 33
12.8.1.11 avy_g 33
12.8.1.12 avy_im_g 33
12.8.1.13 dist_bin 33
12.8.1.14 h 33
12.8.1.15 h_im 33
12.8.1.16 inc_bin 33
12.8.1.17 kg 33
12.8.1.18 omega_g 33
12.8.1.19 phase_g 33
12.8.1.20 phi_bin 33
12.8.1.21 phi_g 33
12.8.1.22 phi_polar_g 33
12.8.1.23 theta_bin 33
12.8.1.24 theta_g 33
12.9 gwgenSpec Struct Reference 34
12.9.1 Member Data Documentation 34
12.9.1.1 sl_alpha 34
12.9.1.2 sl_amp 34
12.9.1.3 st_alpha 34
12.9.1.4 st_amp 34
12.9.1.5 tensor_alpha 34
12.9.1.6 tensor_amp 34

12.9.1.7 vl_alpha 34
12.9.1.8 vl_amp 34
12.10gwSrc Struct Reference 34
12.10.1 Member Data Documentation 35
12.10.1.1 across_g 35
12.10.1.2 across_im_g 35
12.10.1.3 aplus_g 35
12.10.1.4 aplus_im_g 35
12.10.1.5 dist_bin 35
12.10.1.6 h 35
12.10.1.7 h_im 35
12.10.1.8 inc_bin 35
12.10.1.9 kg 35
12.10.1.10omega_g 35
12.10.1.11phase_g 35
12.10.1.12phi_bin 35
12.10.1.13phi_g 35
12.10.1.14phi_polar_g 35
12.10.1.15theta_bin 35
12.10.1.16theta_g 35
12.11interpolation_info Struct Reference 35
12.11.1 Member Data Documentation 36
12.11.1.1 np 36
12.11.1.2 nv 36
12.11.1.3 pc 36
12.11.1.4 twot 36
12.11.1.5 vc 36
12.12jpl_eph_data Struct Reference 36
12.12.1 Member Data Documentation 36
12.12.1.1 au 36
12.12.1.2 cache 36
12.12.1.3 curr_cache_loc 37
12.12.1.4 emrat 37
12.12.1.5 ephem_end 37
12.12.1.6 ephem_start 37
12.12.1.7 ephem_step 37
12.12.1.8 ephemeris_version 37
12.12.1.9 ifile 37
12.12.1.10iinfo 37
12.12.1.11ipt 37

12.12.1.12kernel_size 37
12.12.1.13ncoeff 37
12.12.1.14ncon 37
12.12.1.15pvsun 37
12.12.1.16recsize 37
12.12.1.17swap_bytes 37
12.13observation Struct Reference 37
12.13.1 Detailed Description 39
12.13.2 Member Data Documentation 39
12.13.2.1 addedNoise 39
12.13.2.2 averagebat 39
12.13.2.3 averageerr 39
12.13.2.4 averageres 39
12.13.2.5 bat 39
12.13.2.6 batCorr 39
12.13.2.7 bbat 39
12.13.2.8 clockCorr 39
12.13.2.9 correctionsTT 40
12.13.2.10correctionTT_TB 40
12.13.2.11correctionTT_Teph 40
12.13.2.12correctionUT1 40
12.13.2.13delayCorr 40
12.13.2.14deleted 40
12.13.2.15earth_ssb 40
12.13.2.16earthMoonBary_earth 40
12.13.2.17earthMoonBary_ssb 40
12.13.2.18efac 40
12.13.2.19einsteinRate 40
12.13.2.20equad 40
12.13.2.21flagID 41
12.13.2.22flagVal 41
12.13.2.23fname 41
12.13.2.24freq 41
12.13.2.25freqSSB 41
12.13.2.26jump 41
12.13.2.27jupiter_earth 41
12.13.2.28nclock_correction 41
12.13.2.29neptune_earth 41
12.13.2.30nFlags 41
12.13.2.31nphase 41

12.13.2.32nutations 41
12.13.2.33observatory_earth 41
12.13.2.34obsNjump 41
12.13.2.35origErr 41
12.13.2.36origsat 42
12.13.2.37pet 42
12.13.2.38phase 42
12.13.2.39phaseOffset 42
12.13.2.40planet_ssb 42
12.13.2.41prefitResidual 42
12.13.2.42psrPos 42
12.13.2.43pulseN 42
12.13.2.44residual 42
12.13.2.45roemer 42
12.13.2.46sat 42
12.13.2.47sat_day 42
12.13.2.48sat_sec 42
12.13.2.49saturn_earth 42
12.13.2.50shapiroDelayJupiter 42
12.13.2.51shapiroDelayNeptune 43
12.13.2.52shapiroDelaySaturn 43
12.13.2.53shapiroDelaySun 43
12.13.2.54shapiroDelayUranus 43
12.13.2.55shapiroDelayVenus 43
12.13.2.56shklovskii 43
12.13.2.57siteVel 43
12.13.2.58sun_earth 43
12.13.2.59sun_ssb 43
12.13.2.60tdis1 43
12.13.2.61tdis2 43
12.13.2.62telID 43
12.13.2.63TNDMErr 44
12.13.2.64TNDMSignal 44
12.13.2.65TNGroupErr 44
12.13.2.66TNGroupSignal 44
12.13.2.67TNRedErr 44
12.13.2.68TNRedSignal 44
12.13.2.69toaDMErr 44
12.13.2.70toaErr 44
12.13.2.71torb 44

12.13.2.72troposphericDelay 44
12.13.2.73uranus_earth 44
12.13.2.74venus_earth 44
12.13.2.75zenith 45
12.14observatory Struct Reference 45
12.14.1 Member Data Documentation 45
12.14.1.1 clock_name 45
12.14.1.2 code 45
12.14.1.3 height_grs80 45
12.14.1.4 latitude_grs80 45
12.14.1.5 longitude_grs80 45
12.14.1.6 name 45
12.14.1.7 x 45
12.14.1.8 y 45
12.14.1.9 z 45
12.15parameter Struct Reference 45
12.15.1 Detailed Description 46
12.15.2 Member Data Documentation 46
12.15.2.1 aSize 46
12.15.2.2 err 46
12.15.2.3 fitFlag 46
12.15.2.4 label 46
12.15.2.5 linkFrom 46
12.15.2.6 linkTo 46
12.15.2.7 nLinkFrom 46
12.15.2.8 nLinkTo 46
12.15.2.9 paramSet 46
12.15.2.10prefit 47
12.15.2.11prefitErr 47
12.15.2.12shortlabel 47
12.15.2.13val 47
12.16pulsar Struct Reference 47
12.16.1 Detailed Description 53
12.16.2 Member Data Documentation 53
12.16.2.1 addTNGlobalEQ 53
12.16.2.2 auto_constraints 53
12.16.2.3 AverageEpochWidth 53
12.16.2.4 AverageFlag 53
12.16.2.5 AverageResiduals 53
12.16.2.6 binaryModel 53

12.16.2.7 bootStrap 53
12.16.2.8 calcShapiro 53
12.16.2.9 cgw_angpol 53
12.16.2.10 cgw_cosinc 53
12.16.2.11 cgw_h0 53
12.16.2.12 cgw_mc 53
12.16.2.13 clk_offsE 53
12.16.2.14 clk_offsT 54
12.16.2.15 clk_offsV 54
12.16.2.16 clkOffsN 54
12.16.2.17 clock 54
12.16.2.18 clockFromOverride 54
12.16.2.19 constraints 54
12.16.2.20 correctTroposphere 54
12.16.2.21 covar 54
12.16.2.22 decjStrPost 54
12.16.2.23 decjStrPre 54
12.16.2.24 decsim 54
12.16.2.25 deleteFileName 54
12.16.2.26 dilateFreq 54
12.16.2.27 dmoffsCM 54
12.16.2.28 dmoffsCM_error 54
12.16.2.29 dmoffsCM_mjd 54
12.16.2.30 dmoffsCM_weight 54
12.16.2.31 dmoffsCMnum 55
12.16.2.32 dmoffsDM 55
12.16.2.33 dmoffsDM_error 55
12.16.2.34 dmoffsDM_mjd 55
12.16.2.35 dmoffsDM_weight 55
12.16.2.36 dmoffsDMnum 55
12.16.2.37 dmOffset 55
12.16.2.38 eclCoord 55
12.16.2.39 eopc04_file 55
12.16.2.40 ephemeris 55
12.16.2.41 filterStr 55
12.16.2.42 fitChisq 55
12.16.2.43 fitFunc 55
12.16.2.44 fitinfo 55
12.16.2.45 fitJump 55
12.16.2.46 fitMode 55

12.16.2.47fitNfree 55

12.16.2.48fitParamGlobalI 55

12.16.2.49fitParamGlobalK 55

12.16.2.50fitParamI 56

12.16.2.51fitParamK 56

12.16.2.52fixedFormat 56

12.16.2.53fjumpID 56

12.16.2.54globalNfit 56

12.16.2.55globalNoConstrain 56

12.16.2.56gwb_decj 56

12.16.2.57gwb_epoch 56

12.16.2.58gwb_geom_c 56

12.16.2.59gwb_geom_p 56

12.16.2.60gwb_raj 56

12.16.2.61gwb_width 56

12.16.2.62gwecc_dec 56

12.16.2.63gwecc_distance 56

12.16.2.64gwecc_e 56

12.16.2.65gwecc_epoch 56

12.16.2.66gwecc_inc 56

12.16.2.67gwecc_m1 56

12.16.2.68gwecc_m2 56

12.16.2.69gwecc_nodes_orientation 56

12.16.2.70gwecc_orbital_period 56

12.16.2.71gwecc_psrdist 56

12.16.2.72gwecc_pulsarTermOn 56

12.16.2.73gwecc_ra 56

12.16.2.74gwecc_redshift 57

12.16.2.75gwecc_theta_0 57

12.16.2.76gwecc_theta_nodes 57

12.16.2.77gwm_decj 57

12.16.2.78gwm_dphase 57

12.16.2.79gwm_epoch 57

12.16.2.80gwm_phi 57

12.16.2.81gwm_raj 57

12.16.2.82gwsrc_across_i 57

12.16.2.83gwsrc_across_i_e 57

12.16.2.84gwsrc_across_r 57

12.16.2.85gwsrc_across_r_e 57

12.16.2.86gwsrc_aplus_i 57

12.16.2.87 gwsrc_aplus_i_e . . . 57
12.16.2.88 gwsrc_aplus_r . . . 57
12.16.2.89 gwsrc_aplus_r_e . . . 57
12.16.2.90 gwsrc_dec . . . 57
12.16.2.91 gwsrc_epoch . . . 57
12.16.2.92 gwsrc_psrdist . . . 57
12.16.2.93 gwsrc_ra . . . 57
12.16.2.94 ifunc_weights . . . 57
12.16.2.95 ifuncE . . . 57
12.16.2.96 ifuncN . . . 57
12.16.2.97 ifuncT . . . 57
12.16.2.98 ifuncV . . . 57
12.16.2.99 ipm . . . 57
12.16.2.100 jboFormat . . . 58
12.16.2.101 JPL_EPHEMERIS . . . 58
12.16.2.102 jumpStr . . . 58
12.16.2.103 jumpVal . . . 58
12.16.2.104 jumpValErr . . . 58
12.16.2.105 name . . . 58
12.16.2.106 nCompanion . . . 58
12.16.2.107 nconstraints . . . 58
12.16.2.108 nDMEvents . . . 58
12.16.2.109 ndmx . . . 58
12.16.2.110 ne_sw . . . 58
12.16.2.111 nFit . . . 58
12.16.2.112 nGlobal . . . 58
12.16.2.113 nits . . . 58
12.16.2.114 nJumps . . . 59
12.16.2.115 nobs . . . 59
12.16.2.116 noWarnings . . . 59
12.16.2.117 nParam . . . 59
12.16.2.118 nPhaseJump . . . 59
12.16.2.119 nQuad . . . 59
12.16.2.120 nStorePrecision . . . 59
12.16.2.121 nT2efac . . . 59
12.16.2.122 nT2equad . . . 59
12.16.2.123 nTelDX . . . 59
12.16.2.124 nTelDY . . . 59
12.16.2.125 nTelDZ . . . 59
12.16.2.126 nTNBandNoise . . . 59

12.16.2.127 TNECORR 59
12.16.2.128 TNEF 59
12.16.2.129 TNEQ 59
12.16.2.130 TNGroupNoise 59
12.16.2.131 TNShapeletEvents 59
12.16.2.132 TNSQ 59
12.16.2.133 Toffset 59
12.16.2.134 White 59
12.16.2.135 White_dm 59
12.16.2.136 obsn 60
12.16.2.137 offset 60
12.16.2.138 offset_e 60
12.16.2.139 outputTMatrix 60
12.16.2.140 param 60
12.16.2.141 passStr 60
12.16.2.142 phaseJump 60
12.16.2.143 phaseJumpDir 60
12.16.2.144 phaseJumpID 60
12.16.2.145 planetShapiro 60
12.16.2.146 posPulsar 60
12.16.2.147 quad_across_i 60
12.16.2.148 quad_across_i_e 60
12.16.2.149 quad_across_r 60
12.16.2.150 quad_across_r_e 60
12.16.2.151 quad_aplus_i 60
12.16.2.152 quad_aplus_i_e 61
12.16.2.153 quad_aplus_r 61
12.16.2.154 quad_aplus_r_e 61
12.16.2.155 quad_ifunc_c_DEC 61
12.16.2.156 quad_ifunc_c_RA 61
12.16.2.157 quad_ifunc_geom_c 61
12.16.2.158 quad_ifunc_geom_p 61
12.16.2.159 quad_ifunc_p_DEC 61
12.16.2.160 quad_ifunc_p_RA 61
12.16.2.161 quad_ifuncE_c 61
12.16.2.162 quad_ifuncE_p 61
12.16.2.163 quad_ifuncN_c 61
12.16.2.164 quad_ifuncN_p 61
12.16.2.165 quad_ifuncT_c 61
12.16.2.166 quad_ifuncT_p 61

12.16.2.167 quad_ifuncV_c . . . 61

12.16.2.168 quad_ifuncV_p . . . 61

12.16.2.169 quadDEC . . . 61

12.16.2.170 quadEpoch . . . 61

12.16.2.171 quadRA . . . 61

12.16.2.172 rajStrPost . . . 61

12.16.2.173 rajStrPre . . . 61

12.16.2.174 rasim . . . 61

12.16.2.175 rescaleErrChisq . . . 61

12.16.2.176 rmsPost . . . 61

12.16.2.177 rmsPre . . . 61

12.16.2.178 robust . . . 61

12.16.2.179 setTelVelX . . . 62

12.16.2.180 setTelVelY . . . 62

12.16.2.181 setTelVelZ . . . 62

12.16.2.182 setUnits . . . 62

12.16.2.183 simflag . . . 62

12.16.2.184 sorted . . . 62

12.16.2.185 storePrec . . . 62

12.16.2.186 swm . . . 62

12.16.2.187 t2cMethod . . . 62

12.16.2.188 T2efacFlagID . . . 62

12.16.2.189 T2efacFlagVal . . . 62

12.16.2.190 T2efacVal . . . 62

12.16.2.191 T2equadFlagID . . . 62

12.16.2.192 T2equadFlagVal . . . 62

12.16.2.193 T2equadVal . . . 62

12.16.2.194 T2globalEfac . . . 62

12.16.2.195 telDX_e . . . 62

12.16.2.196 telDX_t . . . 62

12.16.2.197 telDX_v . . . 62

12.16.2.198 telDX_vel . . . 62

12.16.2.199 telDX_vel_e . . . 62

12.16.2.200 telDY_e . . . 62

12.16.2.201 telDY_t . . . 63

12.16.2.202 telDY_v . . . 63

12.16.2.203 telDY_vel . . . 63

12.16.2.204 telDY_vel_e . . . 63

12.16.2.205 telDZ_e . . . 63

12.16.2.206 telDZ_t . . . 63

12.16.2.207 telDZ_v . . . 63
12.16.2.208 telDZ_vel . . . 63
12.16.2.209 telDZ_vel_e . . . 63
12.16.2.210 tempo1 . . . 63
12.16.2.211 timeEphemeris . . . 63
12.16.2.212 TNBandDMAmp . . . 63
12.16.2.213 TNBandDMC . . . 63
12.16.2.214 TNBandDMGam . . . 63
12.16.2.215 TNBandNoiseAmp . . . 63
12.16.2.216 TNBandNoiseC . . . 63
12.16.2.217 TNBandNoiseGam . . . 63
12.16.2.218 TNBandNoiseHF . . . 63
12.16.2.219 TNBandNoiseLF . . . 63
12.16.2.220 TNDMAmp . . . 63
12.16.2.221 TNDMC . . . 63
12.16.2.222 TNDMCoeffs . . . 63
12.16.2.223 TNDMEvAmp . . . 63
12.16.2.224 TNDMEvGam . . . 63
12.16.2.225 TNDMEvLength . . . 63
12.16.2.226 TNDMEvLin . . . 63
12.16.2.227 TNDMEvOff . . . 64
12.16.2.228 TNDMEvQuad . . . 64
12.16.2.229 TNDMEvStart . . . 64
12.16.2.230 TNDMGam . . . 64
12.16.2.231 TNECORRFlagID . . . 64
12.16.2.232 TNECORRFlagVal . . . 64
12.16.2.233 TNECORRVal . . . 64
12.16.2.234 TNEFFlagID . . . 64
12.16.2.235 TNEFFlagVal . . . 64
12.16.2.236 TNEFVal . . . 64
12.16.2.237 TNEQFlagID . . . 64
12.16.2.238 TNEQFlagVal . . . 64
12.16.2.239 TNEQVal . . . 64
12.16.2.240 TNGlobalEF . . . 64
12.16.2.241 TNGlobalEQ . . . 64
12.16.2.242 TNGroupNoiseAmp . . . 64
12.16.2.243 TNGroupNoiseC . . . 64
12.16.2.244 TNGroupNoiseFlagID . . . 64
12.16.2.245 TNGroupNoiseFlagVal . . . 64
12.16.2.246 TNGroupNoiseGam . . . 64

12.16.2.247 TNRedAmp . . . 64
12.16.2.248 TNRedC . . . 64
12.16.2.249 TNRedCoeffs . . . 64
12.16.2.250 TNRedCorner . . . 64
12.16.2.251 TNRedFLow . . . 64
12.16.2.252 TNRedGam . . . 64
12.16.2.253 TNShapeletEvFScale . . . 64
12.16.2.254 TNShapeletEvN . . . 64
12.16.2.255 TNShapeletEvPos . . . 65
12.16.2.256 TNShapeletEvWidth . . . 65
12.16.2.257 TNSQFlagID . . . 65
12.16.2.258 TNSQFlagVal . . . 65
12.16.2.259 TNSQVal . . . 65
12.16.2.260 TNsubtractDM . . . 65
12.16.2.261 TNsubtractRed . . . 65
12.16.2.262 ToAextraCovar . . . 65
12.16.2.263 tOffset . . . 65
12.16.2.264 tOffset_f1 . . . 65
12.16.2.265 tOffset_f2 . . . 65
12.16.2.266 tOffset_t1 . . . 65
12.16.2.267 tOffset_t2 . . . 65
12.16.2.268 tOffsetFlags . . . 65
12.16.2.269 tOffsetSite . . . 65
12.16.2.270 tzrsite . . . 65
12.16.2.271 units . . . 65
12.16.2.272 useCalceph . . . 65
12.16.2.273 useTNOrth . . . 65
12.16.2.274 velPulsar . . . 65
12.16.2.275 wave_cos . . . 65
12.16.2.276 wave_cos_dm . . . 65
12.16.2.277 wave_cos_dm_err . . . 66
12.16.2.278 wave_cos_err . . . 66
12.16.2.279 wave_sine . . . 66
12.16.2.280 wave_sine_dm . . . 66
12.16.2.281 wave_sine_dm_err . . . 66
12.16.2.282 wave_sine_err . . . 66
12.16.2.283 waveScale . . . 66
12.16.2.284 whiteNoiseModelFile . . . 66
12.17 storePrecision Struct Reference . . . 66
12.17.1 Member Data Documentation . . . 66

12.17.1.1 comment 66
12.17.1.2 minPrec 66
12.17.1.3 routine 66
12.18T1Polyco Struct Reference 66
12.18.1 Member Data Documentation 67
12.18.1.1 binary_frequency 67
12.18.1.2 binary_phase 67
12.18.1.3 coeff 67
12.18.1.4 date_string 67
12.18.1.5 dm 67
12.18.1.6 doppler 67
12.18.1.7 frequency_obs 67
12.18.1.8 frequency_psr_0 67
12.18.1.9 log10rms 67
12.18.1.10mjd_mid 67
12.18.1.11ncoeff 67
12.18.1.12psrname 67
12.18.1.13reference_phase 67
12.18.1.14sitename 67
12.18.1.15span 67
12.18.1.16utc_string 67
12.19T1PolycoSet Struct Reference 68
12.19.1 Member Data Documentation 68
12.19.1.1 nsegments 68
12.19.1.2 segments 68
12.20T2Predictor Struct Reference 68
12.20.1 Member Data Documentation 69
12.20.1.1 cheby 69
12.20.1.2 kind 69
12.20.1.3 modelset 69
12.20.1.4 t1 69
12.21TabulatedFunction Struct Reference 70
12.21.1 Member Data Documentation 70
12.21.1.1 fileName 70
12.21.1.2 header_line 70
12.21.1.3 samples 70
12.22TabulatedFunctionSample Struct Reference 70
12.22.1 Member Data Documentation 70
12.22.1.1 x 70
12.22.1.2 y 71

13 File Documentation 73
13.1 cholesky.h File Reference . . . 73
13.1.1 Function Documentation . . . 73
13.1.1.1 cholesky_covarFunc2matrix . . . 73
13.1.1.2 cholesky_dmModel . . . 73
13.1.1.3 cholesky_dmModelCovarParam . . . 73
13.1.1.4 cholesky_ecm . . . 73
13.1.1.5 cholesky_formUinv . . . 73
13.1.1.6 cholesky_powerlawModel . . . 73
13.1.1.7 cholesky_powerlawModel_withBeta . . . 73
13.1.1.8 cholesky_readFromCovarianceFunction . . . 74
13.2 choleskyRoutines.h File Reference . . . 74
13.2.1 Function Documentation . . . 75
13.2.1.1 T2calculateCholesky . . . 75
13.2.1.2 T2calculateCovarFunc . . . 75
13.2.1.3 T2calculateDailyCovariance . . . 75
13.2.1.4 T2calculateSpectra . . . 75
13.2.1.5 T2cholDecomposition . . . 75
13.2.1.6 T2cubicFit . . . 75
13.2.1.7 T2findSmoothCurve . . . 75
13.2.1.8 T2fitSpectra . . . 75
13.2.1.9 T2get_covFunc_automatic . . . 75
13.2.1.10 T2getHighFreqRes . . . 75
13.2.1.11 T2getWhiteNoiseLevel . . . 75
13.2.1.12 T2getWhiteRes . . . 75
13.2.1.13 T2guess_vals . . . 75
13.2.1.14 T2interpolate . . . 75
13.2.1.15 T2obtainTimingResiduals . . . 76
13.2.1.16 T2writeCovarFuncModel . . . 76
13.2.2 Variable Documentation . . . 76
13.2.2.1 EXPSMOOTH . . . 76
13.2.2.2 FCALPHA . . . 76
13.2.2.3 FCFINAL . . . 76
13.2.2.4 NFIT . . . 76
13.2.2.5 UPW . . . 76
13.2.2.6 WNLEVEL . . . 76
13.3 config.h File Reference . . . 76
13.3.1 Macro Definition Documentation . . . 77
13.3.1.1 _DARWIN_USE_64_BIT_INODE . . . 77
13.3.1.2 F77_FUNC . . . 77

13.3.1.3 F77_FUNC_ 77
13.3.1.4 HAVE_BLAS 77
13.3.1.5 HAVE_DLERROR 77
13.3.1.6 HAVE_DLFCN_H 77
13.3.1.7 HAVE_FFTW3 77
13.3.1.8 HAVE_INTTYPES_H 77
13.3.1.9 HAVE_LAPACK 77
13.3.1.10 HAVE_LIBDL 77
13.3.1.11 HAVE_LIBDLLOADER 77
13.3.1.12 HAVE_LIBM 77
13.3.1.13 HAVE_MEMORY_H 77
13.3.1.14 HAVE_PGPLOT 77
13.3.1.15 HAVE_PTHREAD 77
13.3.1.16 HAVE_STDINT_H 77
13.3.1.17 HAVE_STDLIB_H 77
13.3.1.18 HAVE_STRING_H 78
13.3.1.19 HAVE_STRINGS_H 78
13.3.1.20 HAVE_SYS_STAT_H 78
13.3.1.21 HAVE_SYS_TYPES_H 78
13.3.1.22 HAVE_UNISTD_H 78
13.3.1.23 LT_OBJDIR 78
13.3.1.24 PACKAGE 78
13.3.1.25 PACKAGE_BUGREPORT 78
13.3.1.26 PACKAGE_NAME 78
13.3.1.27 PACKAGE_STRING 78
13.3.1.28 PACKAGE_TARNAME 78
13.3.1.29 PACKAGE_URL 78
13.3.1.30 PACKAGE_VERSION 78
13.3.1.31 STDC_HEADERS 78
13.3.1.32 TEMPO2_ARCH 78
13.3.1.33 VERSION 78
13.3.1.34 X_DISPLAY_MISSING 78
13.4 constraints.h File Reference 78
13.4.1 Function Documentation 79
13.4.1.1 autosetDMCM 79
13.4.1.2 computeConstraintWeights 79
13.4.1.3 consFunc_dmmodel_cw 79
13.4.1.4 consFunc_dmmodel_cw_year 79
13.4.1.5 consFunc_dmmodel_dm1 80
13.4.1.6 consFunc_dmmodel_mean 80

13.4.1.7 consFunc_ifunc 80
13.4.1.8 consFunc_ifunc_year 80
13.4.1.9 consFunc_qifunc_c_year 80
13.4.1.10 consFunc_qifunc_p_year 80
13.4.1.11 consFunc_quad_ifunc_c 80
13.4.1.12 consFunc_quad_ifunc_p 80
13.4.1.13 consFunc_tel_dx 80
13.4.1.14 consFunc_tel_dy 80
13.4.1.15 consFunc_tel_dz 80
13.4.1.16 CONSTRAINTfuncs 80
13.4.1.17 get_constraint_name 80
13.4.1.18 standardConstraintFunctions 80
13.5 documentation/1_USER_GUIDE.md File Reference 80
13.6 documentation/2_developers.md File Reference 80
13.7 documentation/3_DEVELOPER_GUIDE.md File Reference 80
13.8 documentation/4_directories.md File Reference 80
13.9 dynarr.h File Reference 81
13.9.1 Function Documentation 81
13.9.1.1 DynamicArray_free 81
13.9.1.2 DynamicArray_init 81
13.9.1.3 DynamicArray_push_back 82
13.9.1.4 DynamicArray_resize 82
13.10GWsim.h File Reference 82
13.10.1 Typedef Documentation 83
13.10.1.1 gwgeneralSrc 83
13.10.1.2 gwgenSpec 83
13.10.1.3 gwSrc 83
13.10.2 Function Documentation 83
13.10.2.1 calculateResidualgeneralGW 83
13.10.2.2 calculateResidualGW 83
13.10.2.3 dadt 83
13.10.2.4 dedt 83
13.10.2.5 dotProduct 83
13.10.2.6 dtdt 83
13.10.2.7 eccRes 83
13.10.2.8 eccResWithEnergy 84
13.10.2.9 Fe 84
13.10.2.10Findphi 84
13.10.2.11GWanisotropicbackground 84
13.10.2.12GWbackground 84

13.10.2.13GWbackground_read 84
13.10.2.14GWbackground_write 84
13.10.2.15GWdipolebackground 84
13.10.2.16GWgeneralanisotropicbackground 84
13.10.2.17GWgeneralbackground 84
13.10.2.18GWgeneralbackground_read 84
13.10.2.19GWgeneralbackground_write 84
13.10.2.20matrixMult 84
13.10.2.21psrangle 84
13.10.2.22Rs 84
13.10.2.23setupgeneralGW 84
13.10.2.24setupGW 84
13.10.2.25setupPulsar_GWsim 84
13.10.2.26sphharm 84
13.11ifteph.h File Reference 84
13.11.1 Macro Definition Documentation 85
13.11.1.1 IFTE_JD0 85
13.11.1.2 IFTE_K 85
13.11.1.3 IFTE_KM1 85
13.11.1.4 IFTE_LC 86
13.11.1.5 IFTE_MJD0 86
13.11.1.6 IFTE_TEPH0 86
13.11.2 Function Documentation 86
13.11.2.1 IFTE_close_file 86
13.11.2.2 IFTE_DeltaT 86
13.11.2.3 IFTE_DeltaTDot 86
13.11.2.4 IFTE_get_DeltaT_DeltaTDot 86
13.11.2.5 IFTE_get_vE 86
13.11.2.6 IFTE_get_vE_vEDot 86
13.11.2.7 IFTE_get_vEDot 86
13.11.2.8 IFTE_init 86
13.12jpl_int.h File Reference 86
13.12.1 Macro Definition Documentation 86
13.12.1.1 JPL_HEADER_SIZE 86
13.12.1.2 MAX_KERNEL_SIZE 86
13.12.2 Typedef Documentation 86
13.12.2.1 JPLlong 86
13.13jpleph.h File Reference 86
13.13.1 Macro Definition Documentation 87
13.13.1.1 DLL_FUNC 87

13.13.1.2 JPL_EPHEM_AU_IN_KM 87
13.13.1.3 JPL_EPHEM_EARTH_MOON_RATIO 87
13.13.1.4 JPL_EPHEM_END_JD 87
13.13.1.5 JPL_EPHEM_EPHEMERIS_VERSION 87
13.13.1.6 JPL_EPHEM_KERNEL_NCOEFF 87
13.13.1.7 JPL_EPHEM_KERNEL_RECORD_SIZE 87
13.13.1.8 JPL_EPHEM_KERNEL_SIZE 87
13.13.1.9 JPL_EPHEM_KERNEL_SWAP_BYTES 87
13.13.1.10JPL_EPHEM_N_CONSTANTS 87
13.13.1.11JPL_EPHEM_START_JD 87
13.13.1.12JPL_EPHEM_STEP 87
13.13.2 Function Documentation 88
13.13.2.1 jpl_close_ephemeris 88
13.13.2.2 jpl_get_double 88
13.13.2.3 jpl_get_long 88
13.13.2.4 jpl_init_ephemeris 88
13.13.2.5 jpl_pleph 88
13.13.2.6 jpl_state 88
13.13.2.7 make_sub_ephem 88
13.14read_fortran.h File Reference 88
13.14.1 Function Documentation 89
13.14.1.1 close_file 89
13.14.1.2 open_file 89
13.14.1.3 read_char 89
13.14.1.4 read_character 89
13.14.1.5 read_double 89
13.14.1.6 read_float 89
13.14.1.7 read_int 89
13.14.1.8 read_record_int 89
13.14.2 Variable Documentation 89
13.14.2.1 c_fileptr 89
13.14.2.2 swapByte 89
13.15read_fortran2.h File Reference 89
13.15.1 Function Documentation 90
13.15.1.1 close_file2 90
13.15.1.2 open_file2 90
13.15.1.3 read_character2 90
13.15.1.4 read_double2 90
13.15.1.5 read_float2 90
13.15.1.6 read_int2 90

13.15.1.7 read_record_int2 90
13.15.2 Variable Documentation 90
13.15.2.1 c_fileptr2 90
13.15.2.2 swapByte2 90
13.16README.md File Reference 90
13.17T2accel.h File Reference 90
13.17.1 Macro Definition Documentation 91
13.17.1.1 ACCEL_LSQ 91
13.17.1.2 ACCEL_MULTMATRIX 91
13.17.1.3 ACCEL_UINV 91
13.17.2 Function Documentation 91
13.17.2.1 accel_lsq_qr 91
13.17.2.2 accel_multMatrix 91
13.17.2.3 accel_multMatrixVec 91
13.17.2.4 accel_uinv 91
13.17.3 Variable Documentation 91
13.17.3.1 useT2accel 91
13.18t2fit.h File Reference 91
13.18.1 Function Documentation 92
13.18.1.1 t2Fit 92
13.18.1.2 t2Fit_buildConstraintsMatrix 92
13.18.1.3 t2Fit_buildDesignMatrix 92
13.18.1.4 t2Fit_fillFitInfo 92
13.18.1.5 t2Fit_fillGlobalFitInfo 92
13.18.1.6 t2Fit_getFitData 92
13.18.1.7 t2Fit_updateParameters 92
13.19t2fit_dmmodel.h File Reference 92
13.19.1 Function Documentation 93
13.19.1.1 t2FitFunc_dmmodelCM 93
13.19.1.2 t2FitFunc_dmmodelDM 93
13.19.1.3 t2UpdateFunc_dmmodelCM 94
13.19.1.4 t2UpdateFunc_dmmodelDM 94
13.20t2fit_fitwaves.h File Reference 94
13.20.1 Function Documentation 94
13.20.1.1 t2FitFunc_fitwaves 94
13.20.1.2 t2UpdateFunc_fitwaves 94
13.21t2fit_glitch.h File Reference 94
13.21.1 Function Documentation 95
13.21.1.1 t2FitFunc_stdGlitch 95
13.21.1.2 t2UpdateFunc_stdGlitch 95

13.22t2fit_ifunc.h File Reference 95
13.22.1 Function Documentation 96
13.22.1.1 ifunc 96
13.22.1.2 sinfunc 96
13.22.1.3 t2FitFunc_ifunc 96
13.22.1.4 t2FitFunc_sifunc 96
13.22.1.5 t2UpdateFunc_ifunc 96
13.23t2fit_position.h File Reference 96
13.23.1 Function Documentation 97
13.23.1.1 t2FitFunc_stdPosition 97
13.23.1.2 t2UpdateFunc_stdPosition 97
13.24t2fit_stdFitFuncs.h File Reference 97
13.24.1 Function Documentation 98
13.24.1.1 t2FitFunc_binaryModels 98
13.24.1.2 t2FitFunc_ifunc 98
13.24.1.3 t2FitFunc_jump 98
13.24.1.4 t2FitFunc_miscDm 98
13.24.1.5 t2FitFunc_planet 98
13.24.1.6 t2FitFunc_stdDm 98
13.24.1.7 t2FitFunc_stdFreq 98
13.24.1.8 t2FitFunc_stdGravWav 98
13.24.1.9 t2FitFunc_telPos 98
13.24.1.10t2FitFunc_zero 98
13.24.1.11t2UpdateFunc_binaryModels 98
13.24.1.12t2UpdateFunc_ifunc 98
13.24.1.13t2UpdateFunc_jump 98
13.24.1.14t2UpdateFunc_miscDm 98
13.24.1.15t2UpdateFunc_planet 98
13.24.1.16t2UpdateFunc_simpleAdd 99
13.24.1.17t2UpdateFunc_simpleMinus 99
13.24.1.18t2UpdateFunc_stdFreq 99
13.24.1.19t2UpdateFunc_stdGravWav 99
13.24.1.20t2UpdateFunc_telPos 99
13.24.1.21t2UpdateFunc_zero 99
13.25T2toolkit.h File Reference 99
13.25.1 Detailed Description 100
13.25.2 Function Documentation 100
13.25.2.1 genrand_int32 100
13.25.2.2 genrand_real1 100
13.25.2.3 init_genrand 100

13.25.2.4 TKconvertFloat1 100
13.25.2.5 TKconvertFloat2 100
13.25.2.6 TKfindMax_d 100
13.25.2.7 TKfindMax_f 100
13.25.2.8 TKfindMedian_d 100
13.25.2.9 TKfindMedian_f 100
13.25.2.10TKfindMin_d 100
13.25.2.11TKfindMin_f 100
13.25.2.12TKfindRMS_d 100
13.25.2.13TKfindRMS_f 100
13.25.2.14TKfindRMSweight_d 100
13.25.2.15TKgaussDev 100
13.25.2.16TKmean_d 100
13.25.2.17TKmean_f 100
13.25.2.18TKranDev 100
13.25.2.19TKrange_d 100
13.25.2.20TKrange_f 100
13.25.2.21TKretMax_d 100
13.25.2.22TKretMax_f 100
13.25.2.23TKretMin_d 101
13.25.2.24TKretMin_f 101
13.25.2.25TKretMin_i 101
13.25.2.26TKsetSeed 101
13.25.2.27TKsign_d 101
13.25.2.28TKsort_2f 101
13.25.2.29TKsort_3d 101
13.25.2.30TKsort_d 101
13.25.2.31TKsort_f 101
13.25.2.32TKvariance_d 101
13.25.2.33TKzeromean_d 101
13.26tabulatedfunction.h File Reference 101
13.26.1 Function Documentation 102
13.26.1.1 TabulatedFunction_getEndX 102
13.26.1.2 TabulatedFunction_getStartX 102
13.26.1.3 TabulatedFunction_getValue 102
13.26.1.4 TabulatedFunction_load 102
13.27tempo2.h File Reference 102
13.27.1 Detailed Description 108
13.27.2 Macro Definition Documentation 109
13.27.2.1 AU_DIST 109

13.27.2.2 AULTSC 109
13.27.2.3 BIG_G 109
13.27.2.4 DM_CONST 109
13.27.2.5 DM_CONST_SI 109
13.27.2.6 ECLIPTIC_OBLIQUITY_VAL 109
13.27.2.7 FB90_TIMEEPH 109
13.27.2.8 GM 109
13.27.2.9 GM_C3 109
13.27.2.10GMJ_C3 109
13.27.2.11GMN_C3 109
13.27.2.12GMS_C3 109
13.27.2.13GMU_C3 110
13.27.2.14GMV_C3 110
13.27.2.15HAVE_GWSIM_H 110
13.27.2.16IF99_TIMEEPH 110
13.27.2.17IFTEPH_FILE 110
13.27.2.18LEAPSECOND_FILE 110
13.27.2.19MASYR2RADS 110
13.27.2.20MAX_BPJ_JUMPS 110
13.27.2.21MAX_CLK_CORR 110
13.27.2.22MAX_CLKCORR 110
13.27.2.23MAX_COEFF 110
13.27.2.24MAX_COMPANIONS 110
13.27.2.25MAX_DM_DERIVATIVES 110
13.27.2.26MAX_DMX 111
13.27.2.27MAX_FILELEN 111
13.27.2.28MAX_FIT 111
13.27.2.29MAX_FLAG_LEN 111
13.27.2.30MAX_FLAGS 111
13.27.2.31MAX_FREQ_DERIVATIVES 111
13.27.2.32MAX_IFUNC 111
13.27.2.33MAX_JUMPS 111
13.27.2.34MAX_LEAPSEC 111
13.27.2.35MAX_MSG 111
13.27.2.36MAX_OBSN_VAL 111
13.27.2.37MAX_PARAMS 111
13.27.2.38MAX_PSR_VAL 112
13.27.2.39MAX_QUAD 112
13.27.2.40MAX_SITE 112
13.27.2.41MAX_STOREPRECISION 112

13.27.2.42MAX_STRLEN 112
13.27.2.43MAX_T2EFAC 112
13.27.2.44MAX_T2EQUAD 112
13.27.2.45MAX_TEL_CLK_OFFS 112
13.27.2.46MAX_TEL_DX 112
13.27.2.47MAX_TEL_DY 112
13.27.2.48MAX_TEL_DZ 112
13.27.2.49MAX_TNBN 112
13.27.2.50MAX_TNDMEv 112
13.27.2.51MAX_TNECORR 113
13.27.2.52MAX_TNEF 113
13.27.2.53MAX_TNEQ 113
13.27.2.54MAX_TNGN 113
13.27.2.55MAX_TNSQ 113
13.27.2.56MAX_TOFFSET 113
13.27.2.57MAX_WHITE 113
13.27.2.58NE_SW_DEFAULT 113
13.27.2.59OBLQ 113
13.27.2.60OBSSYS_FILE 113
13.27.2.61PCM 113
13.27.2.62SECDAY 113
13.27.2.63SECDAYl 114
13.27.2.64SI_UNITS 114
13.27.2.65SOLAR_MASS 114
13.27.2.66SOLAR_RADIUS 114
13.27.2.67SPEED_LIGHT 114
13.27.2.68T2C_IAU2000B 114
13.27.2.69T2C_TEMPO 114
13.27.2.70TDB_UNITS 114
13.27.2.71TDBTDT_FILE 114
13.27.2.72TEMPO2_h_HASH 114
13.27.2.73TEMPO2_h_MAJOR_VER 114
13.27.2.74TEMPO2_h_MINOR_VER 114
13.27.2.75TEMPO2_h_VER 114
13.27.2.76TSUN 114
13.27.2.77UT1_FILE 114
13.27.3 Typedef Documentation 115
13.27.3.1 constraint_label 115
13.27.3.2 constraintDerivFunc 115
13.27.3.3 FitInfo 115

13.27.3.4 observation 115
13.27.3.5 param_label 115
13.27.3.6 paramDerivFunc 115
13.27.3.7 parameter 115
13.27.3.8 paramUpdateFunc 115
13.27.3.9 pulsar 115
13.27.3.10storePrecision 116
13.27.4 Enumeration Type Documentation 116
13.27.4.1 constraint 116
13.27.4.2 label 117
13.27.5 Function Documentation 120
13.27.5.1 allocateMemory 120
13.27.5.2 autoConstraints 120
13.27.5.3 bootstrap 120
13.27.5.4 BTJmodel 120
13.27.5.5 BTmodel 120
13.27.5.6 BTXmodel 120
13.27.5.7 calcRMS 120
13.27.5.8 calculate_bclt 120
13.27.5.9 compute_tropospheric_delays 120
13.27.5.10copyParam 120
13.27.5.11copyPSR 120
13.27.5.12CVSdisplayVersion 120
13.27.5.13DDGRmodel 120
13.27.5.14DDHmodel 120
13.27.5.15DDKmodel 120
13.27.5.16DDmodel 120
13.27.5.17DDSmodel 120
13.27.5.18defineClockCorrectionSequence 120
13.27.5.19destroyMemory 120
13.27.5.20destroyOne 120
13.27.5.21displayMsg 120
13.27.5.22displayParameters 120
13.27.5.23dm_delays 121
13.27.5.24dms_turn 121
13.27.5.25doFit 121
13.27.5.26doFitAll 121
13.27.5.27doFitDCM 121
13.27.5.28doFitGlobal 121
13.27.5.29dotproduct 121

13.27.5.30ELL1Hmodel . . . 121
13.27.5.31ELL1model . . . 121
13.27.5.32equ2ecl . . . 121
13.27.5.33FITfuncs . . . 121
13.27.5.34formBats . . . 121
13.27.5.35formBatsAll . . . 121
13.27.5.36formResiduals . . . 121
13.27.5.37fortran_mod . . . 121
13.27.5.38fortran_nint . . . 121
13.27.5.39fortran_nlong . . . 121
13.27.5.40get_EOP . . . 121
13.27.5.41get_obsCoord . . . 121
13.27.5.42get_obsCoord_IAU2000B . . . 121
13.27.5.43get_OneobsCoord . . . 121
13.27.5.44getCholeskyMatrix . . . 121
13.27.5.45getClockCorrections . . . 121
13.27.5.46getCorrection . . . 121
13.27.5.47getCorrectionTT . . . 121
13.27.5.48getInputs . . . 122
13.27.5.49getObservatory . . . 122
13.27.5.50getParamDeriv . . . 122
13.27.5.51getParameterValue . . . 122
13.27.5.52hms_turn . . . 122
13.27.5.53id_residual . . . 122
13.27.5.54initialise . . . 122
13.27.5.55initialiseOne . . . 122
13.27.5.56JVmodel . . . 122
13.27.5.57logicFlag . . . 122
13.27.5.58lookup_observatory_alias . . . 122
13.27.5.59MSSmodel . . . 122
13.27.5.60polyco . . . 122
13.27.5.61preProcess . . . 122
13.27.5.62preProcessSimple . . . 122
13.27.5.63preProcessSimple1 . . . 122
13.27.5.64preProcessSimple2 . . . 122
13.27.5.65preProcessSimple3 . . . 122
13.27.5.66processFlag . . . 122
13.27.5.67processSimultaneous . . . 122
13.27.5.68readEphemeris . . . 122
13.27.5.69readEphemeris_calceph . . . 122

13.27.5.70 readJBO_bat 122
13.27.5.71 readObsFile 122
13.27.5.72 readOneEphemeris 122
13.27.5.73 readParfile 123
13.27.5.74 readParfileGlobal 123
13.27.5.75 readSimpleParfile 123
13.27.5.76 readTimfile 123
13.27.5.77 recordPrecision 123
13.27.5.78 secularMotion 123
13.27.5.79 setPlugPath 123
13.27.5.80 setStart 123
13.27.5.81 setupParameterFileDefaults 123
13.27.5.82 shapiro_delay 123
13.27.5.83 simplePlot 123
13.27.5.84 solarWindModel 123
13.27.5.85 sortToAs 123
13.27.5.86 T2_PTAmodel 123
13.27.5.87 T2model 123
13.27.5.88 tai2tt 123
13.27.5.89 tai2ut1 123
13.27.5.90 textOutput 123
13.27.5.91 toa2utc 123
13.27.5.92 transform_units 123
13.27.5.93 tt2tb 123
13.27.5.94 turn_deg 123
13.27.5.95 turn_dms 123
13.27.5.96 turn_hms 123
13.27.5.97 updateBatsAll 123
13.27.5.98 updateBT 123
13.27.5.99 updateBTJ 123
13.27.5.100 updateBTX 124
13.27.5.101 updateDD 124
13.27.5.102 updateDDGR 124
13.27.5.103 updateDDH 124
13.27.5.104 updateDDK 124
13.27.5.105 updateDDS 124
13.27.5.106 updateELL1 124
13.27.5.107 updateELL1H 124
13.27.5.108 updateJV 124
13.27.5.109 updateMSS 124

13.27.5.110 updateParameters 124
13.27.5.111 updateT2 124
13.27.5.112 updateT2_PTA 124
13.27.5.113 useSelectFile 124
13.27.5.114 utc2tai 124
13.27.5.115 vectorPulsar 124
13.27.5.116 vectorscale 124
13.27.5.117 vectorsum 124
13.27.5.118 writeTim 124
13.27.5.119 zoom_graphics 124
13.27.6 Variable Documentation 124
13.27.6.1 covarFuncFile 124
13.27.6.2 dcmFile 124
13.27.6.3 displayCVSversion 124
13.27.6.4 ECLIPTIC_OBLIQUITY 124
13.27.6.5 forceGlobalFit 124
13.27.6.6 MAX_OBSN 125
13.27.6.7 MAX_PSR 125
13.27.6.8 NEWFIT 125
13.27.6.9 TEMPO2_ENVIRON 125
13.27.6.10 TEMPO2_ERROR 125
13.27.6.11 tempo2_plug_path 125
13.27.6.12 tempo2_plug_path_len 125
13.27.6.13 tempo2MachineType 125
13.27.6.14 veryFast 125
13.28 tempo2pred.h File Reference 126
13.28.1 Enumeration Type Documentation 127
13.28.1.1 T2PredictorKind 127
13.28.2 Function Documentation 127
13.28.2.1 T2Predictor_Copy 127
13.28.2.2 T2Predictor_Destroy 127
13.28.2.3 T2Predictor_FRead 127
13.28.2.4 T2Predictor_FWrite 127
13.28.2.5 T2Predictor_GetEndFreq 127
13.28.2.6 T2Predictor_GetEndMJD 127
13.28.2.7 T2Predictor_GetFrequency 127
13.28.2.8 T2Predictor_GetPhase 127
13.28.2.9 T2Predictor_GetPlan 128
13.28.2.10 T2Predictor_GetPlan_Ext 128
13.28.2.11 T2Predictor_GetPSRName 128

13.28.2.12T2Predictor_GetSiteName 128
13.28.2.13T2Predictor_GetStartFreq 128
13.28.2.14T2Predictor_GetStartMJD 128
13.28.2.15T2Predictor_Init 128
13.28.2.16T2Predictor_Insert 128
13.28.2.17T2Predictor_Keep 128
13.28.2.18T2Predictor_Kind 128
13.28.2.19T2Predictor_Read 128
13.28.2.20T2Predictor_Write 128
13.28.3 Variable Documentation 128
13.28.3.1 ChebyModelSet_OutOfRange 128
13.29tempo2pred_int.h File Reference 128
13.29.1 Function Documentation 129
13.29.1.1 Cheby2D_Construct 129
13.29.1.2 Cheby2D_Construct_x_Derivative 129
13.29.1.3 Cheby2D_Test 129
13.29.1.4 ChebyModel_Construct 129
13.29.1.5 ChebyModel_Copy 130
13.29.1.6 ChebyModel_Destroy 130
13.29.1.7 ChebyModel_GetFrequency 130
13.29.1.8 ChebyModel_GetPhase 130
13.29.1.9 ChebyModel_Init 130
13.29.1.10ChebyModel_Read 130
13.29.1.11ChebyModel_Test 130
13.29.1.12ChebyModel_Write 130
13.29.1.13ChebyModelSet_Construct 130
13.29.1.14ChebyModelSet_Destroy 130
13.29.1.15ChebyModelSet_GetFrequency 130
13.29.1.16ChebyModelSet_GetNearest 130
13.29.1.17ChebyModelSet_GetPhase 130
13.29.1.18ChebyModelSet_Init 130
13.29.1.19ChebyModelSet_Insert 130
13.29.1.20ChebyModelSet_Keep 130
13.29.1.21ChebyModelSet_Read 130
13.29.1.22ChebyModelSet_Test 130
13.29.1.23ChebyModelSet_Write 130
13.29.1.24T1Polyco_GetFrequency 130
13.29.1.25T1Polyco_GetPhase 130
13.29.1.26T1Polyco_Read 130
13.29.1.27T1Polyco_Write 130

13.29.1.28T1PolycoSet_Destroy 130
13.29.1.29T1PolycoSet_GetFrequency 130
13.29.1.30T1PolycoSet_GetNearest 130
13.29.1.31T1PolycoSet_GetPhase 131
13.29.1.32T1PolycoSet_Read 131
13.29.1.33T1PolycoSet_Write 131
13.30tempo2Util.h File Reference 131
13.30.1 Function Documentation 131
13.30.1.1 dms_turn 131
13.30.1.2 hms_turn 131
13.30.1.3 turn_deg 131
13.31TKcholesky.h File Reference 131
13.31.1 Function Documentation 131
13.31.1.1 cholesky_covarFunc2matrix 131
13.31.1.2 cholesky_dmModel 131
13.31.1.3 cholesky_dmModelCovarParam 131
13.31.1.4 cholesky_ecm 131
13.31.1.5 cholesky_formUinv 132
13.31.1.6 cholesky_powerlawModel 132
13.31.1.7 cholesky_powerlawModel_withBeta 132
13.31.1.8 cholesky_readFromCovarianceFunction 132
13.32TKfit.h File Reference 132
13.32.1 Function Documentation 133
13.32.1.1 TKconstrainedLeastSquares 133
13.32.1.2 TKfindPoly_d 133
13.32.1.3 TKfitPoly 133
13.32.1.4 TKleastSquares 133
13.32.1.5 TKleastSquares_svd 133
13.32.1.6 TKleastSquares_svd_noErr 133
13.32.1.7 TKremovePoly_d 133
13.32.1.8 TKremovePoly_f 133
13.32.1.9 TKrobustConstrainedLeastSquares 133
13.32.1.10TKrobustLeastSquares 133
13.33TKlog.h File Reference 133
13.33.1 Macro Definition Documentation 135
13.33.1.1 _LOG 135
13.33.1.2 BOLDCOLOR 135
13.33.1.3 DEPRECATED 135
13.33.1.4 ENDERR 135
13.33.1.5 ENDL 135

13.33.1.6 ERRORCOLOR 135
13.33.1.7 LOG_OUTFILE 135
13.33.1.8 logdbg 135
13.33.1.9 logerr 135
13.33.1.10logmsg 135
13.33.1.11logtchk 135
13.33.1.12logwarn 135
13.33.1.13RESETCOLOR 135
13.33.1.14TK_MAX_ERROR_LEN 135
13.33.1.15TK_MAX_ERRORS 135
13.33.1.16TK_STORE_ERROR 136
13.33.1.17TK_STORE_WARNING 136
13.33.1.18WARNCOLOR 136
13.33.1.19WHEREARG 136
13.33.1.20WHEREERR 136
13.33.1.21WHERESTR 136
13.33.1.22WHERETCHK 136
13.33.1.23WHEREWARN 136
13.33.2 Function Documentation 136
13.33.2.1 _TKchklog 136
13.33.2.2 logerr_check 136
13.33.3 Variable Documentation 136
13.33.3.1 debugFlag 136
13.33.3.2 tcheck 136
13.33.3.3 timer_clk 136
13.33.3.4 TK_errorCount 136
13.33.3.5 TK_errorlog 136
13.33.3.6 TK_warnCount 136
13.33.3.7 TK_warnlog 136
13.33.3.8 writeResiduals 136
13.34TKlongdouble.float128.h File Reference 136
13.34.1 Macro Definition Documentation 137
13.34.1.1 cosl 137
13.34.1.2 fabsl 137
13.34.1.3 floorl 137
13.34.1.4 FMT_LD 137
13.34.1.5 LD_PI 137
13.34.1.6 longdouble 138
13.34.1.7 LONGDOUBLE_IS_FLOAT128 138
13.34.1.8 LONGDOUBLE_ONE 138

13.34.1.9 sinl 138
13.34.1.10USE_BUILTIN_LONGDOUBLE 138
13.34.2 Typedef Documentation 138
13.34.2.1 longdouble 138
13.34.3 Function Documentation 138
13.34.3.1 ld_fprintf 138
13.34.3.2 ld_printf 138
13.34.3.3 ld_sprintf 138
13.34.3.4 parse_longdouble 138
13.35TKlongdouble.h File Reference 138
13.35.1 Macro Definition Documentation 139
13.35.1.1 cosl 139
13.35.1.2 fabsl 139
13.35.1.3 floorl 139
13.35.1.4 FMT_LD 139
13.35.1.5 LD_PI 139
13.35.1.6 longdouble 139
13.35.1.7 LONGDOUBLE_IS_FLOAT128 139
13.35.1.8 LONGDOUBLE_ONE 139
13.35.1.9 sinl 139
13.35.1.10USE_BUILTIN_LONGDOUBLE 139
13.35.2 Typedef Documentation 139
13.35.2.1 longdouble 139
13.35.3 Function Documentation 139
13.35.3.1 ld_fprintf 139
13.35.3.2 ld_printf 140
13.35.3.3 ld_sprintf 140
13.35.3.4 parse_longdouble 140
13.36TKlongdouble.ld.h File Reference 140
13.36.1 Macro Definition Documentation 140
13.36.1.1 ld_fprintf 140
13.36.1.2 LD_PI 140
13.36.1.3 ld_printf 140
13.36.1.4 ld_sprintf 141
13.36.1.5 longdouble 141
13.36.1.6 LONGDOUBLE_IS_IEEE754 141
13.36.1.7 LONGDOUBLE_ONE 141
13.36.1.8 USE_BUILTIN_LONGDOUBLE 141
13.36.2 Typedef Documentation 141
13.36.2.1 longdouble 141

13.36.3 Function Documentation 141
13.36.3.1 parse_longdouble 141
13.37TKmatrix.h File Reference 141
13.37.1 Function Documentation 142
13.37.1.1 free_2df 142
13.37.1.2 free_blas 142
13.37.1.3 free_uinv 142
13.37.1.4 get_blas_cols 142
13.37.1.5 get_blas_rows 142
13.37.1.6 malloc_2df 142
13.37.1.7 malloc_blas 142
13.37.1.8 malloc_uinv 142
13.37.1.9 TKmultMatrix 142
13.37.1.10TKmultMatrix_sq 142
13.37.1.11TKmultMatrixVec 142
13.37.1.12TKmultMatrixVec_sq 142
13.38TKspectrum.h File Reference 142
13.38.1 Macro Definition Documentation 144
13.38.1.1 ABS 144
13.38.1.2 MAX 144
13.38.1.3 MIN 144
13.38.2 Typedef Documentation 144
13.38.2.1 complexVal 144
13.38.3 Function Documentation 144
13.38.3.1 calcSpectra 144
13.38.3.2 calcSpectra_ri 144
13.38.3.3 calcSpectra_ri_T 144
13.38.3.4 calcSpectraErr 144
13.38.3.5 fit4 144
13.38.3.6 fitCosSineFunc 144
13.38.3.7 fitCosSineFunc 144
13.38.3.8 fitMeanSineFunc 144
13.38.3.9 fitMeanSineFunc_IFUNC 144
13.38.3.10getprtj 144
13.38.3.11getweights 144
13.38.3.12indexx8 144
13.38.3.13mat20 144
13.38.3.14readin 144
13.38.3.15sineFunc 145
13.38.3.16TK_dft 145

13.38.3.17TK_fft 145
13.38.3.18TK_fitSine 145
13.38.3.19TK_fitSinusoids 145
13.38.3.20TK_weightLS 145
13.38.3.21TKaveragePts 145
13.38.3.22TKboxcar 145
13.38.3.23TKcalcSigmaz 145
13.38.3.24TKcmonot 145
13.38.3.25TKfirstDifference 145
13.38.3.26TKhann 145
13.38.3.27TKinterpolateSplineSmoothFixedXPts 145
13.38.3.28TKlomb_d 145
13.38.3.29TKsortit 145
13.38.3.30TKspectrum 145
13.38.3.31TKspline_interpolate 145
13.38.4 Variable Documentation 145
13.38.4.1 GLOBAL_OMEGA 145
13.38.4.2 verbose_calc_spectra 145
13.39TKsvd.h File Reference 145
13.39.1 Function Documentation 146
13.39.1.1 TKbacksubstitution_svd 146
13.39.1.2 TKbidiagonal 146
13.39.1.3 TKpythag 146
13.39.1.4 TKsingularValueDecomposition_lsq 146

Index **147**

Chapter 1

Main Page

- User Guide
- Developer Guide
- Directory structure

Chapter 2

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

1. What this package is

You (or someone else) have checked out tempo2 from the Git (https://bitbucket.org/mkeith/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 suitible makefiles for the default set of plugins. This is controled 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.
- `{$LOGIN_ARCH}` is the result of `` `uname` `` (usualy Linux).
- `{$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 dependances 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 makefiles. This may confuse some people!

Important notes:

- Tempo2 plugins now have a .t2 extention, 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 extention to a .t2 extention. e.g. mv general_Linux_plug.so general_Linux_plug.t2

7. Installation Troubleshooting

7.1 Can't find PGPLOT

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

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

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

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

7.2 Incompatible C and Fortran compilers

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

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

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

and reconfigure.

Chapter 3

User Guide

Chapter 4

Core Developers

Tempo2 development team

Tempo2 was origianaly written by George Hobbs and Rusell Edwards.

Core package maintainers

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

Active contributors

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

Past Contributors

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

Chapter 5

Developer Guide

5.1 Tempo2 Developer Guide

5.1.1 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.

5.1.2 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++ clases 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 avaliable 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 temo2.

5.1.3 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.

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


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

    // use stdint types where possible to avoid confusion on 32-bit vs 64-bit machines.
    // use unsigned types whre sutable
    // use const when a variable will not change
    const uint64_t myconst = 1024;
    // keywords have a space before parenthesis (e.g. if, for, while).
    if (anInt < 10) { // always use braces, even if one line!
        // use TKlog for logging debug messages and warnings.

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

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

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

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

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

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

}
```

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] descrition, note if it is an "output" parameter!
 *
 *  More description if required
 */


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

Chapter 6

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 ephemerdies, 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 7

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 8

Module Index

8.1 Modules

Here is a list of all modules:

libt2toolkit API 25

libtempo2 External API 26

Chapter 9

Class Index

9.1 Class List

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

Cheby2D 27
ChebyModel 27
ChebyModelSet 29
clock_correction 29
complexVal 30
DynamicArray 30
FitInfo
Details of the fit 31
gwgeneralSrc 32
gwgenSpec 34
gwSrc 34
interpolation_info 35
jpl_eph_data 36
observation
A struct containing the details of a single obesrvation 37
observatory 45
parameter
Holds the values for a parameter 45
pulsar
Details for a single pulsar 47
storePrecision 66
T1Polyco 66
T1PolycoSet 68
T2Predictor 68
TabulatedFunction 70
TabulatedFunctionSample 70

Chapter 10

File Index

10.1 File List

Here is a list of all files with brief descriptions:

cholesky.h . . . 73
choleskyRoutines.h . . . 74
config.h . . . 76
constraints.h . . . 78
dynarr.h . . . 81
GWsim.h . . . 82
ifteph.h . . . 84
jpl_int.h . . . 86
jpleph.h . . . 86
read_fortran.h . . . 88
read_fortran2.h . . . 89
T2accel.h . . . 90
t2fit.h . . . 91
t2fit_dmmodel.h . . . 92
t2fit_fitwaves.h . . . 94
t2fit_glitch.h . . . 94
t2fit_ifunc.h . . . 95
t2fit_position.h . . . 96
t2fit_stdFitFuncs.h . . . 97
T2toolkit.h
Set of routines that are commonly used in tempo2 and/or its plugins . . . 99
tabulatedfunction.h . . . 101
tempo2.h
Main interface to libtempo2 . . . 102
tempo2pred.h . . . 126
tempo2pred_int.h . . . 128
tempo2Util.h . . . 131
TKcholesky.h . . . 131
TKfit.h . . . 132
TKlog.h . . . 133
TKlongdouble.float128.h . . . 136
TKlongdouble.h . . . 138
TKlongdouble.ld.h . . . 140
TKmatrix.h . . . 141
TKspectrum.h . . . 142
TKsvd.h . . . 145

Chapter 11

Module Documentation

11.1 libt2toolkit API

Files

- file T2toolkit.h

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

11.1.1 Detailed Description

11.2 libtempo2 External API

Files

- file tempo2.h

  contains the main interface to libtempo2.

11.2.1 Detailed Description

Chapter 12

Class Documentation

12.1 Cheby2D Struct Reference

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

Public Attributes

- int nx
- int ny
- long double ∗ coeff

12.1.1 Member Data Documentation

12.1.1.1 long double∗ Cheby2D::coeff

12.1.1.2 int Cheby2D::nx

12.1.1.3 int Cheby2D::ny

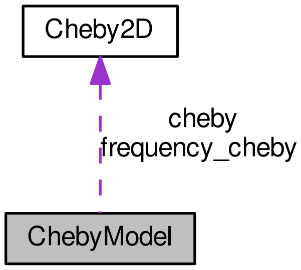
The documentation for this struct was generated from the following file:

- tempo2pred.h

12.2 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

12.2.1 Member Data Documentation

12.2.1.1 Cheby2D ChebyModel::cheby

12.2.1.2 long double ChebyModel::dispersion_constant

12.2.1.3 long double ChebyModel::freq_end

12.2.1.4 long double ChebyModel::freq_start

12.2.1.5 Cheby2D ChebyModel::frequency_cheby

12.2.1.6 long double ChebyModel::mjd_end

12.2.1.7 long double ChebyModel::mjd_start

12.2.1.8 char ChebyModel::psrname[64]

12.2.1.9 char ChebyModel::sitename[64]

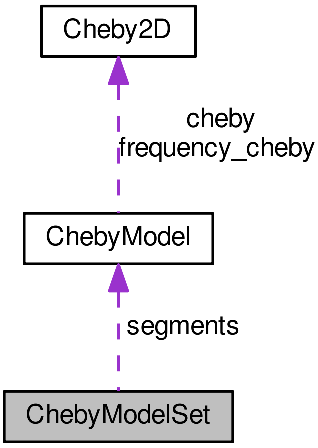
The documentation for this struct was generated from the following file:

- tempo2pred.h

12.3 ChebyModelSet Struct Reference

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

Collaboration diagram for ChebyModelSet:



Public Attributes

- ChebyModel ∗ segments
- int nsegments

12.3.1 Member Data Documentation

12.3.1.1 int ChebyModelSet::nsegments

12.3.1.2 ChebyModel∗ ChebyModelSet::segments

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

- tempo2pred.h

12.4 clock_correction Struct Reference

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

Public Attributes

- double correction
- char corrects_to [32]

12.4.1 Detailed Description

observation contains an array of these, which getClockCorrections() fills in

12.4.2 Member Data Documentation

12.4.2.1 double clock_correction::correction

12.4.2.2 char clock_correction::corrects_to[32]

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

- tempo2.h

12.5 complexVal Struct Reference

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

Public Attributes

- double real
- double imag

12.5.1 Member Data Documentation

12.5.1.1 double complexVal::imag

12.5.1.2 double complexVal::real

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

- TKspectrum.h

12.6 DynamicArray Struct Reference

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

Public Attributes

- void ∗ data
- size_t nelem
- size_t elem_size
- size_t nalloced

12.6.1 Member Data Documentation

12.6.1.1 void∗ DynamicArray::data

12.6.1.2 size_t DynamicArray::elem_size

12.6.1.3 size_t DynamicArray::nalloced

12.6.1.4 size_t DynamicArray::nelem

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

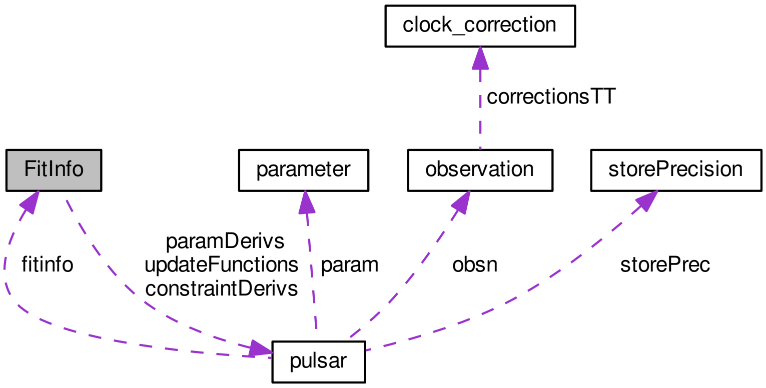
- dynarr.h

12.7 FitInfo Struct Reference

contains details of the fit

`#include <tempo2.h>`

Collaboration diagram for FitInfo:



Public Attributes

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

12.7.1 Detailed Description

contains details of the fit

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

12.7.2 Member Data Documentation

12.7.2.1 int FitInfo::constraintCounters[MAX_FIT]

12.7.2.2 constraintDerivFunc FitInfo::constraintDerivs[MAX_FIT]

12.7.2.3 constraint_label FitInfo::constraintIndex[MAX_FIT]

12.7.2.4 unsigned FitInfo::nConstraints

12.7.2.5 unsigned FitInfo::nParams

12.7.2.6 int FitInfo::paramCounters[MAX_FIT]

12.7.2.7 paramDerivFunc FitInfo::paramDerivs[MAX_FIT]

12.7.2.8 param_label FitInfo::paramIndex[MAX_FIT]

12.7.2.9 paramUpdateFunc FitInfo::updateFunctions[MAX_FIT]

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

- tempo2.h

12.8 gwgeneralSrc Struct Reference

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

Public Attributes

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

12.8.1 Member Data Documentation

12.8.1.1 longdouble gwgeneralSrc::across_g

12.8.1.2 longdouble gwgeneralSrc::across_im_g

12.8.1.3 longdouble gwgeneralSrc::aplus_g

12.8.1.4 longdouble gwgeneralSrc::aplus_im_g

12.8.1.5 longdouble gwgeneralSrc::asl_g

12.8.1.6 longdouble gwgeneralSrc::asl_im_g

12.8.1.7 longdouble gwgeneralSrc::ast_g

12.8.1.8 longdouble gwgeneralSrc::ast_im_g

12.8.1.9 longdouble gwgeneralSrc::avx_g

12.8.1.10 longdouble gwgeneralSrc::avx_im_g

12.8.1.11 longdouble gwgeneralSrc::avy_g

12.8.1.12 longdouble gwgeneralSrc::avy_im_g

12.8.1.13 longdouble gwgeneralSrc::dist_bin

12.8.1.14 longdouble gwgeneralSrc::h[3][3]

12.8.1.15 longdouble gwgeneralSrc::h_im[3][3]

12.8.1.16 longdouble gwgeneralSrc::inc_bin

12.8.1.17 longdouble gwgeneralSrc::kg[3]

12.8.1.18 longdouble gwgeneralSrc::omega_g

12.8.1.19 longdouble gwgeneralSrc::phase_g

12.8.1.20 longdouble gwgeneralSrc::phi_bin

12.8.1.21 longdouble gwgeneralSrc::phi_g

12.8.1.22 longdouble gwgeneralSrc::phi_polar_g

12.8.1.23 longdouble gwgeneralSrc::theta_bin

12.8.1.24 longdouble gwgeneralSrc::theta_g

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

- GWsim.h

12.9 gwgenSpec Struct Reference

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

Public Attributes

- double tensor_amp
- double st_amp
- double sl_amp
- double vl_amp
- double tensor_alpha
- double st_alpha
- double sl_alpha
- double vl_alpha

12.9.1 Member Data Documentation

12.9.1.1 double gwgenSpec::sl_alpha

12.9.1.2 double gwgenSpec::sl_amp

12.9.1.3 double gwgenSpec::st_alpha

12.9.1.4 double gwgenSpec::st_amp

12.9.1.5 double gwgenSpec::tensor_alpha

12.9.1.6 double gwgenSpec::tensor_amp

12.9.1.7 double gwgenSpec::vl_alpha

12.9.1.8 double gwgenSpec::vl_amp

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

- GWsim.h

12.10 gwSrc Struct Reference

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

Public Attributes

- longdouble theta_g
- longdouble phi_g
- longdouble omega_g
- longdouble phi_polar_g
- longdouble phase_g
- longdouble aplus_g
- longdouble aplus_im_g
- longdouble across_g
- longdouble across_im_g

- longdouble phi_bin
- longdouble theta_bin
- longdouble inc_bin
- longdouble dist_bin
- longdouble h [3][3]
- longdouble h_im [3][3]
- longdouble kg [3]

12.10.1 Member Data Documentation

12.10.1.1 longdouble gwSrc::across_g

12.10.1.2 longdouble gwSrc::across_im_g

12.10.1.3 longdouble gwSrc::aplus_g

12.10.1.4 longdouble gwSrc::aplus_im_g

12.10.1.5 longdouble gwSrc::dist_bin

12.10.1.6 longdouble gwSrc::h[3][3]

12.10.1.7 longdouble gwSrc::h_im[3][3]

12.10.1.8 longdouble gwSrc::inc_bin

12.10.1.9 longdouble gwSrc::kg[3]

12.10.1.10 longdouble gwSrc::omega_g

12.10.1.11 longdouble gwSrc::phase_g

12.10.1.12 longdouble gwSrc::phi_bin

12.10.1.13 longdouble gwSrc::phi_g

12.10.1.14 longdouble gwSrc::phi_polar_g

12.10.1.15 longdouble gwSrc::theta_bin

12.10.1.16 longdouble gwSrc::theta_g

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

- GWsim.h

12.11 interpolation_info Struct Reference

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

Public Attributes

- double pc [18]

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

12.11.1 Member Data Documentation

12.11.1.1 int interpolation_info::np

12.11.1.2 int interpolation_info::nv

12.11.1.3 double interpolation_info::pc[18]

12.11.1.4 double interpolation_info::twot

12.11.1.5 double interpolation_info::vc[18]

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

- jpl_int.h

12.12 jpl_eph_data Struct Reference

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

Public Attributes

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

12.12.1 Member Data Documentation

12.12.1.1 double jpl_eph_data::au

12.12.1.2 double∗ jpl_eph_data::cache

12.12.1.3 JPLlong jpl_eph_data::curr_cache_loc

12.12.1.4 double jpl_eph_data::emrat

12.12.1.5 double jpl_eph_data::ephem_end

12.12.1.6 double jpl_eph_data::ephem_start

12.12.1.7 double jpl_eph_data::ephem_step

12.12.1.8 JPLlong jpl_eph_data::ephemeris_version

12.12.1.9 FILE∗ jpl_eph_data::ifile

12.12.1.10 void∗ jpl_eph_data::iinfo

12.12.1.11 JPLlong jpl_eph_data::ipt[13][3]

12.12.1.12 JPLlong jpl_eph_data::kernel_size

12.12.1.13 JPLlong jpl_eph_data::ncoeff

12.12.1.14 JPLlong jpl_eph_data::ncon

12.12.1.15 double jpl_eph_data::pvsun[6]

12.12.1.16 JPLlong jpl_eph_data::recsize

12.12.1.17 JPLlong jpl_eph_data::swap_bytes

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

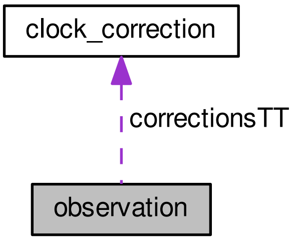
- jpl_int.h

12.13 observation Struct Reference

A struct containing the details of a single obesrvation.

`#include <tempo2.h>`

Collaboration diagram for observation:



Public Attributes

- longdouble sat
- longdouble origsat
- longdouble sat_day
- longdouble sat_sec
- longdouble bat
- longdouble batCorr
- longdouble bbat
- longdouble pet
- int clockCorr
- int delayCorr
- int deleted
- longdouble prefitResidual
- longdouble residual
- double addedNoise
- double TNRedSignal
- double TNRedErr
- double TNDMSignal
- double TNDMErr
- double TNGroupSignal
- double TNGroupErr
- double freq
- double freqSSB
- double toaErr
- double toaDMErr
- double origErr
- double phaseOffset
- double averagebat
- double averageres
- double averageerr
- char fname [MAX_FILELEN]
- char telID [100]
- clock_correction correctionsTT [MAX_CLK_CORR]
- int nclock_correction
- longdouble correctionTT_TB
- double einsteinRate
- longdouble correctionTT_Teph
- longdouble correctionUT1
- double sun_ssb [6]
- double sun_earth [6]
- double planet_ssb [9][6]
- double jupiter_earth [6]
- double saturn_earth [6]
- double venus_earth [6]
- double uranus_earth [6]
- double neptune_earth [6]
- double earthMoonBary_ssb [6]
- double earthMoonBary_earth [6]
- double earth_ssb [6]
- double observatory_earth [6]
- double psrPos [3]
- double zenith [3]
- double nutations [6]
- double siteVel [3]

- longdouble shklovskii
- double shapiroDelaySun
- double shapiroDelayJupiter
- double shapiroDelaySaturn
- double shapiroDelayVenus
- double shapiroDelayUranus
- double shapiroDelayNeptune
- double troposphericDelay
- double tdis1
- double tdis2
- longdouble roemer
- longdouble torb
- longdouble nphase
- longdouble phase
- long long pulseN
- char flagID [MAX_FLAGS][MAX_FLAG_LEN]
- char flagVal [MAX_FLAGS][MAX_FLAG_LEN]
- int nFlags
- int jump [MAX_FLAGS]
- int obsNjump
- double efac
- double equad

12.13.1 Detailed Description

A struct containing the details of a single obesrvation.

12.13.2 Member Data Documentation

12.13.2.1 double observation::addedNoise

12.13.2.2 double observation::averagebat

12.13.2.3 double observation::averageerr

12.13.2.4 double observation::averageres

12.13.2.5 longdouble observation::bat

Infinite frequency barycentric arrival time

12.13.2.6 longdouble observation::batCorr

12.13.2.7 longdouble observation::bbat

Arrival time at binary barycentre

12.13.2.8 int observation::clockCorr

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

12.13.2.9 clock_correction observation::correctionsTT[MAX_CLK_CORR]

chain of corrections from site TOA to chosen realisation of TT

12.13.2.10 longdouble observation::correctionTT_TB

Correction to TDB/TCB

12.13.2.11 longdouble observation::correctionTT_Teph

Correction to Teph

12.13.2.12 longdouble observation::correctionUT1

Correction from site TOA to UT1

12.13.2.13 int observation::delayCorr

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

12.13.2.14 int observation::deleted

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

12.13.2.15 double observation::earth_ssb[6]

Centre of Earth w.r.t. SSB

12.13.2.16 double observation::earthMoonBary_earth[6]

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

12.13.2.17 double observation::earthMoonBary_ssb[6]

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

12.13.2.18 double observation::efac

Error multiplication factor

12.13.2.19 double observation::einsteinRate

Derivative of correctionTT_TB

12.13.2.20 double observation::equad

Value to add in quadrature

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

Flags in .tim file

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

12.13.2.23 char observation::fname[MAX_FILELEN]

Name of data file giving TOA

12.13.2.24 double observation::freq

Frequency of observation (in MHz)

12.13.2.25 double observation::freqSSB

Frequency of observation in barycentric frame (in Hz)

12.13.2.26 int observation::jump[MAX_FLAGS]

Jump region

12.13.2.27 double observation::jupiter_earth[6]

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

12.13.2.28 int observation::nclock_correction

12.13.2.29 double observation::neptune_earth[6]

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

12.13.2.30 int observation::nFlags

12.13.2.31 longdouble observation::nphase

allows the pulse number to be determined

12.13.2.32 double observation::nutations[6]

12.13.2.33 double observation::observatory_earth[6]

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

12.13.2.34 int observation::obsNjump

Number of jumps for this observation

12.13.2.35 double observation::origErr

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

12.13.2.36 longdouble observation::origsat

12.13.2.37 longdouble observation::pet

Pulsar emission time

12.13.2.38 longdouble observation::phase

12.13.2.39 double observation::phaseOffset

Phase offset

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

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

12.13.2.41 longdouble observation::prefitResidual

Pre-fit residual

12.13.2.42 double observation::psrPos[3]

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

12.13.2.43 long long observation::pulseN

Pulse number

12.13.2.44 longdouble observation::residual

residual

12.13.2.45 longdouble observation::roemer

Roemer delay

12.13.2.46 longdouble observation::sat

Site arrival time

12.13.2.47 longdouble observation::sat_day

12.13.2.48 longdouble observation::sat_sec

12.13.2.49 double observation::saturn_earth[6]

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

12.13.2.50 double observation::shapiroDelayJupiter

Shapiro Delay due to Jupiter

12.13.2.51 double observation::shapiroDelayNeptune

Shapiro Delay due to Neptune

12.13.2.52 double observation::shapiroDelaySaturn

Shapiro Delay due to Saturn

12.13.2.53 double observation::shapiroDelaySun

Shapiro Delay due to the Sun

12.13.2.54 double observation::shapiroDelayUranus

Shapiro Delay due to Uranus

12.13.2.55 double observation::shapiroDelayVenus

Shapiro Delay due to Venus

12.13.2.56 longdouble observation::shklovskii

Shklovskii delay term

12.13.2.57 double observation::siteVel[3]

Observatory velocity w.r.t. geocentre

12.13.2.58 double observation::sun_earth[6]

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

12.13.2.59 double observation::sun_ssb[6]

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

12.13.2.60 double observation::tdis1

Interstellar dispersion measure delay

12.13.2.61 double observation::tdis2

Dispersion measure delay due to solar system

12.13.2.62 char observation::tellD[100]

Telescope ID

12.13.2.63 double observation::TNDMErr

Error on Model DM signal from temponest fit

12.13.2.64 double observation::TNDMSignal

Model DM signal from temponest fit

12.13.2.65 double observation::TNGroupErr

Error on Model Group Noise signal from temponest fit

12.13.2.66 double observation::TNGroupSignal

Model Group Noise signal from temponest fit

12.13.2.67 double observation::TNRedErr

Error on Model red noise signal from temponest fit

12.13.2.68 double observation::TNRedSignal

Model red noise signal from temponest fit

12.13.2.69 double observation::toaDMErr

Error on TOA due to DM (in us)

12.13.2.70 double observation::toaErr

Error on TOA (in us)

12.13.2.71 longdouble observation::torb

Combined binary delays

12.13.2.72 double observation::troposphericDelay

Delay due to neutral refraction in atmosphere

12.13.2.73 double observation::uranus_earth[6]

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

12.13.2.74 double observation::venus_earth[6]

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

12.13.2.75 double observation::zenith[3]

Zenith vector, in BC frame. Length=geodetic height

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

- tempo2.h

12.14 observatory Struct Reference

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

Public Attributes

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

12.14.1 Member Data Documentation

12.14.1.1 char observatory::clock_name[16]

12.14.1.2 char observatory::code[16]

12.14.1.3 double observatory::height_grs80

12.14.1.4 double observatory::latitude_grs80

12.14.1.5 double observatory::longitude_grs80

12.14.1.6 char observatory::name[32]

12.14.1.7 double observatory::x

12.14.1.8 double observatory::y

12.14.1.9 double observatory::z

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

- tempo2.h

12.15 parameter Struct Reference

Holds the values for a parameter.

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

Public Attributes

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

12.15.1 Detailed Description

Holds the values for a parameter.

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

Note

> If this structure is modified - must update copyParam in tempo2Util.C

12.15.2 Member Data Documentation

12.15.2.1 int parameter::aSize

Number of elements in the array for this parameter

12.15.2.2 longdouble∗ parameter::err

Uncertainty on parameter value

12.15.2.3 int∗ parameter::fitFlag

= 1 if fitting required, = 2 for global fit

12.15.2.4 char∗∗ parameter::label

Label about this parameter

12.15.2.5 int parameter::linkFrom[5]

12.15.2.6 int parameter::linkTo[5]

12.15.2.7 int parameter::nLinkFrom

12.15.2.8 int parameter::nLinkTo

12.15.2.9 int∗ parameter::paramSet

= 1 if parameter has been set

12.15.2.10 longdouble∗ parameter::prefit

Pre-fit value of the parameter

12.15.2.11 longdouble∗ parameter::prefitErr

Pre-fit value of the uncertainty

12.15.2.12 char∗∗ parameter::shortlabel

Label about this parameter without units

12.15.2.13 longdouble∗ parameter::val

Value of parameter

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

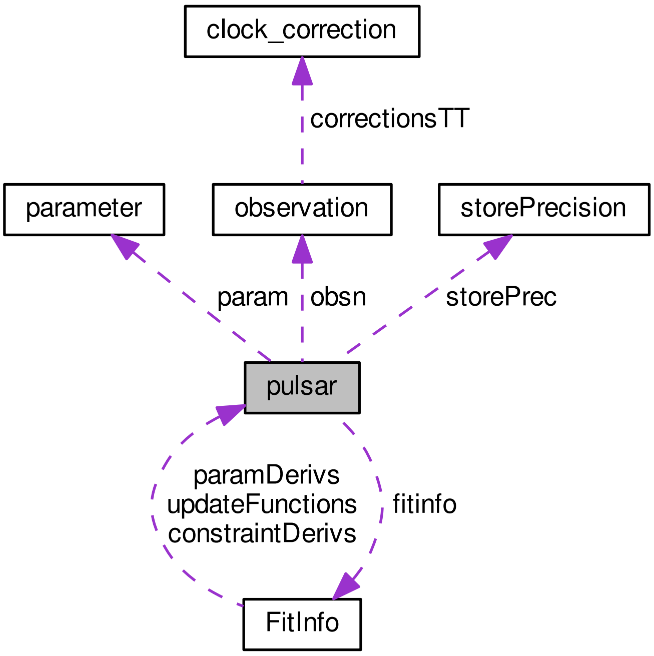
- tempo2.h

12.16 pulsar Struct Reference

contains the details for a single pulsar.

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

Collaboration diagram for pulsar:



Public Attributes

- char name [100]
- char eopc04_file [MAX_FILELEN]
- int fixedFormat
- parameter param [MAX_PARAMS]
- char rajStrPre [100]
- char decjStrPre [100]
- char rajStrPost [100]
- char decjStrPost [100]
- char binaryModel [100]
- double $**$ ToAextraCovar
- int dmoffsDMnum
- int dmoffsCMnum
- double dmoffsDM_mjd [MAX_IFUNC]
- double dmoffsDM [MAX_IFUNC]
- double dmoffsDM_error [MAX_IFUNC]
- double dmoffsDM_weight [MAX_IFUNC]
- double dmoffsCM_mjd [MAX_IFUNC]
- double dmoffsCM [MAX_IFUNC]
- double dmoffsCM_error [MAX_IFUNC]
- double dmoffsCM_weight [MAX_IFUNC]
- double gwsrc_ra
- double gwsrc_dec
- double gwsrc_aplus_r
- double gwsrc_aplus_i
- double gwsrc_across_r
- double gwsrc_across_i
- double gwsrc_aplus_r_e
- double gwsrc_aplus_i_e
- double gwsrc_across_r_e
- double gwsrc_across_i_e
- double gwsrc_epoch
- double gwsrc_psrdist
- double cgw_h0
- double cgw_cosinc
- double cgw_angpol
- double cgw_mc
- double gwm_raj
- double gwm_decj
- double gwm_epoch
- double gwm_phi
- double gwm_dphase
- double gwb_epoch
- double gwb_width
- double gwb_raj
- double gwb_decj
- double gwb_geom_c
- double gwb_geom_p
- double gwecc_ra
- double gwecc_dec
- double gwecc_m1
- double gwecc_m2
- double gwecc_e
- double gwecc_inc

- double gwecc_theta_nodes
- double gwecc_nodes_orientation
- double gwecc_theta_0
- double gwecc_orbital_period
- double gwecc_distance
- double gwecc_redshift
- double gwecc_epoch
- double gwecc_psrdist
- int gwecc_pulsarTermOn
- double posPulsar [3]
- double velPulsar [3]
- longdouble phaseJump [MAX_JUMPS]
- int phaseJumpDir [MAX_JUMPS]
- int phaseJumpID [MAX_JUMPS]
- int nPhaseJump
- double dmOffset
- double ne_sw
- int nCompanion
- int eclCoord
- int nJumps
- char fjumpID [16]
- double jumpVal [MAX_JUMPS]
- int fitJump [MAX_JUMPS]
- double jumpValErr [MAX_JUMPS]
- char jumpStr [MAX_JUMPS][MAX_STRLEN]
- char filterStr [MAX_STRLEN]
- char passStr [MAX_STRLEN]
- double tOffset [MAX_TOFFSET]
- double tOffset_f1 [MAX_TOFFSET]
- double tOffset_f2 [MAX_TOFFSET]
- double tOffset_t1 [MAX_TOFFSET]
- double tOffset_t2 [MAX_TOFFSET]
- char tOffsetSite [MAX_TOFFSET][100]
- char tOffsetFlags [MAX_TOFFSET][1000]
- int nToffset
- int ndmx
- double fitChisq
- int fitNfree
- int globalNfit
- int globalNoConstrain
- int nFit
- int nParam
- int nGlobal
- int fitParamGlobalI [MAX_FIT]
- int fitParamGlobalK [MAX_FIT]
- int fitParamI [MAX_FIT]
- int fitParamK [MAX_FIT]
- int fitMode
- char robust
- int rescaleErrChisq
- double offset
- double offset_e
- double ∗∗ covar
- int calcShapiro
- int planetShapiro

- int jboFormat
- observation ∗ obsn
- int nobs
- int units
- int setUnits
- int tempo1
- int dilateFreq
- int timeEphemeris
- int t2cMethod
- int correctTroposphere
- int noWarnings
- char sorted
- char clock [16]
- char clockFromOverride [64]
- char JPL_EPHEMERIS [MAX_FILELEN]
- char ephemeris [MAX_FILELEN]
- int useCalceph
- storePrecision storePrec [MAX_STOREPRECISION]
- int nStorePrecision
- int bootStrap
- char tzrsite [100]
- double rmsPre
- double rmsPost
- char deleteFileName [100]
- int nits
- int ipm
- int swm
- double wave_sine [MAX_WHITE]
- double wave_sine_err [MAX_WHITE]
- double wave_cos [MAX_WHITE]
- double wave_cos_err [MAX_WHITE]
- double wave_sine_dm [MAX_WHITE]
- double wave_sine_dm_err [MAX_WHITE]
- double wave_cos_dm [MAX_WHITE]
- double wave_cos_dm_err [MAX_WHITE]
- int nWhite
- int nWhite_dm
- double waveScale
- double quad_aplus_r [MAX_QUAD]
- double quad_aplus_r_e [MAX_QUAD]
- double quad_aplus_i [MAX_QUAD]
- double quad_aplus_i_e [MAX_QUAD]
- double quad_across_r [MAX_QUAD]
- double quad_across_r_e [MAX_QUAD]
- double quad_across_i [MAX_QUAD]
- double quad_across_i_e [MAX_QUAD]
- double quadEpoch
- double quadRA
- double quadDEC
- int nQuad
- double ifuncT [MAX_IFUNC]
- double ifuncV [MAX_IFUNC]
- double ifuncE [MAX_IFUNC]
- double ifunc_weights [MAX_IFUNC]
- int ifuncN

- double clk_offsT [MAX_TEL_CLK_OFFS]
- double clk_offsV [MAX_TEL_CLK_OFFS]
- double clk_offsE [MAX_TEL_CLK_OFFS]
- int clkOffsN
- double quad_ifuncT_p [MAX_IFUNC]
- double quad_ifuncV_p [MAX_IFUNC]
- double quad_ifuncE_p [MAX_IFUNC]
- int quad_ifuncN_p
- double quad_ifuncT_c [MAX_IFUNC]
- double quad_ifuncV_c [MAX_IFUNC]
- double quad_ifuncE_c [MAX_IFUNC]
- int quad_ifuncN_c
- double quad_ifunc_p_RA
- double quad_ifunc_p_DEC
- double quad_ifunc_c_RA
- double quad_ifunc_c_DEC
- double quad_ifunc_geom_p
- double quad_ifunc_geom_c
- int nTelDX
- int setTelVelX
- double telDX_t [MAX_TEL_DX]
- double telDX_v [MAX_TEL_DX]
- double telDX_e [MAX_TEL_DX]
- double telDX_vel [MAX_TEL_DX]
- double telDX_vel_e [MAX_TEL_DX]
- int nTelDY
- int setTelVelY
- double telDY_t [MAX_TEL_DY]
- double telDY_v [MAX_TEL_DY]
- double telDY_e [MAX_TEL_DY]
- double telDY_vel [MAX_TEL_DY]
- double telDY_vel_e [MAX_TEL_DY]
- int nTelDZ
- int setTelVelZ
- double telDZ_v [MAX_TEL_DZ]
- double telDZ_t [MAX_TEL_DZ]
- double telDZ_e [MAX_TEL_DZ]
- double telDZ_vel [MAX_TEL_DZ]
- double telDZ_vel_e [MAX_TEL_DZ]
- int nT2efac
- int nT2equad
- char T2efacFlagID [MAX_T2EFAC][MAX_FLAG_LEN]
- char T2efacFlagVal [MAX_T2EFAC][MAX_FLAG_LEN]
- double T2efacVal [MAX_T2EFAC]
- char T2equadFlagID [MAX_T2EQUAD][MAX_FLAG_LEN]
- char T2equadFlagVal [MAX_T2EQUAD][MAX_FLAG_LEN]
- double T2equadVal [MAX_T2EQUAD]
- double T2globalEfac
- int nTNEF
- int nTNEQ
- int nTNSQ
- int nTNECORR
- char TNEFFlagID [MAX_TNEF][MAX_FLAG_LEN]
- char TNEFFlagVal [MAX_TNEF][MAX_FLAG_LEN]
- double TNEFVal [MAX_TNEF]

- double TNGlobalEF
- char TNEQFlagID [MAX_TNEQ][MAX_FLAG_LEN]
- char TNEQFlagVal [MAX_TNEQ][MAX_FLAG_LEN]
- double TNEQVal [MAX_TNEQ]
- double TNGlobalEQ
- double addTNGlobalEQ
- char TNSQFlagID [MAX_TNSQ][MAX_FLAG_LEN]
- char TNSQFlagVal [MAX_TNSQ][MAX_FLAG_LEN]
- double TNSQVal [MAX_TNSQ]
- char TNECORRFlagID [MAX_TNECORR][MAX_FLAG_LEN]
- char TNECORRFlagVal [MAX_TNECORR][MAX_FLAG_LEN]
- double TNECORRVal [MAX_TNECORR]
- double TNRedAmp
- double TNRedGam
- int TNRedC
- double TNRedCoeffs [200]
- double TNRedFLow
- double TNRedCorner
- double TNDMAmp
- double TNDMGam
- int TNDMC
- double TNDMCoeffs [200]
- int TNsubtractDM
- int TNsubtractRed
- int AverageResiduals
- char AverageFlag [MAX_FLAG_LEN]
- float AverageEpochWidth
- int outputTMatrix
- int useTNOrth
- double TNBandDMAmp
- double TNBandDMGam
- int TNBandDMC
- int nTNBandNoise
- double TNBandNoiseLF [MAX_TNBN]
- double TNBandNoiseHF [MAX_TNBN]
- double TNBandNoiseAmp [MAX_TNBN]
- double TNBandNoiseGam [MAX_TNBN]
- int TNBandNoiseC [MAX_TNBN]
- int nTNGroupNoise
- char TNGroupNoiseFlagID [MAX_TNGN][MAX_FLAG_LEN]
- char TNGroupNoiseFlagVal [MAX_TNGN][MAX_FLAG_LEN]
- double TNGroupNoiseAmp [MAX_TNGN]
- double TNGroupNoiseGam [MAX_TNGN]
- int TNGroupNoiseC [MAX_TNGN]
- int nDMEvents
- double TNDMEvStart [MAX_TNDMEv]
- double TNDMEvLength [MAX_TNDMEv]
- double TNDMEvAmp [MAX_TNDMEv]
- double TNDMEvGam [MAX_TNDMEv]
- int TNDMEvOff [MAX_TNDMEv]
- int TNDMEvLin [MAX_TNDMEv]
- int TNDMEvQuad [MAX_TNDMEv]
- int nTNShapeletEvents
- int TNShapeletEvN [MAX_TNDMEv]
- double TNShapeletEvPos [MAX_TNDMEv]

- double TNShapeletEvWidth [MAX_TNDMEv]
- double TNShapeletEvFScale [MAX_TNDMEv]
- char whiteNoiseModelFile [MAX_STRLEN]
- double rasim
- double decsim
- int simflag
- char fitFunc [MAX_FILELEN]
- int nconstraints
- enum constraint constraints [MAX_PARAMS]
- char auto_constraints
- FitInfo fitinfo

12.16.1 Detailed Description

contains the details for a single pulsar.

Includes an array of observations and parameters

12.16.2 Member Data Documentation

12.16.2.1 double pulsar::addTNGlobalEQ

12.16.2.2 char pulsar::auto_constraints

12.16.2.3 float pulsar::AverageEpochWidth

12.16.2.4 char pulsar::AverageFlag[MAX_FLAG_LEN]

12.16.2.5 int pulsar::AverageResiduals

12.16.2.6 char pulsar::binaryModel[100]

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

12.16.2.7 int pulsar::bootStrap

> 0 if calculating errors using bootstrap Monte-Carlo method

12.16.2.8 int pulsar::calcShapiro

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

12.16.2.9 double pulsar::cgw_angpol

12.16.2.10 double pulsar::cgw_cosinc

12.16.2.11 double pulsar::cgw_h0

12.16.2.12 double pulsar::cgw_mc

12.16.2.13 double pulsar::clk_offsE[MAX_TEL_CLK_OFFS]

12.16.2.14 double pulsar::clk_offsT[MAX_TEL_CLK_OFFS]

12.16.2.15 double pulsar::clk_offsV[MAX_TEL_CLK_OFFS]

12.16.2.16 int pulsar::clkOffsN

12.16.2.17 char pulsar::clock[16]

Clock standard to use as "UTC"

12.16.2.18 char pulsar::clockFromOverride[64]

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

12.16.2.19 enum constraint pulsar::constraints[MAX_PARAMS]

Which constraints are specified

12.16.2.20 int pulsar::correctTroposphere

whether or not do correct for tropospheric delay

12.16.2.21 double∗∗ pulsar::covar

12.16.2.22 char pulsar::decjStrPost[100]

String containing RAJ and DECJ (postfit)

12.16.2.23 char pulsar::decjStrPre[100]

String containing RAJ and DECJ (prefit)

12.16.2.24 double pulsar::decsim

12.16.2.25 char pulsar::deleteFileName[100]

File name containing deleted points

12.16.2.26 int pulsar::dilateFreq

whether or not to apply SS time dilation to RFs

12.16.2.27 double pulsar::dmoffsCM[MAX_IFUNC]

12.16.2.28 double pulsar::dmoffsCM_error[MAX_IFUNC]

12.16.2.29 double pulsar::dmoffsCM_mjd[MAX_IFUNC]

12.16.2.30 double pulsar::dmoffsCM_weight[MAX_IFUNC]

12.16.2.31 int pulsar::dmoffsCMnum

12.16.2.32 double pulsar::dmoffsDM[MAX_IFUNC]

12.16.2.33 double pulsar::dmoffsDM_error[MAX_IFUNC]

12.16.2.34 double pulsar::dmoffsDM_mjd[MAX_IFUNC]

12.16.2.35 double pulsar::dmoffsDM_weight[MAX_IFUNC]

12.16.2.36 int pulsar::dmoffsDMnum

12.16.2.37 double pulsar::dmOffset

Value to add to DM flags

12.16.2.38 int pulsar::eclCoord

= 1 for ecliptic coords otherwise celestial coords

12.16.2.39 char pulsar::eopc04_file[MAX_FILELEN]

12.16.2.40 char pulsar::ephemeris[MAX_FILELEN]

12.16.2.41 char pulsar::filterStr[MAX_STRLEN]

String describing filters

12.16.2.42 double pulsar::fitChisq

Chisq value from the fit

12.16.2.43 char pulsar::fitFunc[MAX_FILELEN]

12.16.2.44 FitInfo pulsar::fitinfo

12.16.2.45 int pulsar::fitJump[MAX_JUMPS]

= 1 if fit for jump

12.16.2.46 int pulsar::fitMode

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

12.16.2.47 int pulsar::fitNfree

Number of degrees of freedom in fit

12.16.2.48 int pulsar::fitParamGlobalI[MAX_FIT]

12.16.2.49 int pulsar::fitParamGlobalK[MAX_FIT]

12.16.2.50 int pulsar::fitParamI[MAX_FIT]

12.16.2.51 int pulsar::fitParamK[MAX_FIT]

12.16.2.52 int pulsar::fixedFormat

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

12.16.2.53 char pulsar::fjumpID[16]

12.16.2.54 int pulsar::globalNfit

Total number of parameters in the fit

12.16.2.55 int pulsar::globalNoConstrain

Total number of points without constraints

12.16.2.56 double pulsar::gwb_decj

12.16.2.57 double pulsar::gwb_epoch

12.16.2.58 double pulsar::gwb_geom_c

12.16.2.59 double pulsar::gwb_geom_p

12.16.2.60 double pulsar::gwb_raj

12.16.2.61 double pulsar::gwb_width

12.16.2.62 double pulsar::gwecc_dec

12.16.2.63 double pulsar::gwecc_distance

12.16.2.64 double pulsar::gwecc_e

12.16.2.65 double pulsar::gwecc_epoch

12.16.2.66 double pulsar::gwecc_inc

12.16.2.67 double pulsar::gwecc_m1

12.16.2.68 double pulsar::gwecc_m2

12.16.2.69 double pulsar::gwecc_nodes_orientation

12.16.2.70 double pulsar::gwecc_orbital_period

12.16.2.71 double pulsar::gwecc_psrdist

12.16.2.72 int pulsar::gwecc_pulsarTermOn

12.16.2.73 double pulsar::gwecc_ra

12.16.2.74 double pulsar::gwecc_redshift

12.16.2.75 double pulsar::gwecc_theta_0

12.16.2.76 double pulsar::gwecc_theta_nodes

12.16.2.77 double pulsar::gwm_decj

12.16.2.78 double pulsar::gwm_dphase

12.16.2.79 double pulsar::gwm_epoch

12.16.2.80 double pulsar::gwm_phi

12.16.2.81 double pulsar::gwm_raj

12.16.2.82 double pulsar::gwsrc_across_i

12.16.2.83 double pulsar::gwsrc_across_i_e

12.16.2.84 double pulsar::gwsrc_across_r

12.16.2.85 double pulsar::gwsrc_across_r_e

12.16.2.86 double pulsar::gwsrc_aplus_i

12.16.2.87 double pulsar::gwsrc_aplus_i_e

12.16.2.88 double pulsar::gwsrc_aplus_r

12.16.2.89 double pulsar::gwsrc_aplus_r_e

12.16.2.90 double pulsar::gwsrc_dec

12.16.2.91 double pulsar::gwsrc_epoch

12.16.2.92 double pulsar::gwsrc_psrdist

12.16.2.93 double pulsar::gwsrc_ra

12.16.2.94 double pulsar::ifunc_weights[MAX_IFUNC]

12.16.2.95 double pulsar::ifuncE[MAX_IFUNC]

12.16.2.96 int pulsar::ifuncN

12.16.2.97 double pulsar::ifuncT[MAX_IFUNC]

12.16.2.98 double pulsar::ifuncV[MAX_IFUNC]

12.16.2.99 int pulsar::ipm

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

12.16.2.100 int pulsar::jboFormat

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

12.16.2.101 char pulsar::JPL_EPHEMERIS[MAX_FILELEN]

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

String describing jump

12.16.2.103 double pulsar::jumpVal[MAX_JUMPS]

Value of jump

12.16.2.104 double pulsar::jumpValErr[MAX_JUMPS]

Error on jump

12.16.2.105 char pulsar::name[100]

12.16.2.106 int pulsar::nCompanion

Number of binary companions

12.16.2.107 int pulsar::nconstraints

Number of fit constraints specified

12.16.2.108 int pulsar::nDMEvents

12.16.2.109 int pulsar::ndmx

Number of DM steps

12.16.2.110 double pulsar::ne_sw

Electron density at 1AU due to the solar wind

12.16.2.111 int pulsar::nFit

Number of points in the fit

12.16.2.112 int pulsar::nGlobal

Number of global parameters in the fit

12.16.2.113 int pulsar::nits

Number of iterations for the fit

12.16.2.114 int pulsar::nJumps

Number of jumps

12.16.2.115 int pulsar::nobs

Number of observations in .tim file

12.16.2.116 int pulsar::noWarnings

= 1, do not display warning messages

12.16.2.117 int pulsar::nParam

Number of parameters in the fit

12.16.2.118 int pulsar::nPhaseJump

Number of phase jumps

12.16.2.119 int pulsar::nQuad

12.16.2.120 int pulsar::nStorePrecision

12.16.2.121 int pulsar::nT2efac

12.16.2.122 int pulsar::nT2equad

12.16.2.123 int pulsar::nTelDX

12.16.2.124 int pulsar::nTelDY

12.16.2.125 int pulsar::nTelDZ

12.16.2.126 int pulsar::nTNBandNoise

12.16.2.127 int pulsar::nTNECORR

12.16.2.128 int pulsar::nTNEF

12.16.2.129 int pulsar::nTNEQ

12.16.2.130 int pulsar::nTNGroupNoise

12.16.2.131 int pulsar::nTNShapeletEvents

12.16.2.132 int pulsar::nTNSQ

12.16.2.133 int pulsar::nToffset

12.16.2.134 int pulsar::nWhite

12.16.2.135 int pulsar::nWhite_dm

12.16.2.136 observation∗ pulsar::obsn

[MAX_OBSN_VAL];

12.16.2.137 double pulsar::offset

Offset, always fitted for

12.16.2.138 double pulsar::offset_e

Error in the offset

12.16.2.139 int pulsar::outputTMatrix

12.16.2.140 parameter pulsar::param[MAX_PARAMS]

12.16.2.141 char pulsar::passStr[MAX_STRLEN]

String describing filters

12.16.2.142 longdouble pulsar::phaseJump[MAX_JUMPS]

Time of phase jump

12.16.2.143 int pulsar::phaseJumpDir[MAX_JUMPS]

Size and direction of phase jump

12.16.2.144 int pulsar::phaseJumpID[MAX_JUMPS]

ID of closest point to the phase jump

12.16.2.145 int pulsar::planetShapiro

= 1 if included otherwise 0

12.16.2.146 double pulsar::posPulsar[3]

3-vector pointing at pulsar

12.16.2.147 double pulsar::quad_across_i[MAX_QUAD]

12.16.2.148 double pulsar::quad_across_i_e[MAX_QUAD]

12.16.2.149 double pulsar::quad_across_r[MAX_QUAD]

12.16.2.150 double pulsar::quad_across_r_e[MAX_QUAD]

12.16.2.151 double pulsar::quad_aplus_i[MAX_QUAD]

12.16.2.152 double pulsar::quad_aplus_i_e[MAX_QUAD]

12.16.2.153 double pulsar::quad_aplus_r[MAX_QUAD]

12.16.2.154 double pulsar::quad_aplus_r_e[MAX_QUAD]

12.16.2.155 double pulsar::quad_ifunc_c_DEC

12.16.2.156 double pulsar::quad_ifunc_c_RA

12.16.2.157 double pulsar::quad_ifunc_geom_c

12.16.2.158 double pulsar::quad_ifunc_geom_p

12.16.2.159 double pulsar::quad_ifunc_p_DEC

12.16.2.160 double pulsar::quad_ifunc_p_RA

12.16.2.161 double pulsar::quad_ifuncE_c[MAX_IFUNC]

12.16.2.162 double pulsar::quad_ifuncE_p[MAX_IFUNC]

12.16.2.163 int pulsar::quad_ifuncN_c

12.16.2.164 int pulsar::quad_ifuncN_p

12.16.2.165 double pulsar::quad_ifuncT_c[MAX_IFUNC]

12.16.2.166 double pulsar::quad_ifuncT_p[MAX_IFUNC]

12.16.2.167 double pulsar::quad_ifuncV_c[MAX_IFUNC]

12.16.2.168 double pulsar::quad_ifuncV_p[MAX_IFUNC]

12.16.2.169 double pulsar::quadDEC

12.16.2.170 double pulsar::quadEpoch

12.16.2.171 double pulsar::quadRA

12.16.2.172 char pulsar::rajStrPost[100]

12.16.2.173 char pulsar::rajStrPre[100]

12.16.2.174 double pulsar::rasim

12.16.2.175 int pulsar::rescaleErrChisq

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

12.16.2.176 double pulsar::rmsPost

12.16.2.177 double pulsar::rmsPre

12.16.2.178 char pulsar::robust

12.16.2.179 int pulsar::setTelVelX

12.16.2.180 int pulsar::setTelVelY

12.16.2.181 int pulsar::setTelVelZ

12.16.2.182 int pulsar::setUnits

12.16.2.183 int pulsar::simflag

Which fit function are we using

12.16.2.184 char pulsar::sorted

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

12.16.2.185 storePrecision pulsar::storePrec[MAX_STOREPRECISION]

12.16.2.186 int pulsar::swm

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

12.16.2.187 int pulsar::t2cMethod

How to transform from terrestrial to celestial coords

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

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

12.16.2.190 double pulsar::T2efacVal[MAX_T2EFAC]

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

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

12.16.2.193 double pulsar::T2equadVal[MAX_T2EQUAD]

12.16.2.194 double pulsar::T2globalEfac

12.16.2.195 double pulsar::telDX_e[MAX_TEL_DX]

12.16.2.196 double pulsar::telDX_t[MAX_TEL_DX]

12.16.2.197 double pulsar::telDX_v[MAX_TEL_DX]

12.16.2.198 double pulsar::telDX_vel[MAX_TEL_DX]

12.16.2.199 double pulsar::telDX_vel_e[MAX_TEL_DX]

12.16.2.200 double pulsar::telDY_e[MAX_TEL_DY]

12.16.2.201 double pulsar::telDY_t[MAX_TEL_DY]

12.16.2.202 double pulsar::telDY_v[MAX_TEL_DY]

12.16.2.203 double pulsar::telDY_vel[MAX_TEL_DY]

12.16.2.204 double pulsar::telDY_vel_e[MAX_TEL_DY]

12.16.2.205 double pulsar::telDZ_e[MAX_TEL_DZ]

12.16.2.206 double pulsar::telDZ_t[MAX_TEL_DZ]

12.16.2.207 double pulsar::telDZ_v[MAX_TEL_DZ]

12.16.2.208 double pulsar::telDZ_vel[MAX_TEL_DZ]

12.16.2.209 double pulsar::telDZ_vel_e[MAX_TEL_DZ]

12.16.2.210 int pulsar::tempo1

= 1 if tempo1 is emulated

12.16.2.211 int pulsar::timeEphemeris

Which code to use for Einstein delay

12.16.2.212 double pulsar::TNBandDMAmp

12.16.2.213 int pulsar::TNBandDMC

12.16.2.214 double pulsar::TNBandDMGam

12.16.2.215 double pulsar::TNBandNoiseAmp[MAX_TNBN]

12.16.2.216 int pulsar::TNBandNoiseC[MAX_TNBN]

12.16.2.217 double pulsar::TNBandNoiseGam[MAX_TNBN]

12.16.2.218 double pulsar::TNBandNoiseHF[MAX_TNBN]

12.16.2.219 double pulsar::TNBandNoiseLF[MAX_TNBN]

12.16.2.220 double pulsar::TNDMAmp

12.16.2.221 int pulsar::TNDMC

12.16.2.222 double pulsar::TNDMCoeffs[200]

12.16.2.223 double pulsar::TNDMEvAmp[MAX_TNDMEv]

12.16.2.224 double pulsar::TNDMEvGam[MAX_TNDMEv]

12.16.2.225 double pulsar::TNDMEvLength[MAX_TNDMEv]

12.16.2.226 int pulsar::TNDMEvLin[MAX_TNDMEv]

12.16.2.227 int pulsar::TNDMEvOff[MAX_TNDMEv]

12.16.2.228 int pulsar::TNDMEvQuad[MAX_TNDMEv]

12.16.2.229 double pulsar::TNDMEvStart[MAX_TNDMEv]

12.16.2.230 double pulsar::TNDMGam

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

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

12.16.2.233 double pulsar::TNECORRVal[MAX_TNECORR]

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

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

12.16.2.236 double pulsar::TNEFVal[MAX_TNEF]

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

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

12.16.2.239 double pulsar::TNEQVal[MAX_TNEQ]

12.16.2.240 double pulsar::TNGlobalEF

12.16.2.241 double pulsar::TNGlobalEQ

12.16.2.242 double pulsar::TNGroupNoiseAmp[MAX_TNGN]

12.16.2.243 int pulsar::TNGroupNoiseC[MAX_TNGN]

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

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

12.16.2.246 double pulsar::TNGroupNoiseGam[MAX_TNGN]

12.16.2.247 double pulsar::TNRedAmp

12.16.2.248 int pulsar::TNRedC

12.16.2.249 double pulsar::TNRedCoeffs[200]

12.16.2.250 double pulsar::TNRedCorner

12.16.2.251 double pulsar::TNRedFLow

12.16.2.252 double pulsar::TNRedGam

12.16.2.253 double pulsar::TNShapeletEvFScale[MAX_TNDMEv]

12.16.2.254 int pulsar::TNShapeletEvN[MAX_TNDMEv]

12.16.2.255 double pulsar::TNShapeletEvPos[MAX_TNDMEv]

12.16.2.256 double pulsar::TNShapeletEvWidth[MAX_TNDMEv]

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

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

12.16.2.259 double pulsar::TNSQVal[MAX_TNSQ]

12.16.2.260 int pulsar::TNsubtractDM

12.16.2.261 int pulsar::TNsubtractRed

12.16.2.262 double∗∗ pulsar::ToAextraCovar

12.16.2.263 double pulsar::tOffset[MAX_TOFFSET]

Offsets in TOAs in seconds

12.16.2.264 double pulsar::tOffset_f1[MAX_TOFFSET]

12.16.2.265 double pulsar::tOffset_f2[MAX_TOFFSET]

Range for offset to be applied

12.16.2.266 double pulsar::tOffset_t1[MAX_TOFFSET]

12.16.2.267 double pulsar::tOffset_t2[MAX_TOFFSET]

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

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

12.16.2.270 char pulsar::tzrsite[100]

Site-code for polyco

12.16.2.271 int pulsar::units

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

12.16.2.272 int pulsar::useCalceph

12.16.2.273 int pulsar::useTNOrth

12.16.2.274 double pulsar::velPulsar[3]

3-vector giving pulsar's velocity

12.16.2.275 double pulsar::wave_cos[MAX_WHITE]

12.16.2.276 double pulsar::wave_cos_dm[MAX_WHITE]

12.16.2.277 double pulsar::wave_cos_dm_err[MAX_WHITE]

12.16.2.278 double pulsar::wave_cos_err[MAX_WHITE]

12.16.2.279 double pulsar::wave_sine[MAX_WHITE]

12.16.2.280 double pulsar::wave_sine_dm[MAX_WHITE]

12.16.2.281 double pulsar::wave_sine_dm_err[MAX_WHITE]

12.16.2.282 double pulsar::wave_sine_err[MAX_WHITE]

12.16.2.283 double pulsar::waveScale

12.16.2.284 char pulsar::whiteNoiseModelFile[MAX_STRLEN]

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

- tempo2.h

12.17 storePrecision Struct Reference

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

Public Attributes

- longdouble minPrec
- char routine [100]
- char comment [MAX_STRLEN]

12.17.1 Member Data Documentation

12.17.1.1 char storePrecision::comment[MAX_STRLEN]

12.17.1.2 longdouble storePrecision::minPrec

12.17.1.3 char storePrecision::routine[100]

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

- tempo2.h

12.18 T1Polyco Struct Reference

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

Public Attributes

- char psrname [64]
- char date_string [10]
- char utc_string [13]

- long double mjd_mid
- double dm
- double doppler
- double log10rms
- long double reference_phase
- long double frequency_psr_0
- char sitename [5]
- int span
- int ncoeff
- double frequency_obs
- double binary_phase
- double binary_frequency
- long double coeff [32]

12.18.1 Member Data Documentation

12.18.1.1 double T1Polyco::binary_frequency

12.18.1.2 double T1Polyco::binary_phase

12.18.1.3 long double T1Polyco::coeff[32]

12.18.1.4 char T1Polyco::date_string[10]

12.18.1.5 double T1Polyco::dm

12.18.1.6 double T1Polyco::doppler

12.18.1.7 double T1Polyco::frequency_obs

12.18.1.8 long double T1Polyco::frequency_psr_0

12.18.1.9 double T1Polyco::log10rms

12.18.1.10 long double T1Polyco::mjd_mid

12.18.1.11 int T1Polyco::ncoeff

12.18.1.12 char T1Polyco::psrname[64]

12.18.1.13 long double T1Polyco::reference_phase

12.18.1.14 char T1Polyco::sitename[5]

12.18.1.15 int T1Polyco::span

12.18.1.16 char T1Polyco::utc_string[13]

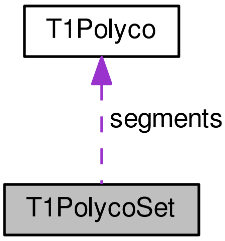
The documentation for this struct was generated from the following file:

- tempo2pred.h

12.19 T1PolycoSet Struct Reference

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

Collaboration diagram for T1PolycoSet:



Public Attributes

- T1Polyco ∗ segments
- int nsegments

12.19.1 Member Data Documentation

12.19.1.1 int T1PolycoSet::nsegments

12.19.1.2 T1Polyco∗ T1PolycoSet::segments

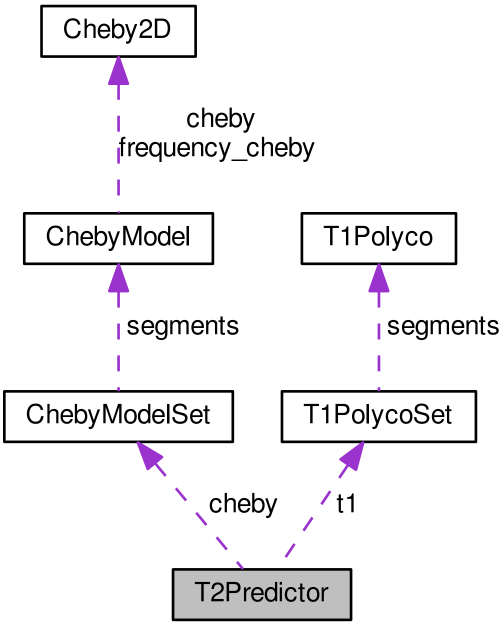
The documentation for this struct was generated from the following file:

- tempo2pred.h

12.20 T2Predictor Struct Reference

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

Collaboration diagram for T2Predictor:



Public Attributes

- T2PredictorKind kind
- union {
  ChebyModelSet cheby
  T1PolycoSet t1
  } modelset

12.20.1 Member Data Documentation

12.20.1.1 ChebyModelSet T2Predictor::cheby

12.20.1.2 T2PredictorKind T2Predictor::kind

12.20.1.3 union { ... } T2Predictor::modelset

12.20.1.4 T1PolycoSet T2Predictor::t1

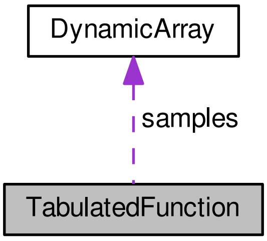
The documentation for this struct was generated from the following file:

- tempo2pred.h

12.21 TabulatedFunction Struct Reference

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

Collaboration diagram for TabulatedFunction:



Public Attributes

- char fileName [256]
- char header_line [256]
- DynamicArray samples

12.21.1 Member Data Documentation

12.21.1.1 char TabulatedFunction::fileName[256]

12.21.1.2 char TabulatedFunction::header_line[256]

12.21.1.3 DynamicArray TabulatedFunction::samples

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

- tabulatedfunction.h

12.22 TabulatedFunctionSample Struct Reference

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

Public Attributes

- double x
- double y

12.22.1 Member Data Documentation

12.22.1.1 double TabulatedFunctionSample::x

12.22.1.2 double TabulatedFunctionSample::y

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

- tabulatedfunction.h

Chapter 13

File Documentation

13.1 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)

13.1.1 Function Documentation

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

13.1.1.2 void cholesky_dmModel (double ∗∗ *m,* double *D,* double *d,* double *ref_freq,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

13.1.1.3 void cholesky_dmModelCovarParam (double ∗∗ *m,* double *alpha,* double *a,* double *b,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

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

13.1.1.5 int cholesky_formUinv (double ∗∗ *uinv,* double ∗∗ *m,* int *np*)

13.1.1.6 void cholesky_powerlawModel (double ∗∗ *m,* double *modelAlpha,* double *modelFc,* double *modelA,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

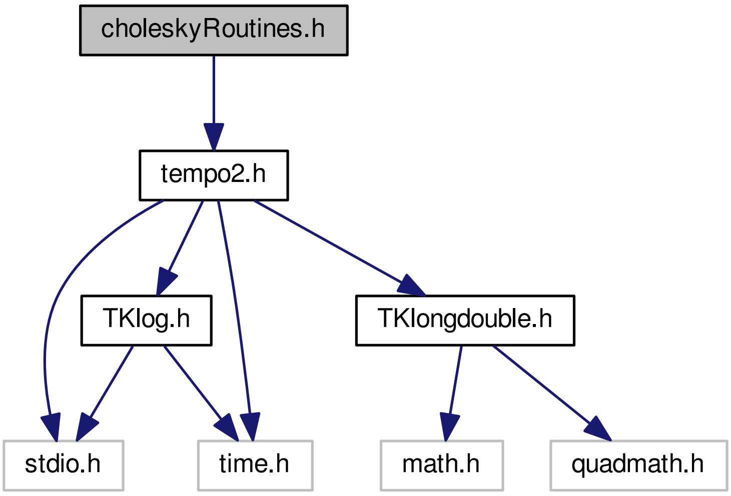
13.1.1.7 void cholesky_powerlawModel_withBeta (double ∗∗ *m,* double *modelAlpha,* double *beta,* double *modelFc,* double *modelA,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

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

13.2 choleskyRoutines.h File Reference

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

Include dependency graph for choleskyRoutines.h:



Functions

- void T2writeCovarFuncModel (double alpha, double fc, double val, double white, char ∗fname)
- void T2get_covFunc_automatic (pulsar ∗psr, double expSmooth, char ∗outname, double ∗fc_w, double ∗fc←_r, double ∗modelAlpha_out, double ∗modelVal, double ∗whiteNoiseLevel, int realflag, int dcmflag)
- void T2cubicFit (double ∗resx, double ∗resy, double ∗rese, int nres, double ∗cubicVal, double ∗cubicErr)
- void T2findSmoothCurve (double ∗resx, double ∗resy, double ∗rese, int nres, double ∗cubicVal, double ∗smoothModel, double expSmooth)
- void T2interpolate (double ∗resx, double ∗resy, double ∗rese, int nres, double ∗cubicVal, double ∗interpX, double ∗interpY, int ∗nInterp, int interpTime, double expSmooth)
- void T2getHighFreqRes (double ∗resy, double ∗smoothModel, int nres, double ∗highFreqRes)
- int T2calculateSpectra (double ∗x, double ∗y, double ∗e, int n, int useErr, int preWhite, int specType, double ∗specX, double ∗specY)
- int T2fitSpectra (double ∗preWhiteSpecX, double ∗preWhiteSpecY, int nPreWhiteSpec, double ∗modelAlpha, double ∗modelFc, int ∗modelNfit, double ∗modelScale, double ∗fitVar, int aval, int ipw, double ifc, double iexp, int inpt, double amp, int useBeta, double ∗betaVal, double cutoff=0.0)
- void T2calculateCholesky (double modelAlpha, double modelFc, double modelScale, double fitVar, double ∗∗uinv, double ∗covFunc, double ∗resx, double ∗resy, double ∗rese, int np, double ∗highFreqRes, double ∗errorScaleFactor, int dcmflag, int useBeta, double betaVal)
- int T2calculateCovarFunc (double modelAlpha, double modelFc, double modelA, int useBeta, double betaVal, double ∗covFunc, double ∗resx, double ∗resy, double ∗rese, int np)
- void T2getWhiteRes (double ∗resx, double ∗resy, double ∗rese, int nres, double ∗∗uinv, double ∗cholWhiteY)
- void T2calculateDailyCovariance (double ∗x, double ∗y, double ∗e, int n, double ∗cv, int ∗in, double ∗zl, int usewt)

- int T2obtainTimingResiduals (pulsar ∗psr, double ∗resx, double ∗resy, double ∗rese)
- int T2guess_vals (double ∗x, double ∗y, int n, double ∗alpha, double ∗amp, double ∗fc, int ∗nfit, double wn, double ∗fc_white, int prewhite)
- void T2getWhiteNoiseLevel (int n, double ∗y, int nlast, double ∗av)
- void T2cholDecomposition (double ∗∗a, int n, double ∗p)

Variables

- double FCALPHA
- double WNLEVEL
- double EXPSMOOTH
- double UPW
- double NFIT
- double FCFINAL

13.2.1 Function Documentation

13.2.1.1 void T2calculateCholesky (double *modelAlpha*, double *modelFc*, double *modelScale*, double *fitVar*, double ∗∗ *uinv*, double ∗ *covFunc*, double ∗ *resx*, double ∗ *resy*, double ∗ *rese*, int *np*, double ∗ *highFreqRes*, double ∗ *errorScaleFactor*, int *dcmflag*, int *useBeta*, double *betaVal*)

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

13.2.1.3 void T2calculateDailyCovariance (double ∗ *x*, double ∗ *y*, double ∗ *e*, int *n*, double ∗ *cv*, int ∗ *in*, double ∗ *zl*, int *usewt*)

13.2.1.4 int T2calculateSpectra (double ∗ *x*, double ∗ *y*, double ∗ *e*, int *n*, int *useErr*, int *preWhite*, int *specType*, double ∗ *specX*, double ∗ *specY*)

13.2.1.5 void T2cholDecomposition (double ∗∗ *a*, int *n*, double ∗ *p*)

13.2.1.6 void T2cubicFit (double ∗ *resx*, double ∗ *resy*, double ∗ *rese*, int *nres*, double ∗ *cubicVal*, double ∗ *cubicErr*)

13.2.1.7 void T2findSmoothCurve (double ∗ *resx*, double ∗ *resy*, double ∗ *rese*, int *nres*, double ∗ *cubicVal*, double ∗ *smoothModel*, double *expSmooth*)

13.2.1.8 int T2fitSpectra (double ∗ *preWhiteSpecX*, double ∗ *preWhiteSpecY*, int *nPreWhiteSpec*, double ∗ *modelAlpha*, double ∗ *modelFc*, int ∗ *modelNfit*, double ∗ *modelScale*, double ∗ *fitVar*, int *aval*, int *ipw*, double *ifc*, double *iexp*, int *inpt*, double *amp*, int *useBeta*, double ∗ *betaVal*, double *cutoff* = `0.0`)

13.2.1.9 void T2get_covFunc_automatic (pulsar ∗ *psr*, double *expSmooth*, char ∗ *outname*, double ∗ *fc_w*, double ∗ *fc_r*, double ∗ *modelAlpha_out*, double ∗ *modelVal*, double ∗ *whiteNoiseLevel*, int *realflag*, int *dcmflag*)

13.2.1.10 void T2getHighFreqRes (double ∗ *resy*, double ∗ *smoothModel*, int *nres*, double ∗ *highFreqRes*)

13.2.1.11 void T2getWhiteNoiseLevel (int *n*, double ∗ *y*, int *nlast*, double ∗ *av*)

13.2.1.12 void T2getWhiteRes (double ∗ *resx*, double ∗ *resy*, double ∗ *rese*, int *nres*, double ∗∗ *uinv*, double ∗ *cholWhiteY*)

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

13.2.1.14 void T2interpolate (double ∗ *resx*, double ∗ *resy*, double ∗ *rese*, int *nres*, double ∗ *cubicVal*, double ∗ *interpX*, double ∗ *interpY*, int ∗ *nInterp*, int *interpTime*, double *expSmooth*)

13.2.1.15 int T2obtainTimingResiduals (pulsar ∗ *psr,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese*)

13.2.1.16 void T2writeCovarFuncModel (double *alpha,* double *fc,* double *val,* double *white,* char ∗ *fname*)

13.2.2 Variable Documentation

13.2.2.1 double EXPSMOOTH

13.2.2.2 double FCALPHA

13.2.2.3 double FCFINAL

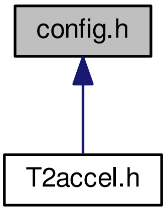
13.2.2.4 double NFIT

13.2.2.5 double UPW

13.2.2.6 double WNLEVEL

13.3 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_DLERROR 1
- #define HAVE_DLFCN_H 1
- #define HAVE_FFTW3 1
- #define HAVE_INTTYPES_H 1
- #define HAVE_LAPACK 1
- #define HAVE_LIBDL 1
- #define HAVE_LIBDLLOADER 1
- #define HAVE_LIBM 1
- #define HAVE_MEMORY_H 1
- #define HAVE_PGPLOT 1
- #define HAVE_PTHREAD 1
- #define HAVE_STDINT_H 1
- #define HAVE_STDLIB_H 1

- #define HAVE_STRINGS_H 1
- #define HAVE_STRING_H 1
- #define HAVE_SYS_STAT_H 1
- #define HAVE_SYS_TYPES_H 1
- #define HAVE_UNISTD_H 1
- #define LT_OBJDIR ".libs/"
- #define PACKAGE "tempo2"
- #define PACKAGE_BUGREPORT "george.hobbs@csiro.au"
- #define PACKAGE_NAME "Tempo2"
- #define PACKAGE_STRING "Tempo2 2015.09.0"
- #define PACKAGE_TARNAME "tempo2"
- #define PACKAGE_URL "http://www.bitbucket.org/mkeith/tempo2"
- #define PACKAGE_VERSION "2015.09.0"
- #define STDC_HEADERS 1
- #define TEMPO2_ARCH "linux-gnu"
- #define VERSION "2015.09.0"
- #define X_DISPLAY_MISSING 1
- #define _DARWIN_USE_64_BIT_INODE 1

13.3.1 Macro Definition Documentation

13.3.1.1 #define _DARWIN_USE_64_BIT_INODE 1

13.3.1.2 #define F77_FUNC(*name,* *NAME*) name ## _

13.3.1.3 #define F77_FUNC_(*name,* *NAME*) name ## _

13.3.1.4 #define HAVE_BLAS 1

13.3.1.5 #define HAVE_DLERROR 1

13.3.1.6 #define HAVE_DLFCN_H 1

13.3.1.7 #define HAVE_FFTW3 1

13.3.1.8 #define HAVE_INTTYPES_H 1

13.3.1.9 #define HAVE_LAPACK 1

13.3.1.10 #define HAVE_LIBDL 1

13.3.1.11 #define HAVE_LIBDLLOADER 1

13.3.1.12 #define HAVE_LIBM 1

13.3.1.13 #define HAVE_MEMORY_H 1

13.3.1.14 #define HAVE_PGPLOT 1

13.3.1.15 #define HAVE_PTHREAD 1

13.3.1.16 #define HAVE_STDINT_H 1

13.3.1.17 #define HAVE_STDLIB_H 1

13.3.1.18 #define HAVE_STRING_H 1

13.3.1.19 #define HAVE_STRINGS_H 1

13.3.1.20 #define HAVE_SYS_STAT_H 1

13.3.1.21 #define HAVE_SYS_TYPES_H 1

13.3.1.22 #define HAVE_UNISTD_H 1

13.3.1.23 #define LT_OBJDIR ".libs/"

13.3.1.24 #define PACKAGE "tempo2"

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

13.3.1.26 #define PACKAGE_NAME "Tempo2"

13.3.1.27 #define PACKAGE_STRING "Tempo2 2015.09.0"

13.3.1.28 #define PACKAGE_TARNAME "tempo2"

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

13.3.1.30 #define PACKAGE_VERSION "2015.09.0"

13.3.1.31 #define STDC_HEADERS 1

13.3.1.32 #define TEMPO2_ARCH "linux-gnu"

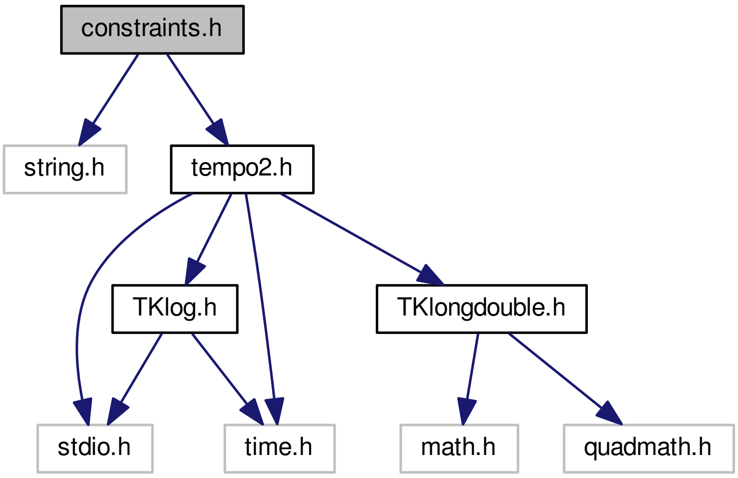
13.3.1.33 #define VERSION "2015.09.0"

13.3.1.34 #define X_DISPLAY_MISSING 1

13.4 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)

13.4.1 Function Documentation

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

13.4.1.2 void computeConstraintWeights (**pulsar** ∗ *psr*)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

13.4.1.17 std::string get_constraint_name (enum constraint *c*)

13.4.1.18 double standardConstraintFunctions (pulsar ∗ *psr,* int *ipsr,* int *iconstraint,* int *iparam,* int *constraintk,* int *k*)

13.5 documentation/1_USER_GUIDE.md File Reference

13.6 documentation/2_developers.md File Reference

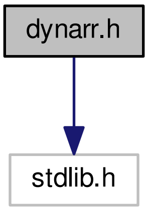
13.7 documentation/3_DEVELOPER_GUIDE.md File Reference

13.8 documentation/4_directories.md File Reference

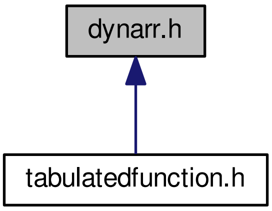
13.9 dynarr.h File Reference

```
#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 ∗)

13.9.1 Function Documentation

13.9.1.1 void DynamicArray_free (**DynamicArray** ∗)

13.9.1.2 void DynamicArray_init (**DynamicArray** ∗ , size_t *elemSize*)

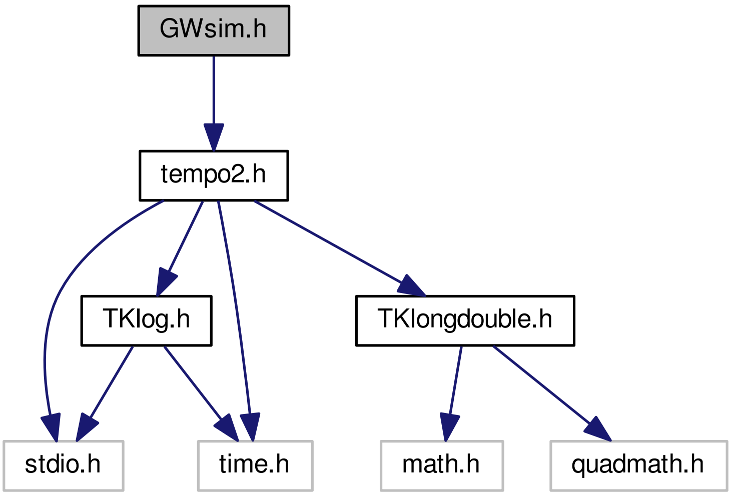
13.9.1.3 void∗ DynamicArray_push_back (**DynamicArray** ∗ , void ∗ *elem*)

13.9.1.4 void DynamicArray_resize (**DynamicArray** ∗ , size_t *nelem*)

13.10 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)

13.10.1 Typedef Documentation

13.10.1.1 typedef struct **gwgeneralSrc gwgeneralSrc**

13.10.1.2 typedef struct **gwgenSpec gwgenSpec**

13.10.1.3 typedef struct **gwSrc gwSrc**

13.10.2 Function Documentation

13.10.2.1 **longdouble** calculateResidualgeneralGW (**longdouble** ∗ *kp,* **gwgeneralSrc** ∗ *gw,* **longdouble** *time,* **longdouble** *dist*)

13.10.2.2 **longdouble** calculateResidualGW (**longdouble** ∗ *kp,* **gwSrc** ∗ *gw,* **longdouble** *time,* **longdouble** *dist*)

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

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

13.10.2.5 **longdouble** dotProduct (**longdouble** ∗ *m1,* **longdouble** ∗ *m2*)

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

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

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

13.10.2.9 double Fe (double *ec*)

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

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

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

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

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

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

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

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

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

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

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

13.10.2.21 double psrangle (double *centre_long,* double *centre_lat,* double *psr_long,* double *psr_lat*)

13.10.2.22 double Rs (double *m1*)

13.10.2.23 void setupgeneralGW (gwgeneralSrc * *gw*)

13.10.2.24 void setupGW (gwSrc * *gw*)

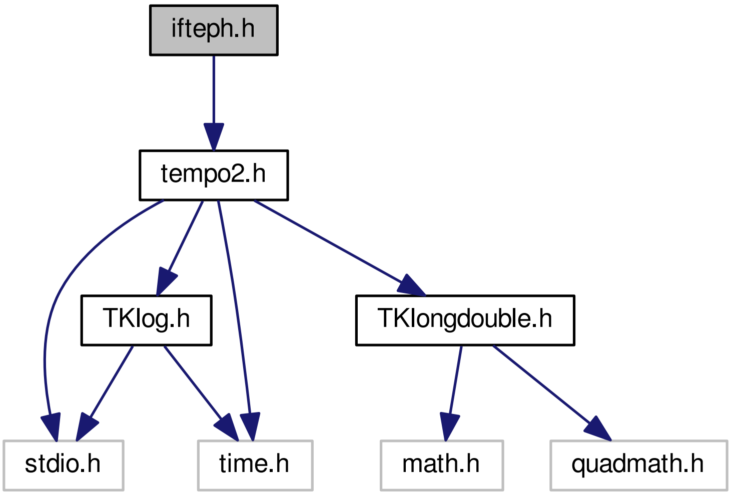
13.10.2.25 void setupPulsar_GWsim (longdouble *ra_p,* longdouble *dec_p,* longdouble * *kp*)

13.10.2.26 double sphharm (int *l,* int *m,* double *x*)

13.11 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)

13.11.1 Macro Definition Documentation

13.11.1.1 #define IFTE_JD0 2443144.5003725 /∗ Epoch of TCB, TCG and TT ∗/

13.11.1.2 #define IFTE_K (((longdouble)1.0) + ((longdouble)IFTE_KM1)) /∗ needs quad precision ∗/

13.11.1.3 #define IFTE_KM1 1.55051979176e-8

13.11.1.4 #define IFTE_LC 1.48082686742e-8

13.11.1.5 #define IFTE_MJD0 43144.0003725

13.11.1.6 #define IFTE_TEPH0 -65.564518e-6

13.11.2 Function Documentation

13.11.2.1 void IFTE_close_file ( )

13.11.2.2 double IFTE_DeltaT (double *Teph0*, double *Teph1*)

13.11.2.3 double IFTE_DeltaTDot (double *Teph0*, double *Teph1*)

13.11.2.4 void IFTE_get_DeltaT_DeltaTDot (double *Teph0*, double *Teph1*, double ∗ *DeltaT*, double ∗ *DeltaTDot*)

13.11.2.5 void IFTE_get_vE (double *Teph0*, double *Teph1*, double ∗ *vE*)

13.11.2.6 void IFTE_get_vE_vEDot (double *Teph0*, double *Teph1*, double ∗ *ve*, double ∗ *vEDot*)

13.11.2.7 void IFTE_get_vEDot (double *Teph0*, double *Teph1*, double ∗ *vEDot*)

13.11.2.8 void IFTE_init (const char ∗ *fname*)

13.12 jpl_int.h File Reference

Classes

- struct jpl_eph_data
- struct interpolation_info

Macros

- #define MAX_KERNEL_SIZE 2036
- #define JPL_HEADER_SIZE (5 ∗ sizeof(double) + 41 ∗ sizeof(JPLlong))

Typedefs

- typedef unsigned int JPLlong

13.12.1 Macro Definition Documentation

13.12.1.1 #define JPL_HEADER_SIZE (5 ∗ sizeof(double) + 41 ∗ sizeof(JPLlong))

13.12.1.2 #define MAX_KERNEL_SIZE 2036

13.12.2 Typedef Documentation

13.12.2.1 typedef unsigned int JPLlong

13.13 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_EPHEMERIS_VERSION 200
- #define JPL_EPHEM_KERNEL_SIZE 204
- #define JPL_EPHEM_KERNEL_RECORD_SIZE 208
- #define JPL_EPHEM_KERNEL_NCOEFF 212
- #define JPL_EPHEM_KERNEL_SWAP_BYTES 216

Functions

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

13.13.1 Macro Definition Documentation

13.13.1.1 #define DLL_FUNC

13.13.1.2 #define JPL_EPHEM_AU_IN_KM 28

13.13.1.3 #define JPL_EPHEM_EARTH_MOON_RATIO 36

13.13.1.4 #define JPL_EPHEM_END_JD 8

13.13.1.5 #define JPL_EPHEM_EPHEMERIS_VERSION 200

13.13.1.6 #define JPL_EPHEM_KERNEL_NCOEFF 212

13.13.1.7 #define JPL_EPHEM_KERNEL_RECORD_SIZE 208

13.13.1.8 #define JPL_EPHEM_KERNEL_SIZE 204

13.13.1.9 #define JPL_EPHEM_KERNEL_SWAP_BYTES 216

13.13.1.10 #define JPL_EPHEM_N_CONSTANTS 24

13.13.1.11 #define JPL_EPHEM_START_JD 0

13.13.1.12 #define JPL_EPHEM_STEP 16

13.13.2 Function Documentation

13.13.2.1 void **DLL_FUNC** jpl_close_ephemeris (void ∗ *ephem*)

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

13.13.2.3 double **DLL_FUNC** jpl_get_long (const void ∗ *ephem,* const int *value*)

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

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

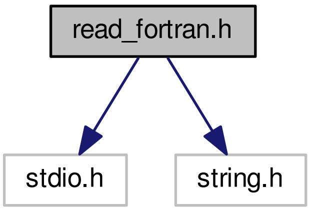
13.13.2.6 int **DLL_FUNC** jpl_state (void ∗ *ephem,* const double *et[2],* const int *list[12],* double *pv[ ][6],* double *nut[4],* const int *bary*)

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

13.14 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

13.14.1 Function Documentation

13.14.1.1 void close_file ( )

13.14.1.2 int open_file (char ∗ *fname*)

13.14.1.3 char read_char ( )

13.14.1.4 void read_character (int *len,* char ∗ *str*)

13.14.1.5 double read_double ( )

13.14.1.6 float read_float ( )

13.14.1.7 int read_int ( )

13.14.1.8 int read_record_int ( )

13.14.2 Variable Documentation

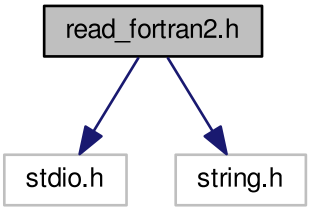
13.14.2.1 FILE∗ c_fileptr

13.14.2.2 int swapByte

13.15 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

13.15.1 Function Documentation

13.15.1.1 void close_file2 ( )

13.15.1.2 void open_file2 (char ∗ *fname,* int ∗ *swap*)

13.15.1.3 void read_character2 (int *len,* char ∗ *str*)

13.15.1.4 double read_double2 ( )

13.15.1.5 float read_float2 ( )

13.15.1.6 int read_int2 ( )

13.15.1.7 int read_record_int2 ( )

13.15.2 Variable Documentation

13.15.2.1 FILE∗ c_fileptr2

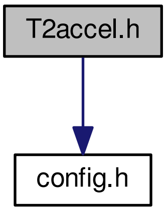
13.15.2.2 int swapByte2

13.16 README.md File Reference

13.17 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)
- 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

13.17.1 Macro Definition Documentation

13.17.1.1 #define ACCEL_LSQ

13.17.1.2 #define ACCEL_MULTMATRIX

13.17.1.3 #define ACCEL_UINV

13.17.2 Function Documentation

13.17.2.1 double accel_lsq_qr (double ∗∗ *dm,* double ∗ *data,* double ∗ *oparm,* int *ndata,* int *nparam,* double ∗∗ *Ocvm*)

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

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

13.17.2.4 int accel_uinv (double ∗ *_m,* int *n*)

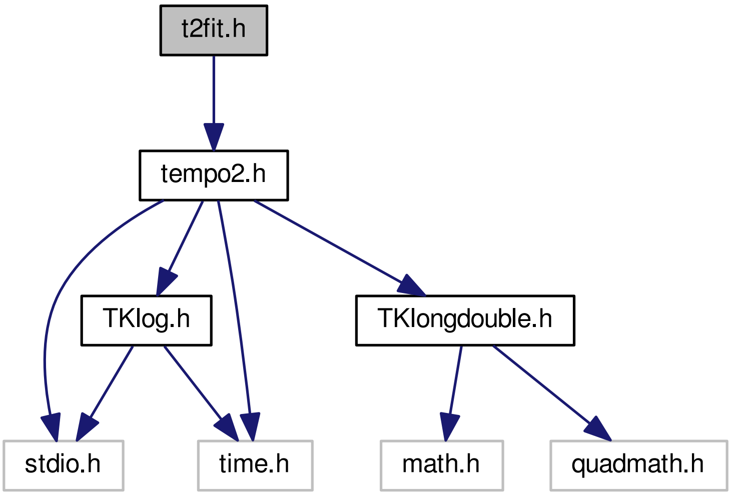
13.17.3 Variable Documentation

13.17.3.1 char useT2accel

13.18 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)
- 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)

13.18.1 Function Documentation

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

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

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

13.18.1.4 void t2Fit_fillFitInfo (**pulsar** ∗ *psr,* **FitInfo** & *OUT*)

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

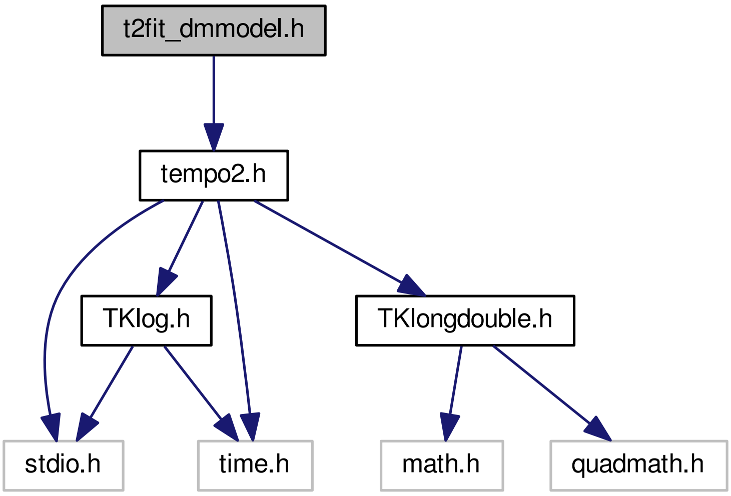
13.18.1.6 unsigned int t2Fit_getFitData (**pulsar** ∗ *psr,* double ∗ *x,* double ∗ *y,* double ∗ *e,* int ∗ *ip*)

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

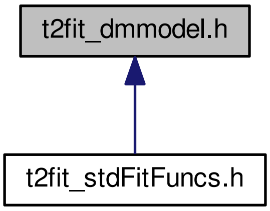
13.19 t2fit_dmmodel.h File Reference

```
#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)

13.19.1 Function Documentation

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

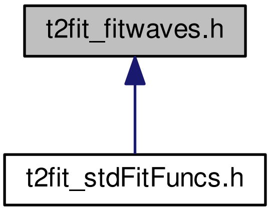
13.19.1.2 double t2FitFunc_dmmodelDM (pulsar ∗ *psr,* int *ipsr,* double *x,* int *ipos,* param_label *label,* int *k*)

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

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

13.20 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)

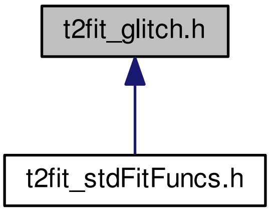
13.20.1 Function Documentation

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

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

13.21 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)

13.21.1 Function Documentation

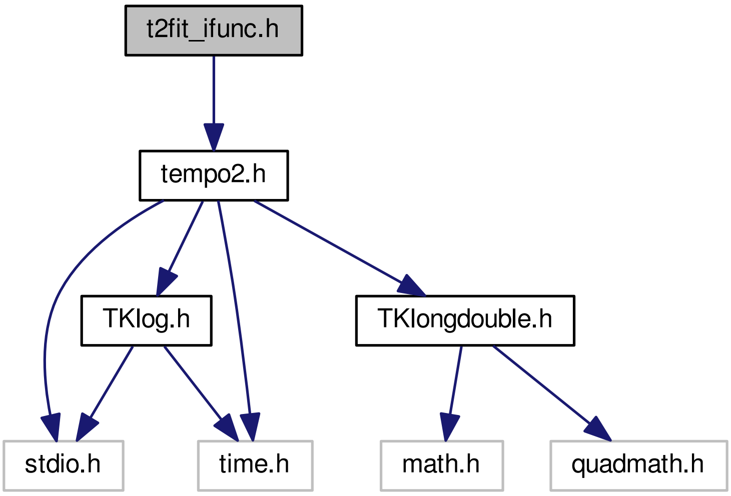
13.21.1.1 double t2FitFunc_stdGlitch (**pulsar** ∗ *psr,* int *ipsr,* double *x,* int *ipos,* **param_label** *label,* int *k*)

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

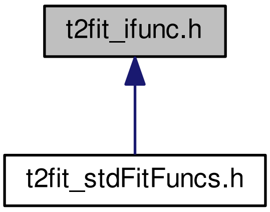
13.22 t2fit_ifunc.h File Reference

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

Include dependency graph for t2fit_ifunc.h:



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



Functions

- double ifunc (const double ∗mjd, const double t, const int N, const int k)
- double sinfunc (const double ∗T, const double t, const int k)
- double t2FitFunc_sifunc (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- double t2FitFunc_ifunc (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_ifunc (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)

13.22.1 Function Documentation

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

13.22.1.2 double sinfunc (const double ∗ *T,* const double *t,* const int *k*)

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

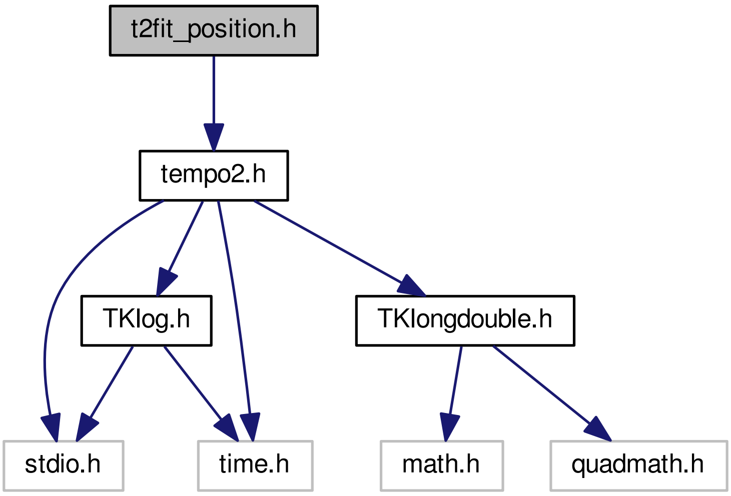
13.22.1.4 double t2FitFunc_sifunc (pulsar ∗ *psr,* int *ipsr,* double *x,* int *ipos,* param_label *label,* int *k*)

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

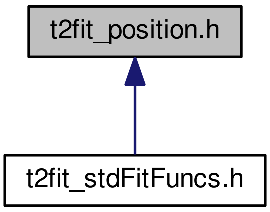
13.23 t2fit_position.h File Reference

`#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)

13.23.1 Function Documentation

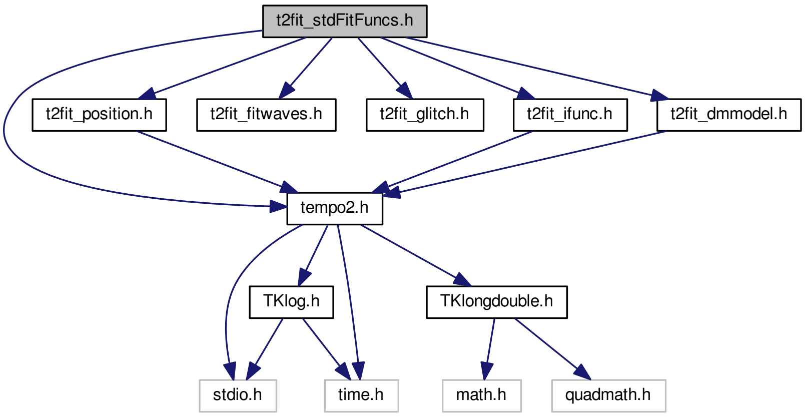
13.23.1.1 double t2FitFunc_stdPosition (pulsar ∗ *psr,* int *ipsr,* double *x,* int *ipos,* param_label *label,* int *k*)

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

13.24 t2fit_stdFitFuncs.h File Reference

```
#include <tempo2.h>
#include "t2fit_position.h"
#include "t2fit_fitwaves.h"
#include "t2fit_glitch.h"
#include "t2fit_ifunc.h"
#include "t2fit_dmmodel.h"
```

Include dependency graph for t2fit_stdFitFuncs.h:



Functions

- void t2UpdateFunc_simpleAdd (pulsar ∗psr, int ipsr, param_label label, int k, double val, double error)
- void t2UpdateFunc_simpleMinus (pulsar ∗psr, int ipsr, param_label label, int k, double val, double error)
- double t2FitFunc_zero (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_zero (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_stdFreq (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_stdFreq (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_binaryModels (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_binaryModels (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_planet (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_planet (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_stdDm (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- double t2FitFunc_stdGravWav (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_stdGravWav (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_telPos (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_telPos (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_ifunc (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_ifunc (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_miscDm (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_miscDm (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_jump (pulsar ∗psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_jump (pulsar ∗psr, int ipsr, param_label label, int k, double val, double err)

13.24.1 Function Documentation

13.24.1.1 double t2FitFunc_binaryModels (pulsar ∗ *psr,* int *ipsr,* double *x,* int *ipos,* param_label *label,* int *k*)

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

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

13.24.1.4 double t2FitFunc_miscDm (pulsar ∗ *psr,* int *ipsr,* double *x,* int *ipos,* param_label *label,* int *k*)

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

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

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

13.24.1.8 double t2FitFunc_stdGravWav (pulsar ∗ *psr,* int *ipsr,* double *x,* int *ipos,* param_label *label,* int *k*)

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

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

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

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

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

13.24.1.14 void t2UpdateFunc_miscDm (pulsar ∗ *psr,* int *ipsr,* param_label *label,* int *k,* double *val,* double *err*)

13.24.1.15 void t2UpdateFunc_planet (pulsar ∗ *psr,* int *ipsr,* param_label *label,* int *k,* double *val,* double *err*)

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

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

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

13.24.1.19 void t2UpdateFunc_stdGravWav (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

13.24.1.20 void t2UpdateFunc_telPos (**pulsar** ∗ *psr,* int *ipsr,* **param_label** *label,* int *k,* double *val,* double *err*)

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

13.25 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)

13.25.1 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

13.25.2 Function Documentation

13.25.2.1 unsigned long genrand_int32 (void)

13.25.2.2 double genrand_real1 (void)

13.25.2.3 void init_genrand (unsigned long *s*)

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

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

13.25.2.6 double TKfindMax_d (double ∗ *x,* int *n*)

13.25.2.7 float TKfindMax_f (float ∗ *x,* int *n*)

13.25.2.8 double TKfindMedian_d (double ∗ *val,* int *count*)

13.25.2.9 float TKfindMedian_f (float ∗ *val,* int *count*)

13.25.2.10 double TKfindMin_d (double ∗ *x,* int *n*)

13.25.2.11 float TKfindMin_f (float ∗ *x,* int *n*)

13.25.2.12 double TKfindRMS_d (double ∗ *x,* int *n*)

13.25.2.13 float TKfindRMS_f (float ∗ *x,* int *n*)

13.25.2.14 float TKfindRMSweight_d (double ∗ *x,* double ∗ *e,* int *n*)

13.25.2.15 double TKgaussDev (long ∗ *seed*)

13.25.2.16 double TKmean_d (double ∗ *x,* int *n*)

13.25.2.17 float TKmean_f (float ∗ *x,* int *n*)

13.25.2.18 double TKranDev (long ∗ *seed*)

13.25.2.19 double TKrange_d (double ∗ *x,* int *n*)

13.25.2.20 float TKrange_f (float ∗ *x,* int *n*)

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

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

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

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

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

13.25.2.26 long TKsetSeed ( )

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

13.25.2.28 void TKsort_2f (float ∗ *val,* float ∗ *val2,* int *nobs*)

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

13.25.2.30 void TKsort_d (double ∗ *val,* int *nobs*)

13.25.2.31 void TKsort_f (float ∗ *val,* int *nobs*)

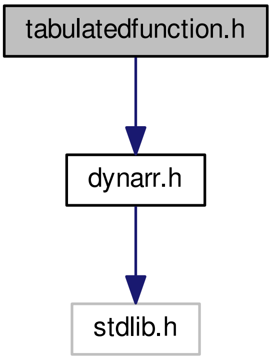
13.25.2.32 double TKvariance_d (double ∗ *x,* int *n*)

13.25.2.33 void TKzeromean_d (int *n,* double ∗ *y*)

13.26 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)

13.26.1 Function Documentation

13.26.1.1 double TabulatedFunction_getEndX (**TabulatedFunction** ∗ *func*)

13.26.1.2 double TabulatedFunction_getStartX (**TabulatedFunction** ∗ *func*)

13.26.1.3 double TabulatedFunction_getValue (**TabulatedFunction** ∗ *func,* double *x*)

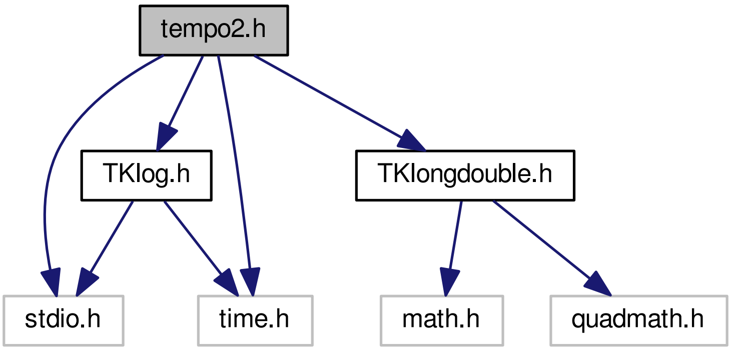
13.26.1.4 void TabulatedFunction_load (**TabulatedFunction** ∗ *func,* char ∗ *fileName*)

13.27 tempo2.h File Reference

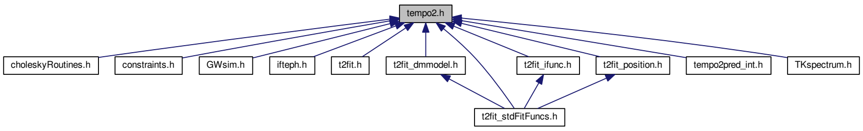
contains the main interface to libtempo2.

```
#include <stdio.h>
#include <time.h>
#include "TKlongdouble.h"
#include "TKlog.h"
```

Include dependency graph for tempo2.h:



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



Classes

- struct FitInfo

contains details of the fit

- struct storePrecision
- struct parameter

  Holds the values for a parameter.
- struct clock_correction
- struct observation

  A struct containing the details of a single obesrvation.
- struct pulsar

  contains the details for a single pulsar.
- struct observatory

Macros

- #define TEMPO2_h_HASH "$Id: da810cd817da8229f1a155b119a771e9e962a9b7 $"
- #define TEMPO2_h_VER "2015.09.0"
- #define TEMPO2_h_MAJOR_VER 2015.09
- #define TEMPO2_h_MINOR_VER 0
- #define TSUN longdouble(4.925490947e-6)
- #define MAX_FREQ_DERIVATIVES 13
- #define MAX_DM_DERIVATIVES 10
- #define MAX_PSR_VAL 40
- #define MAX_COMPANIONS 4
- #define NE_SW_DEFAULT 4
- #define ECLIPTIC_OBLIQUITY_VAL 84381.4059
- #define MAX_COEFF 5000
- #define MAX_CLKCORR 5000
- #define MAX_LEAPSEC 100
- #define MAX_STRLEN 1000
- #define MAX_FILELEN 500
- #define MAX_STOREPRECISION 50
- #define MAX_OBSN_VAL 20000
- #define MAX_SITE 100
- #define MAX_PARAMS 2000
- #define MAX_JUMPS 2000
- #define MAX_WHITE 100
- #define MAX_IFUNC 1000
- #define MAX_TEL_CLK_OFFS 500
- #define MAX_TEL_DX 500
- #define MAX_TEL_DY 500
- #define MAX_TEL_DZ 500
- #define MAX_FIT 10000
- #define MAX_T2EFAC 100
- #define MAX_T2EQUAD 100
- #define MAX_TNEF 50
- #define MAX_TNEQ 50
- #define MAX_TNGN 50
- #define MAX_TNBN 50 /∗maximum number of TNBandNoise parameters allowd∗/
- #define MAX_TNECORR 50
- #define MAX_TNDMEv 10 /∗Maximum number of TNDMEvents allowed ∗/
- #define MAX_TNSQ 50
- #define MAX_BPJ_JUMPS 5
- #define MAX_TOFFSET 10
- #define MAX_QUAD 150
- #define MAX_DMX 512

- #define MAX_FLAGS 20
- #define MAX_FLAG_LEN 32
- #define MAX_CLK_CORR 30
- #define SECDAY 86400.0
- #define SECDAYl longdouble(86400.0)
- #define SPEED_LIGHT 299792458.0
- #define SOLAR_MASS 1.98892e30
- #define SOLAR_RADIUS 6.96e8
- #define BIG_G 6.673e-11
- #define GM 1.3271243999e20
- #define GM_C3 4.925490947e-6
- #define GMJ_C3 4.70255e-9
- #define GMS_C3 1.40797e-9
- #define GMV_C3 1.2061e-11
- #define GMU_C3 2.14539e-10
- #define GMN_C3 2.54488e-10
- #define OBLQ 23.4458333333333333
- #define AULTSC 499.00478364
- #define AU_DIST 1.49598e11
- #define DM_CONST 2.41e-4
- #define DM_CONST_SI 7.436e6
- #define PCM 3.08568025e16
- #define MASYR2RADS 1.53628185e-16
- #define MAX_MSG 50
- #define LEAPSECOND_FILE "/clock/leap.sec"
- #define UT1_FILE "/clock/ut1.dat"
- #define TDBTDT_FILE "/ephemeris/TDB.1950.2050"
- #define IFTEPH_FILE "/ephemeris/TIMEEPH_short.te405"
- #define OBSSYS_FILE "/observatory/newobsys.dat"
- #define SI_UNITS 1
- #define TDB_UNITS 2
- #define IF99_TIMEEPH 1
- #define FB90_TIMEEPH 2
- #define T2C_IAU2000B 1
- #define T2C_TEMPO 2
- #define HAVE_GWSIM_H

Typedefs

- typedef int param_label
- typedef int constraint_label
- typedef double(∗ paramDerivFunc) (struct pulsar ∗, int, double, int, param_label, int)

  a function used to get the derivative of a parameter w.r.t. data.
- typedef double(∗ constraintDerivFunc) (struct pulsar ∗, int, constraint_label, param_label, int, int)

  a function used to get the derivative of a parameter w.r.t. constraint.
- typedef void(∗ paramUpdateFunc) (struct pulsar ∗, int, param_label, int, double, double)

  a function used to update the parameters after a fit.
- typedef struct FitInfo FitInfo

  contains details of the fit
- typedef struct storePrecision storePrecision
- typedef struct parameter parameter

  Holds the values for a parameter.
- typedef struct observation observation

  A struct containing the details of a single obesrvation.
- typedef struct pulsar pulsar

  contains the details for a single pulsar.

Enumerations

- enum label {
  param_raj, param_decj, param_f, param_pepoch,
  param_posepoch, param_dmepoch, param_dm, param_pmra,
  param_pmdec, param_px, param_sini, param_pb,
  param_fb, param_t0, param_a1, param_om,
  param_pmrv, 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_tzrfrq, param_fddc,
  param_fddi, param_fd, param_dr, param_dtheta,
  param_tspan, param_bpjep, param_bpjph, param_bpja1,
  param_bpjec, param_bpjom, param_bpjpb, param_wave_om,
  param_kom, param_kin, param_shapmax, param_dth,
  param_a0, param_b0, param_xomdot, param_afac,
  param_eps1dot, param_eps2dot, param_tres, param_wave_dm,
  param_waveepoch_dm, param_dshk, param_ephver, param_daop,
  param_iperharm, param_dmassplanet, param_waveepoch, param_ifunc,
  param_clk_offs, param_dmx, param_dmxr1, param_dmxr2,
  param_dmmodel, param_gwsingle, param_cgw, param_quad_om,
  param_h3, param_h4, param_nharm, param_stig,
  param_telx, param_tely, param_telz, param_telEpoch,
  param_quad_ifunc_p, param_quad_ifunc_c, param_tel_dx, param_tel_dy,
  param_tel_dz, param_tel_vx, param_tel_vy, param_tel_vz,
  param_tel_x0, param_tel_y0, param_tel_z0, param_gwm_amp,
  param_gwecc, param_gwb_amp, param_dm_sin1yr, param_dm_cos1yr,
  param_brake, param_stateSwitchT, param_df1, param_LAST,
  param_ZERO, param_JUMP }

  enumeration for the various parameters that appear in a .par file

- enum constraint {
  constraint_dmmodel_mean, constraint_dmmodel_dm1, constraint_dmmodel_cw_0, constraint_dmmodel_cw_1,
  constraint_dmmodel_cw_2, constraint_dmmodel_cw_3, constraint_ifunc_0, constraint_ifunc_1,
  constraint_ifunc_2, constraint_tel_dx_0, constraint_tel_dx_1, constraint_tel_dx_2,
  constraint_tel_dy_0, constraint_tel_dy_1, constraint_tel_dy_2, constraint_tel_dz_0,
  constraint_tel_dz_1, constraint_tel_dz_2, constraint_quad_ifunc_p_0, constraint_quad_ifunc_p_1,
  constraint_quad_ifunc_p_2, constraint_quad_ifunc_c_0, constraint_quad_ifunc_c_1, constraint_quad_ifunc_c_2,
  constraint_dmmodel_cw_year_sin, constraint_dmmodel_cw_year_cos, constraint_dmmodel_cw_year_xsin, constraint_dmmodel_cw_year_xcos,
  constraint_dmmodel_cw_year_sin2, constraint_dmmodel_cw_year_cos2, constraint_dmmodel_cw_px, constraint_ifunc_year_sin,
  constraint_ifunc_year_cos, constraint_ifunc_year_xsin, constraint_ifunc_year_xcos, constraint_ifunc_year_sin2,
  constraint_ifunc_year_cos2, constraint_qifunc_p_year_sin, constraint_qifunc_p_year_cos, constraint_qifunc_p_year_xsin,
  constraint_qifunc_p_year_xcos, constraint_qifunc_p_year_sin2, constraint_qifunc_p_year_cos2, constraint_qifunc_c_year_sin,
  constraint_qifunc_c_year_cos, constraint_qifunc_c_year_xsin, constraint_qifunc_c_year_xcos, constraint_qifunc_c_year_sin2,
  constraint_qifunc_c_year_cos2, constraint_LAST }

  These represent the possible constraints to the fit that have been implemented.

Functions

- int id_residual (float xcurs, float ycurs)
- float setStart (float xcurs, float ycurs, int flag)
- int zoom_graphics (float xcurs2, float ycurs2, int flag)
- void getInputs (pulsar ∗psr, int argc, char ∗argv[ ], char timFile[ ][MAX_FILELEN], char parFile[ ][MAX_FIL↩ ELEN], 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_ST↩ RLEN][MAX_FILELEN])
- int readSimpleParfile (FILE ∗fin, pulsar ∗p)
- int setupParameterFileDefaults (pulsar ∗p)
- void displayParameters (int pos, char timeFile[ ][MAX_FILELEN], char parFile[ ][MAX_FILELEN], pulsar ∗psr, int npsr)
- void initialise (pulsar ∗psr, int noWarnings)
- void initialiseOne (pulsar ∗psr, int noWarnings, int fullSetup)
- void destroyOne (pulsar ∗psr)
- void recordPrecision (pulsar ∗psr, longdouble prec, const char ∗routine, const char ∗comment)
- void readTimfile (pulsar ∗psr, char timFile[ ][MAX_FILELEN], int npsr)
- void formBats (pulsar ∗psr, int npsr)
- void formBatsAll (pulsar ∗psr, int npsr)
- void updateBatsAll (pulsar ∗psr, int npsr)
- void formResiduals (pulsar ∗psr, int npsr, int removeMean)
- int bootstrap (pulsar ∗psr, int p, int npsr)
- void doFitAll (pulsar ∗psr, int npsr, const char ∗covarFuncFile) DEPRECATED
- void doFit (pulsar ∗psr, int npsr, int writeModel) DEPRECATED
- void doFitDCM (pulsar ∗psr, const char ∗dcmFile, const char ∗covarFuncFile, int npsr, int writeModel) DE↩ PRECATED
- void doFitGlobal (pulsar ∗psr, int npsr, double ∗globalParameter, int nGlobal, int writeModel) DEPRECATED
- void getCholeskyMatrix (double ∗∗uinv, const char ∗fname, pulsar ∗psr, double ∗resx, double ∗resy, double ∗rese, int np, int nc, int ∗ip)
- double getParamDeriv (pulsar ∗psr, int ipos, double x, int i, int k) DEPRECATED
- void textOutput (pulsar ∗psr, int npsr, double globalParameter, int nGlobal, int outRes, int newpar, const char ∗fname)
- void shapiro_delay (pulsar ∗psr, int npsr, int p, int i, double delt, double dt_SSB)
- void dm_delays (pulsar ∗psr, int npsr, int p, int i, double delt, double dt_SSB)
- void calculate_bclt (pulsar ∗psr, int npsr)
- void secularMotion (pulsar ∗psr, int npsr)
- void autoConstraints (pulsar ∗psr, int ipsr, int npsr)
- void setPlugPath ()
- void sortToAs (pulsar ∗psr)
- void preProcess (pulsar ∗psr, int npsr, int argc, char ∗argv[ ])
- void preProcessSimple (pulsar ∗psr)
- void preProcessSimple1 (pulsar ∗psr, int tempo1, double thelast)
- void preProcessSimple2 (pulsar ∗psr, float startdmmjd, int ndm, float ∗dmvals, int trimonly)
- void preProcessSimple3 (pulsar ∗psr)
- void useSelectFile (char ∗fname, pulsar ∗psr, int npsr)
- void processSimultaneous (char ∗line, pulsar ∗psr, int npsr)
- void processFlag (char ∗line, pulsar ∗psr, int npsr)
- void logicFlag (char ∗line, pulsar ∗psr, int npsr)

- void toa2utc (pulsar ∗psr, int npsr)
- void utc2tai (pulsar ∗psr, int npsr)
- void tt2tb (pulsar ∗psr, int npsr)
- void tai2tt (pulsar ∗psr, int npsr)
- void tai2ut1 (pulsar ∗psr, int npsr)
- void vectorPulsar (pulsar ∗psr, int npsr)
- void readEphemeris (pulsar ∗psr, int npsr, int addEphemNoise)
- void readOneEphemeris (pulsar ∗psr, int npsr, int addEphemNoise, int obsNumber)
- void readEphemeris_calceph (pulsar ∗psr, int npsr)
- void get_obsCoord (pulsar ∗psr, int npsr)
- void get_OneobsCoord (pulsar ∗psr, int npsr, int obs)
- double calcRMS (pulsar ∗psr, int p)
- void allocateMemory (pulsar ∗psr, int realloc)
- void destroyMemory (pulsar ∗psr)
- void readJBO_bat (char ∗fname, pulsar ∗psr, int p)
- void readObsFile (double alat[MAX_SITE], double along[MAX_SITE], double elev[MAX_SITE], int icoord[MAX_SITE], char obsnam[MAX_SITE][100], char obscode[MAX_SITE][100], int ∗nobservatory, int obsnum[MAX_SITE])
- double dotproduct (double ∗v1, double ∗v2)
- void vectorsum (double ∗res, double ∗v1, double ∗v2)
- void vectorscale (double ∗v, double k)
- void writeTim (const char ∗timname, pulsar ∗psr, const char ∗fileFormat)
- int turn_hms (double turn, char ∗hms)
- int turn_dms (double turn, char ∗dms)
- double dms_turn (char ∗line)
- double hms_turn (char ∗line)
- double turn_deg (double turn)
- longdouble fortran_mod (longdouble a, longdouble p)
- int fortran_nint (double x)
- long fortran_nlong (longdouble x)
- void equ2ecl (double ∗x)
- void copyParam (parameter p1, parameter ∗p2)
- void copyPSR (pulsar ∗p, int p1, int p2)
- longdouble getParameterValue (pulsar ∗psr, int param, int arr)
- void simplePlot (pulsar ∗psr, double unitFlag)
- double solarWindModel (pulsar psr, int iobs)
- double MSSmodel (pulsar ∗psr, int p, int obs, int param)
- void updateMSS (pulsar ∗psr, double val, double err, int pos)
- double BTmodel (pulsar ∗psr, int p, int obs, int param)
- void updateBT (pulsar ∗psr, double val, double err, int pos)
- double BTJmodel (pulsar ∗psr, int p, int obs, int param, int arr)
- void updateBTJ (pulsar ∗psr, double val, double err, int pos, int arr)
- double BTXmodel (pulsar ∗psr, int p, int obs, int param, int arr)
- void updateBTX (pulsar ∗psr, double val, double err, int pos, int arr)
- double ELL1model (pulsar ∗psr, int p, int obs, int param)
- void updateELL1 (pulsar ∗psr, double val, double err, int pos)
- longdouble DDmodel (pulsar ∗psr, int p, int obs, int param)
- void updateDD (pulsar ∗psr, double val, double err, int pos)
- double T2model (pulsar ∗psr, int p, int obs, int param, int arr)
- void updateT2 (pulsar ∗psr, double val, double err, int pos, int arr)
- double T2_PTAmodel (pulsar ∗psr, int p, int obs, int param, int arr)
- void updateT2_PTA (pulsar ∗psr, double val, double err, int pos, int arr)
- double JVmodel (pulsar ∗psr, int p, int obs, int param, int arr)
- void updateJV (pulsar ∗psr, double val, double err, int pos, int arr)
- double DDKmodel (pulsar ∗psr, int p, int obs, int param)

- void updateDDK (pulsar ∗psr, double val, double err, int pos)
- double DDSmodel (pulsar ∗psr, int p, int obs, int param)
- void updateDDS (pulsar ∗psr, double val, double err, int pos)
- double DDGRmodel (pulsar ∗psr, int p, int obs, int param)
- void updateDDGR (pulsar ∗psr, double val, double err, int pos)
- double DDHmodel (pulsar ∗psr, int p, int obs, int param)
- void updateDDH (pulsar ∗psr, double val, double err, int pos)
- double ELL1Hmodel (pulsar ∗psr, int p, int obs, int param)
- void updateELL1H (pulsar ∗psr, double val, double err, int pos)
- void displayMsg (int type, const char ∗key, const char ∗searchStr, const char ∗variableStr, int noWarnings)
- void CVSdisplayVersion (const char ∗file, const char ∗func, const char ∗verNum)
- void transform_units (struct pulsar ∗psr, int from, int to)
- void FITfuncs (double x, double afunc[ ], int ma, pulsar ∗psr, int ipos, int ipsr)
- void updateParameters (pulsar ∗psr, int p, double ∗val, double ∗error)
- void defineClockCorrectionSequence (char ∗fileList, int dispWarnings)
- void getClockCorrections (observation ∗obs, const char ∗clockFrom, const char ∗clockTo, int warnings)
- double getCorrectionTT (observation ∗obs)
- double getCorrection (observation ∗obs, const char ∗clockFrom, const char ∗clockTo, int warnings)
- observatory ∗ getObservatory (char ∗code)
- void lookup_observatory_alias (char ∗incode, char ∗outcode)
- void get_obsCoord_IAU2000B (double observatory_trs[3], double zenith_trs[3], longdouble tt_mjd, longdouble utc_mjd, double observatory_crs[3], double zenith_crs[3], double observatory_velocity_crs[3])
- void get_EOP (double mjd, double ∗xp, double ∗yp, double ∗dut1, double ∗dut1dot, int dispWarnings, char ∗eopcFile)
- void compute_tropospheric_delays (pulsar ∗psr, int npsr)

Variables

- char TEMPO2_ENVIRON [ ]
- char TEMPO2_ERROR [ ]
- char NEWFIT
- int MAX_PSR
- int MAX_OBSN
- double ECLIPTIC_OBLIQUITY
- int forceGlobalFit
- int veryFast
- char tempo2MachineType [MAX_FILELEN]
- int displayCVSversion
- char dcmFile [MAX_FILELEN]
- char covarFuncFile [MAX_FILELEN]
- char tempo2_plug_path [32][MAX_STRLEN]
- int tempo2_plug_path_len

13.27.1 Detailed Description

contains the main interface to libtempo2.

Note

some parts of this to be moved to an internal interface

13.27.2 Macro Definition Documentation

13.27.2.1 #define AU_DIST 1.49598e11

1 AU in m

13.27.2.2 #define AULTSC 499.00478364

Number of light seconds in 1 AU

13.27.2.3 #define BIG_G 6.673e-11

Gravitational constant

13.27.2.4 #define DM_CONST 2.41e-4

13.27.2.5 #define DM_CONST_SI 7.436e6

Dispersion constant in SI units

13.27.2.6 #define ECLIPTIC_OBLIQUITY_VAL 84381.4059

mean obliquity of ecliptic in arcsec

13.27.2.7 #define FB90_TIMEEPH 2

Fairhead & Bretagnon time ephemeris

13.27.2.8 #define GM 1.3271243999e20

Gravitational constant ∗ mass sun

13.27.2.9 #define GM_C3 4.925490947e-6

GM_odot/c^3 (in seconds)

13.27.2.10 #define GMJ_C3 4.70255e-9

GM_jupiter/c^3 (in seconds)

13.27.2.11 #define GMN_C3 2.54488e-10

GM_neptune/c^3 (in seconds)

13.27.2.12 #define GMS_C3 1.40797e-9

GM_saturn/c^3 (in seconds)

13.27.2.13 #define GMU_C3 2.14539e-10

GM_uranus/c^3 (in seconds)

13.27.2.14 #define GMV_C3 1.2061e-11

GM_venus/c^3 (in seconds)

13.27.2.15 #define HAVE_GWSIM_H

13.27.2.16 #define IF99_TIMEEPH 1

Irwin & Fukushima time ephemeris

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

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

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

13.27.2.19 #define MASYR2RADS 1.53628185e-16

Converts from mas/yr to rad/s

13.27.2.20 #define MAX_BPJ_JUMPS 5

Maximum number of jumps in binary params - for BPJ model

13.27.2.21 #define MAX_CLK_CORR 30

Maximum number of steps in the correction to TT

13.27.2.22 #define MAX_CLKCORR 5000

Maximum number of lines in time.dat file

13.27.2.23 #define MAX_COEFF 5000

Maximum number of coefficients in polyco

13.27.2.24 #define MAX_COMPANIONS 4

Maximum number of binary companions

13.27.2.25 #define MAX_DM_DERIVATIVES 10

DM0 -> DMn where n=10

13.27.2.26 #define MAX_DMX 512

Max number of DM steps allowed

13.27.2.27 #define MAX_FILELEN 500

Maximum filename length

13.27.2.28 #define MAX_FIT 10000

Maximum number of parameters to fit for

13.27.2.29 #define MAX_FLAG_LEN 32

Maximum number of characters in each flag

13.27.2.30 #define MAX_FLAGS 20

Maximum number of flags in .tim file/observation

13.27.2.31 #define MAX_FREQ_DERIVATIVES 13

F0 -> Fn where n=10

13.27.2.32 #define MAX_IFUNC 1000

Maximum number of parameters for interpolation function

13.27.2.33 #define MAX_JUMPS 2000

Maximum number of phase jumps

13.27.2.34 #define MAX_LEAPSEC 100

Maximum number of line in the leap second file

13.27.2.35 #define MAX_MSG 50

Maximum number of different warnings

13.27.2.36 #define MAX_OBSN_VAL 20000

Maximum number of TOAs

13.27.2.37 #define MAX_PARAMS 2000

Maximum number of parameters

13.27.2.38 #define MAX_PSR_VAL 40

Maximum number of pulsars

13.27.2.39 #define MAX_QUAD 150

Maximum number of frequency channels in quadrupolar function

13.27.2.40 #define MAX_SITE 100

Maximum number of observatory sites

13.27.2.41 #define MAX_STOREPRECISION 50

How many routines in TEMPO2 store precision information

13.27.2.42 #define MAX_STRLEN 1000

Maximum length for strings

13.27.2.43 #define MAX_T2EFAC 100

Maximum number of T2EFACs allowed

13.27.2.44 #define MAX_T2EQUAD 100

Maximum number of T2EQUADs allowed

13.27.2.45 #define MAX_TEL_CLK_OFFS 500

Maximum number of parameters for telescope clock offset

13.27.2.46 #define MAX_TEL_DX 500

Maximum number of parameters for interpolation function

13.27.2.47 #define MAX_TEL_DY 500

Maximum number of parameters for interpolation function

13.27.2.48 #define MAX_TEL_DZ 500

Maximum number of parameters for interpolation function

13.27.2.49 #define MAX_TNBN 50 /*maximum number of TNBandNoise parameters allowd*/

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

13.27.2.51 #define MAX_TNECORR 50

Maximum number of TNECORRss allowed

13.27.2.52 #define MAX_TNEF 50

Maximum number of TNEFACs allowed

13.27.2.53 #define MAX_TNEQ 50

Maximum number of TNEQUADs allowed

13.27.2.54 #define MAX_TNGN 50

maximum number of TNGroupNoise parameters allowed

13.27.2.55 #define MAX_TNSQ 50

Maximum number of TNEQUADs allowed

13.27.2.56 #define MAX_TOFFSET 10

Number of time jumps allowed in .par file

13.27.2.57 #define MAX_WHITE 100

Maximum number of parameters for whitening

13.27.2.58 #define NE_SW_DEFAULT 4

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

13.27.2.59 #define OBLQ 23.4458333333333333

Obliquity of the ecliptic

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

Path for file containing Observatory data (obsys.dat)

13.27.2.61 #define PCM 3.08568025e16

one parsec in meters

13.27.2.62 #define SECDAY 86400.0

Number of seconds in 1 day

13.27.2.63 #define SECDAYl longdouble(86400.0)

Number of seconds in 1 day

13.27.2.64 #define SI_UNITS 1

New tempo2 mode

13.27.2.65 #define SOLAR_MASS 1.98892e30

Mass of Sun (kg)

13.27.2.66 #define SOLAR_RADIUS 6.96e8

Radius of the Sun (in meters)

13.27.2.67 #define SPEED_LIGHT 299792458.0

Speed of light (m/s)

13.27.2.68 #define T2C_IAU2000B 1

13.27.2.69 #define T2C_TEMPO 2

13.27.2.70 #define TDB_UNITS 2

original tempo mode

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

Path for file containing TDB-TDT ephemeris

13.27.2.72 #define TEMPO2_h_HASH "$Id: da810cd817da8229f1a155b119a771e9e962a9b7 $"

13.27.2.73 #define TEMPO2_h_MAJOR_VER 2015.09

13.27.2.74 #define TEMPO2_h_MINOR_VER 0

13.27.2.75 #define TEMPO2_h_VER "2015.09.0"

13.27.2.76 #define TSUN longdouble(4.925490947e-6)

Solar constant for mass calculations.

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

Path for the file containing TAI-UT1

13.27.3 Typedef Documentation

13.27.3.1 typedef int **constraint_label**

for 'strong typing' - type for enum constraint

13.27.3.2 typedef double(∗ constraintDerivFunc) (struct **pulsar** ∗, int, **constraint_label**, **param_label**, int, int)

a function used to get the derivative of a parameter w.r.t. constraint.

Used to build the derivative matrix for the least squares solvers.

13.27.3.3 typedef struct **FitInfo** **FitInfo**

contains details of the fit

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

13.27.3.4 typedef struct **observation** **observation**

A struct containing the details of a single obesrvation.

13.27.3.5 typedef int **param_label**

for 'strong typing' - type for enum label

13.27.3.6 typedef double(∗ paramDerivFunc) (struct **pulsar** ∗, int, double, int, **param_label**, int)

a function used to get the derivative of a parameter w.r.t. data.

Used to build the derivative matrix for the least squares solvers.

13.27.3.7 typedef struct **parameter** **parameter**

Holds the values for a parameter.

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

Note

> If this structure is modified - must update copyParam in tempo2Util.C

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

a function used to update the parameters after a fit.

13.27.3.9 typedef struct **pulsar** **pulsar**

contains the details for a single pulsar.

Includes an array of observations and parameters

13.27.3.10 typedef struct storePrecision storePrecision

13.27.4 Enumeration Type Documentation

13.27.4.1 enum constraint

These represent the possible constraints to the fit that have been implemented.

Enumerator

constraint_dmmodel_mean

constraint_dmmodel_dm1

constraint_dmmodel_cw_0

constraint_dmmodel_cw_1

constraint_dmmodel_cw_2

constraint_dmmodel_cw_3

constraint_ifunc_0

constraint_ifunc_1

constraint_ifunc_2

constraint_tel_dx_0

constraint_tel_dx_1

constraint_tel_dx_2

constraint_tel_dy_0

constraint_tel_dy_1

constraint_tel_dy_2

constraint_tel_dz_0

constraint_tel_dz_1

constraint_tel_dz_2

constraint_quad_ifunc_p_0

constraint_quad_ifunc_p_1

constraint_quad_ifunc_p_2

constraint_quad_ifunc_c_0

constraint_quad_ifunc_c_1

constraint_quad_ifunc_c_2

constraint_dmmodel_cw_year_sin

constraint_dmmodel_cw_year_cos

constraint_dmmodel_cw_year_xsin

constraint_dmmodel_cw_year_xcos

constraint_dmmodel_cw_year_sin2

constraint_dmmodel_cw_year_cos2

constraint_dmmodel_cw_px

constraint_ifunc_year_sin

constraint_ifunc_year_cos

constraint_ifunc_year_xsin

constraint_ifunc_year_xcos

constraint_ifunc_year_sin2

constraint_ifunc_year_cos2

constraint_qifunc_p_year_sin

constraint_qifunc_p_year_cos

constraint_qifunc_p_year_xsin

constraint_qifunc_p_year_xcos

constraint_qifunc_p_year_sin2

constraint_qifunc_p_year_cos2

constraint_qifunc_c_year_sin

constraint_qifunc_c_year_cos

constraint_qifunc_c_year_xsin

constraint_qifunc_c_year_xcos

constraint_qifunc_c_year_sin2

constraint_qifunc_c_year_cos2

constraint_LAST marker for the last constraint

13.27.4.2 enum label

enumeration for the various parameters that appear in a .par file

The last parameter is param_LAST, but there are enumerations after this for spectial fits. It is important not to change the order of the elements

Note

when adding a new parameter, initialise it in intialise.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_pmrv

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_tzrfrq

param_fddc

param_fddi

param_fd

param_dr

param_dtheta

param_tspan

param_bpjep

param_bpjph

param_bpja1

param_bpjec

param_bpjom

param_bpjpb

param_wave_om

param_kom

param_kin

param_shapmax

param_dth

param_a0

param_b0

param_xomdot

param_afac

param_eps1dot

param_eps2dot

param_tres

param_wave_dm

param_waveepoch_dm

param_dshk

param_ephver

param_daop

param_iperharm

param_dmassplanet

param_waveepoch

param_ifunc

param_clk_offs

param_dmx

param_dmxr1

param_dmxr2

param_dmmodel

param_gwsingle

param_cgw

param_quad_om

param_h3

param_h4

param_nharm

param_stig

param_telx

param_tely

param_telz

param_telEpoch

param_quad_ifunc_p

param_quad_ifunc_c

param_tel_dx

param_tel_dy

param_tel_dz

param_tel_vx

param_tel_vy

param_tel_vz

param_tel_x0

param_tel_y0

param_tel_z0

param_gwm_amp

param_gwecc

param_gwb_amp

param_dm_sin1yr

param_dm_cos1yr

param_brake

param_stateSwitchT

param_df1

param_LAST Marker for the last param to be used in for loops

param_ZERO virtual parameter for DC offset

param_JUMP virtual parameter for jumps

13.27.5 Function Documentation

13.27.5.1 void allocateMemory (pulsar ∗ *psr,* int *realloc*)

13.27.5.2 void autoConstraints (pulsar ∗ *psr,* int *ipsr,* int *npsr*)

13.27.5.3 int bootstrap (pulsar ∗ *psr,* int *p,* int *npsr*)

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

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

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

13.27.5.7 double calcRMS (pulsar ∗ *psr,* int *p*)

13.27.5.8 void calculate_bclt (pulsar ∗ *psr,* int *npsr*)

13.27.5.9 void compute_tropospheric_delays (pulsar ∗ *psr,* int *npsr*)

13.27.5.10 void copyParam (parameter *p1,* parameter ∗ *p2*)

13.27.5.11 void copyPSR (pulsar ∗ *p,* int *p1,* int *p2*)

13.27.5.12 void CVSdisplayVersion (const char ∗ *file,* const char ∗ *func,* const char ∗ *verNum*)

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

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

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

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

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

13.27.5.18 void defineClockCorrectionSequence (char ∗ *fileList,* int *dispWarnings*)

13.27.5.19 void destroyMemory (pulsar ∗ *psr*)

13.27.5.20 void destroyOne (pulsar ∗ *psr*)

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

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

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

13.27.5.24 double dms_turn (char ∗ *line*)

13.27.5.25 void doFit (pulsar ∗ *psr,* int *npsr,* int *writeModel*)

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

13.27.5.27 void doFitDCM (pulsar ∗ *psr,* const char ∗ *dcmFile,* const char ∗ *covarFuncFile,* int *npsr,* int *writeModel*)

13.27.5.28 void doFitGlobal (pulsar ∗ *psr,* int *npsr,* double ∗ *globalParameter,* int *nGlobal,* int *writeModel*)

13.27.5.29 double dotproduct (double ∗ *v1,* double ∗ *v2*)

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

13.27.5.31 double ELL1model (pulsar ∗ *psr,* int *p,* int *obs,* int *param*)

13.27.5.32 void equ2ecl (double ∗ *x*)

13.27.5.33 void FITfuncs (double *x,* double *afunc[ ],* int *ma,* pulsar ∗ *psr,* int *ipos,* int *ipsr*)

13.27.5.34 void formBats (pulsar ∗ *psr,* int *npsr*)

13.27.5.35 void formBatsAll (pulsar ∗ *psr,* int *npsr*)

13.27.5.36 void formResiduals (pulsar ∗ *psr,* int *npsr,* int *removeMean*)

13.27.5.37 longdouble fortran_mod (longdouble *a,* longdouble *p*)

13.27.5.38 int fortran_nint (double *x*)

13.27.5.39 long fortran_nlong (longdouble *x*)

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

13.27.5.41 void get_obsCoord (pulsar ∗ *psr,* int *npsr*)

13.27.5.42 void get_obsCoord_IAU2000B (double *observatory_trs[3],* double *zenith_trs[3],* longdouble *tt_mjd,* longdouble *utc_mjd,* double *observatory_crs[3],* double *zenith_crs[3],* double *observatory_velocity_crs[3]*)

13.27.5.43 void get_OneobsCoord (pulsar ∗ *psr,* int *npsr,* int *obs*)

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

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

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

13.27.5.47 double getCorrectionTT (observation ∗ *obs*)

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

13.27.5.49 observatory∗ getObservatory (char ∗ *code*)

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

13.27.5.51 longdouble getParameterValue (pulsar ∗ *psr,* int *param,* int *arr*)

13.27.5.52 double hms_turn (char ∗ *line*)

13.27.5.53 int id_residual (float *xcurs,* float *ycurs*)

13.27.5.54 void initialise (pulsar ∗ *psr,* int *noWarnings*)

13.27.5.55 void initialiseOne (pulsar ∗ *psr,* int *noWarnings,* int *fullSetup*)

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

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

13.27.5.58 void lookup_observatory_alias (char ∗ *incode,* char ∗ *outcode*)

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

13.27.5.60 void polyco (pulsar ∗ *psr,* int *npsr,* longdouble *polyco_MJD1,* longdouble *polyco_MJD2,* int *nspan,* int *ncoeff,* longdouble *maxha,* char ∗ *sitename,* longdouble *freq,* longdouble *coeff[MAX_COEFF],* int *trueDM,* char ∗ *polyco_file*)

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

13.27.5.62 void preProcessSimple (pulsar ∗ *psr*)

13.27.5.63 void preProcessSimple1 (pulsar ∗ *psr,* int *tempo1,* double *thelast*)

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

13.27.5.65 void preProcessSimple3 (pulsar ∗ *psr*)

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

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

13.27.5.68 void readEphemeris (pulsar ∗ *psr,* int *npsr,* int *addEphemNoise*)

13.27.5.69 void readEphemeris_calceph (pulsar ∗ *psr,* int *npsr*)

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

13.27.5.71 void readObsFile (double *alat[MAX_SITE],* double *along[MAX_SITE],* double *elev[MAX_SITE],* int *icoord[MAX_SITE],* char *obsnam[MAX_SITE][100],* char *obscode[MAX_SITE][100],* int ∗ *nobservatory,* int *obsnum[MAX_SITE]*)

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

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

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

13.27.5.75 int readSimpleParfile (FILE ∗ *fin,* pulsar ∗ *p*)

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

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

13.27.5.78 void secularMotion (pulsar ∗ *psr,* int *npsr*)

13.27.5.79 void setPlugPath ( )

13.27.5.80 float setStart (float *xcurs,* float *ycurs,* int *flag*)

13.27.5.81 int setupParameterFileDefaults (pulsar ∗ *p*)

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

13.27.5.83 void simplePlot (pulsar ∗ *psr,* double *unitFlag*)

13.27.5.84 double solarWindModel (pulsar *psr,* int *iobs*)

13.27.5.85 void sortToAs (pulsar ∗ *psr*)

13.27.5.86 double T2_PTAmodel (pulsar ∗ *psr,* int *p,* int *obs,* int *param,* int *arr*)

13.27.5.87 double T2model (pulsar ∗ *psr,* int *p,* int *obs,* int *param,* int *arr*)

13.27.5.88 void tai2tt (pulsar ∗ *psr,* int *npsr*)

13.27.5.89 void tai2ut1 (pulsar ∗ *psr,* int *npsr*)

13.27.5.90 void textOutput (pulsar ∗ *psr,* int *npsr,* double *globalParameter,* int *nGlobal,* int *outRes,* int *newpar,* const char ∗ *fname*)

13.27.5.91 void toa2utc (pulsar ∗ *psr,* int *npsr*)

13.27.5.92 void transform_units (struct pulsar ∗ *psr,* int *from,* int *to*)

13.27.5.93 void tt2tb (pulsar ∗ *psr,* int *npsr*)

13.27.5.94 double turn_deg (double *turn*)

13.27.5.95 int turn_dms (double *turn,* char ∗ *dms*)

13.27.5.96 int turn_hms (double *turn,* char ∗ *hms*)

13.27.5.97 void updateBatsAll (pulsar ∗ *psr,* int *npsr*)

13.27.5.98 void updateBT (pulsar ∗ *psr,* double *val,* double *err,* int *pos*)

13.27.5.99 void updateBTJ (pulsar ∗ *psr,* double *val,* double *err,* int *pos,* int *arr*)

13.27.5.100 void updateBTX (pulsar ∗ *psr*, double *val*, double *err*, int *pos*, int *arr*)

13.27.5.101 void updateDD (pulsar ∗ *psr*, double *val*, double *err*, int *pos*)

13.27.5.102 void updateDDGR (pulsar ∗ *psr*, double *val*, double *err*, int *pos*)

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

13.27.5.104 void updateDDK (pulsar ∗ *psr*, double *val*, double *err*, int *pos*)

13.27.5.105 void updateDDS (pulsar ∗ *psr*, double *val*, double *err*, int *pos*)

13.27.5.106 void updateELL1 (pulsar ∗ *psr*, double *val*, double *err*, int *pos*)

13.27.5.107 void updateELL1H (pulsar ∗ *psr*, double *val*, double *err*, int *pos*)

13.27.5.108 void updateJV (pulsar ∗ *psr*, double *val*, double *err*, int *pos*, int *arr*)

13.27.5.109 void updateMSS (pulsar ∗ *psr*, double *val*, double *err*, int *pos*)

13.27.5.110 void updateParameters (pulsar ∗ *psr*, int *p*, double ∗ *val*, double ∗ *error*)

13.27.5.111 void updateT2 (pulsar ∗ *psr*, double *val*, double *err*, int *pos*, int *arr*)

13.27.5.112 void updateT2_PTA (pulsar ∗ *psr*, double *val*, double *err*, int *pos*, int *arr*)

13.27.5.113 void useSelectFile (char ∗ *fname*, pulsar ∗ *psr*, int *npsr*)

13.27.5.114 void utc2tai (pulsar ∗ *psr*, int *npsr*)

13.27.5.115 void vectorPulsar (pulsar ∗ *psr*, int *npsr*)

13.27.5.116 void vectorscale (double ∗ *v*, double *k*)

13.27.5.117 void vectorsum (double ∗ *res*, double ∗ *v1*, double ∗ *v2*)

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

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

13.27.6 Variable Documentation

13.27.6.1 char covarFuncFile[MAX_FILELEN]

13.27.6.2 char dcmFile[MAX_FILELEN]

13.27.6.3 int displayCVSversion

Display CVS version

13.27.6.4 double ECLIPTIC_OBLIQUITY

13.27.6.5 int forceGlobalFit

Global = 1 if we are forcing a global fit

13.27.6.6 int MAX_OBSN

size of the arrays of observations inside each pulsar

13.27.6.7 int MAX_PSR

size of the array of pulsars used in tempo2

13.27.6.8 char NEWFIT

global boolean used to enable new fit.

Warning

this will be removed in future.

13.27.6.9 char TEMPO2_ENVIRON[ ]

TEMPO2 environment variable

13.27.6.10 char TEMPO2_ERROR[ ]

TEMPO2 error messages

13.27.6.11 char tempo2_plug_path[32][MAX_STRLEN]

paths to search for plugins

13.27.6.12 int tempo2_plug_path_len

13.27.6.13 char tempo2MachineType[MAX_FILELEN]

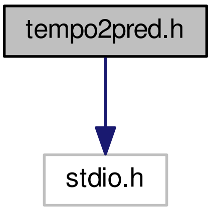
13.27.6.14 int veryFast

Global to run the code fast

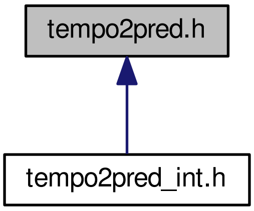
13.28 tempo2pred.h File Reference

```
#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

13.28.1 Enumeration Type Documentation

13.28.1.1 enum T2PredictorKind

Enumerator

NonePredType

Cheby

T1

13.28.2 Function Documentation

13.28.2.1 void T2Predictor_Copy (T2Predictor ∗ *into_t2p,* const T2Predictor ∗ *from_t2p*)

13.28.2.2 void T2Predictor_Destroy (T2Predictor ∗ *t2p*)

13.28.2.3 int T2Predictor_FRead (T2Predictor ∗ *t2p,* FILE ∗ *f*)

13.28.2.4 void T2Predictor_FWrite (const T2Predictor ∗ *t2p,* FILE ∗ *f*)

13.28.2.5 long double T2Predictor_GetEndFreq (T2Predictor ∗ *t2p*)

13.28.2.6 long double T2Predictor_GetEndMJD (T2Predictor ∗ *t2p*)

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

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

13.28.2.9 int T2Predictor_GetPlan (char ∗ *filename,* long double *mjd_start,* long double *mjd_end,* long double *step,* long double *freq,* long double ∗ *phase0,* int ∗ *nsegments,* long double ∗ *pulse_frequencies*)

13.28.2.10 int T2Predictor_GetPlan_Ext (char ∗ *filename,* long double *mjd_start,* long double *mjd_end,* long double *step,* long double *freq,* char ∗ *psrname,* char ∗ *sitename,* long double ∗ *phase0,* int ∗ *nsegments,* long double ∗ *pulse_frequencies*)

13.28.2.11 char∗ T2Predictor_GetPSRName (T2Predictor ∗ *t2p*)

13.28.2.12 char∗ T2Predictor_GetSiteName (T2Predictor ∗ *t2p*)

13.28.2.13 long double T2Predictor_GetStartFreq (T2Predictor ∗ *t2p*)

13.28.2.14 long double T2Predictor_GetStartMJD (T2Predictor ∗ *t2p*)

13.28.2.15 void T2Predictor_Init (T2Predictor ∗ *t2p*)

13.28.2.16 int T2Predictor_Insert (T2Predictor ∗ *into_t2p,* const T2Predictor ∗ *from_t2p*)

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

13.28.2.18 T2PredictorKind T2Predictor_Kind (T2Predictor ∗ *t2p*)

13.28.2.19 int T2Predictor_Read (T2Predictor ∗ *t2p,* char ∗ *fname*)

13.28.2.20 void T2Predictor_Write (const T2Predictor ∗ *t2p,* char ∗ *fname*)

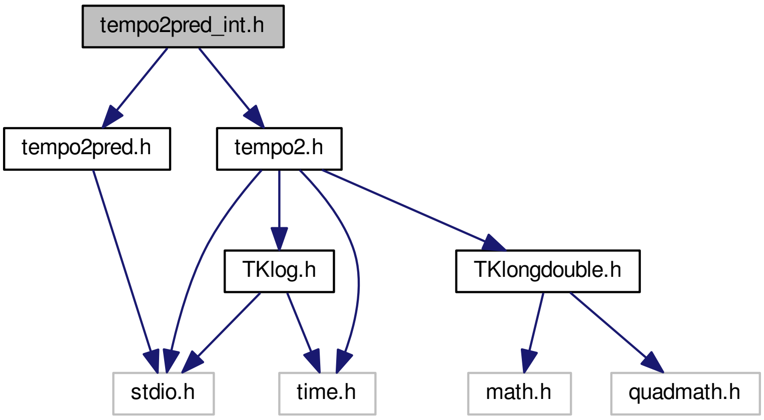
13.28.3 Variable Documentation

13.28.3.1 int ChebyModelSet_OutOfRange

13.29 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)

13.29.1 Function Documentation

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

13.29.1.2 void Cheby2D_Construct_x_Derivative (Cheby2D ∗ *dcheby,* const Cheby2D ∗ *cheby*)

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

13.29.1.4 void ChebyModel_Construct (ChebyModel ∗ *cm,* const pulsar ∗ *psr*)

13.29.1.5 void ChebyModel_Copy (**ChebyModel** ∗ *cm,* **ChebyModel** ∗ *from*)

13.29.1.6 void ChebyModel_Destroy (**ChebyModel** ∗ *cm*)

13.29.1.7 long double ChebyModel_GetFrequency (const **ChebyModel** ∗ *cm,* long double *mjd,* long double *freq*)

13.29.1.8 long double ChebyModel_GetPhase (const **ChebyModel** ∗ *cm,* long double *mjd,* long double *freq*)

13.29.1.9 void ChebyModel_Init (**ChebyModel** ∗ *cmodel,* int *nmjdcoeff,* int *nfreqcoeff*)

13.29.1.10 int ChebyModel_Read (**ChebyModel** ∗ *cm,* FILE ∗ *f*)

13.29.1.11 void ChebyModel_Test (**ChebyModel** ∗ *cm,* const **pulsar** ∗ *psr,* int *nmjd,* int *nfreq,* long double ∗ *residualRMS,* long double ∗ *residualMAV*)

13.29.1.12 void ChebyModel_Write (const **ChebyModel** ∗ *cm,* FILE ∗ *f*)

13.29.1.13 void ChebyModelSet_Construct (**ChebyModelSet** ∗ *cms,* const **pulsar** ∗ *psr,* const char ∗ *sitename,* long double *mjd_start,* long double *mjd_end,* long double *segment_length,* long double *overlap,* long double *freq_start,* long double *freq_end,* int *nmjdcoeff,* int *nfreqcoeff*)

13.29.1.14 void ChebyModelSet_Destroy (**ChebyModelSet** ∗ *cms*)

13.29.1.15 long double ChebyModelSet_GetFrequency (const **ChebyModelSet** ∗ *cms,* long double *mjd,* long double *freq*)

13.29.1.16 **ChebyModel**∗ ChebyModelSet_GetNearest (const **ChebyModelSet** ∗ *cms,* long double *mjd*)

13.29.1.17 long double ChebyModelSet_GetPhase (const **ChebyModelSet** ∗ *cms,* long double *mjd,* long double *freq*)

13.29.1.18 void ChebyModelSet_Init (**ChebyModelSet** ∗ *cms*)

13.29.1.19 int ChebyModelSet_Insert (**ChebyModelSet** ∗ *cms,* const **ChebyModelSet** ∗ *from*)

13.29.1.20 void ChebyModelSet_Keep (**ChebyModelSet** ∗ *cms,* unsigned *nmjd,* const long double ∗ *mjd*)

13.29.1.21 int ChebyModelSet_Read (**ChebyModelSet** ∗ *cms,* FILE ∗ *f*)

13.29.1.22 void ChebyModelSet_Test (**ChebyModelSet** ∗ *cms,* const **pulsar** ∗ *psr,* int *nmjd,* int *nfreq,* long double ∗ *residualRMS,* long double ∗ *residualMAV*)

13.29.1.23 void ChebyModelSet_Write (const **ChebyModelSet** ∗ *cms,* FILE ∗ *f*)

13.29.1.24 long double T1Polyco_GetFrequency (const **T1Polyco** ∗ *t1p,* long double *mjd,* long double *freq*)

13.29.1.25 long double T1Polyco_GetPhase (const **T1Polyco** ∗ *t1p,* long double *mjd,* long double *freq*)

13.29.1.26 int T1Polyco_Read (**T1Polyco** ∗ *t1p,* FILE ∗ *f*)

13.29.1.27 void T1Polyco_Write (const **T1Polyco** ∗ *t1p,* FILE ∗ *f*)

13.29.1.28 void T1PolycoSet_Destroy (**T1PolycoSet** ∗ *t1ps*)

13.29.1.29 long double T1PolycoSet_GetFrequency (const **T1PolycoSet** ∗ *t1ps,* long double *mjd,* long double *freq*)

13.29.1.30 **T1Polyco**∗ T1PolycoSet_GetNearest (long double *mjd*)

13.29.1.31 long double T1PolycoSet_GetPhase (const T1PolycoSet ∗ *t1ps,* long double *mjd,* long double *freq*)

13.29.1.32 int T1PolycoSet_Read (T1PolycoSet ∗ *t1ps,* FILE ∗ *f*)

13.29.1.33 void T1PolycoSet_Write (const T1PolycoSet ∗ *t1ps,* FILE ∗ *f*)

13.30 tempo2Util.h File Reference

Functions

- double turn_deg (double turn)
- double dms_turn (char ∗line)
- double hms_turn (char ∗line)

13.30.1 Function Documentation

13.30.1.1 double dms_turn (char ∗ *line*)

13.30.1.2 double hms_turn (char ∗ *line*)

13.30.1.3 double turn_deg (double *turn*)

13.31 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)

13.31.1 Function Documentation

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

13.31.1.2 void cholesky_dmModel (double ∗∗ *m,* double *D,* double *d,* double *ref_freq,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

13.31.1.3 void cholesky_dmModelCovarParam (double ∗∗ *m,* double *alpha,* double *a,* double *b,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

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

13.31.1.5 int cholesky_formUinv (double ∗∗ *uinv,* double ∗∗ *m,* int *np*)

13.31.1.6 void cholesky_powerlawModel (double ∗∗ *m,* double *modelAlpha,* double *modelFc,* double *modelA,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

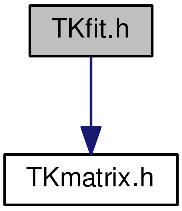
13.31.1.7 void cholesky_powerlawModel_withBeta (double ∗∗ *m,* double *modelAlpha,* double *beta,* double *modelFc,* double *modelA,* double ∗ *resx,* double ∗ *resy,* double ∗ *rese,* int *np,* int *nc*)

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

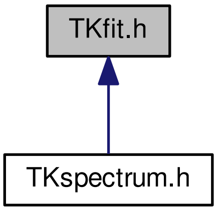
13.32 TKfit.h File Reference

```
#include "TKmatrix.h"
```

Include dependency graph for TKfit.h:



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



Functions

- double TKleastSquares (double ∗b, double ∗white_b, double ∗∗designMatrix, double ∗∗white_designMatrix, int n, int nf, double tol, char rescale_errors, double ∗outP, double ∗e, double ∗∗CVM)
- double TKrobustLeastSquares (double ∗b, double ∗white_b, double ∗∗designMatrix, double ∗∗white_←designMatrix, int n, int nf, double tol, char rescale_errors, double ∗outP, double ∗e, double ∗∗cvm, char robust)

- double TKconstrainedLeastSquares (double ∗b, double ∗white_b, double ∗∗designMatrix, double ∗∗white_←designMatrix, double ∗∗constraintsMatrix, int n, int nf, int nconstraints, double tol, char rescale_errors, double ∗outP, double ∗e, double ∗∗cvm)
- double TKrobustConstrainedLeastSquares (double ∗b, double ∗white_b, double ∗∗designMatrix, double ∗∗white_designMatrix, double ∗∗constraintsMatrix, int n, int nf, int nconstraints, double tol, char rescale←_errors, double ∗outP, double ∗e, double ∗∗cvm, char robust)
- void TKleastSquares_svd (double ∗x, double ∗y, double ∗sig, int n, double ∗p, double ∗e, int nf, double ∗∗cvm, double ∗chisq, void(∗fitFuncs)(double, double[ ], int), int weight)
- void TKleastSquares_svd_noErr (double ∗x, double ∗y, int n, double ∗p, int nf, void(∗fitFuncs)(double, double[ ], int))
- void TKremovePoly_f (float ∗px, float ∗py, int n, int m)
- void TKremovePoly_d (double ∗px, double ∗py, int n, int m)
- void TKfindPoly_d (double ∗px, double ∗py, int n, int m, double ∗p)
- void TKfitPoly (double x, double ∗v, int m)

13.32.1 Function Documentation

13.32.1.1 double TKconstrainedLeastSquares (double ∗ *b,* double ∗ *white_b,* double ∗∗ *designMatrix,* double ∗∗ *white_designMatrix,* double ∗∗ *constraintsMatrix,* int *n,* int *nf,* int *nconstraints,* double *tol,* char *rescale_errors,* double ∗ *outP,* double ∗ *e,* double ∗∗ *cvm*)

13.32.1.2 void TKfindPoly_d (double ∗ *px,* double ∗ *py,* int *n,* int *m,* double ∗ *p*)

13.32.1.3 void TKfitPoly (double *x,* double ∗ *v,* int *m*)

13.32.1.4 double TKleastSquares (double ∗ *b,* double ∗ *white_b,* double ∗∗ *designMatrix,* double ∗∗ *white_designMatrix,* int *n,* int *nf,* double *tol,* char *rescale_errors,* double ∗ *outP,* double ∗ *e,* double ∗∗ *CVM*)

13.32.1.5 void TKleastSquares_svd (double ∗ *x,* double ∗ *y,* double ∗ *sig,* int *n,* double ∗ *p,* double ∗ *e,* int *nf,* double ∗∗ *cvm,* double ∗ *chisq,* void(∗)(double, double[ ], int) *fitFuncs,* int *weight*)

13.32.1.6 void TKleastSquares_svd_noErr (double ∗ *x,* double ∗ *y,* int *n,* double ∗ *p,* int *nf,* void(∗)(double, double[ ], int) *fitFuncs*)

13.32.1.7 void TKremovePoly_d (double ∗ *px,* double ∗ *py,* int *n,* int *m*)

13.32.1.8 void TKremovePoly_f (float ∗ *px,* float ∗ *py,* int *n,* int *m*)

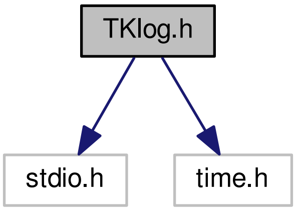
13.32.1.9 double TKrobustConstrainedLeastSquares (double ∗ *b,* double ∗ *white_b,* double ∗∗ *designMatrix,* double ∗∗ *white_designMatrix,* double ∗∗ *constraintsMatrix,* int *n,* int *nf,* int *nconstraints,* double *tol,* char *rescale_errors,* double ∗ *outP,* double ∗ *e,* double ∗∗ *cvm,* char *robust*)

13.32.1.10 double TKrobustLeastSquares (double ∗ *b,* double ∗ *white_b,* double ∗∗ *designMatrix,* double ∗∗ *white_designMatrix,* int *n,* int *nf,* double *tol,* char *rescale_errors,* double ∗ *outP,* double ∗ *e,* double ∗∗ *cvm,* char *robust*)

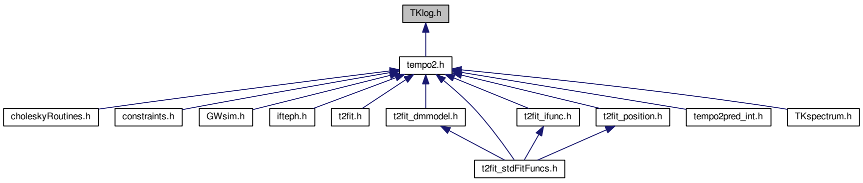
13.33 TKlog.h File Reference

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

Include dependency graph for TKlog.h:



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



Macros

- #define TK_MAX_ERRORS 16
- #define TK_MAX_ERROR_LEN 128
- #define LOG_OUTFILE stdout
- #define RESETCOLOR "\033[0m"
- #define WARNCOLOR RESETCOLOR "\033[0;35m"
- #define BOLDCOLOR RESETCOLOR "\033[1m"
- #define ERRORCOLOR RESETCOLOR "\033[1;31m"
- #define WHERESTR "[%s:%d] "
- #define WHEREARG __FILE__, __LINE__
- #define ENDL "\n"
- #define WHEREERR ERRORCOLOR "∗∗∗ERROR∗∗∗\n [%s:%d] " RESETCOLOR
- #define WHEREWARN BOLDCOLOR "[%s:%d] " WARNCOLOR "Warning: " RESETCOLOR
- #define ENDERR "\n∗∗∗!!!!!∗∗∗"
- #define WHERETCHK "[%s:%d] T=%.2f s: "
- #define _LOG(_fmt, ...) _TKchklog(LOG_OUTFILE,_fmt,##__VA_ARGS__)
- #define logmsg(_fmt, ...) _LOG(WHERESTR _fmt ENDL, WHEREARG,##__VA_ARGS__)
- #define logdbg(_fmt, ...) if(debugFlag)logmsg(_fmt,##__VA_ARGS__)
- #define logerr(_fmt, ...) do{TK_STORE_ERROR(_fmt,##__VA_ARGS__); _LOG(WHEREERR _fmt ENDERR 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 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]

13.33.1 Macro Definition Documentation

13.33.1.1 #define _LOG(*_fmt, ...*) _TKchklog(LOG_OUTFILE,_fmt,##__VA_ARGS__)

13.33.1.2 #define BOLDCOLOR RESETCOLOR "\033[1m"

13.33.1.3 #define DEPRECATED

13.33.1.4 #define ENDERR "\n∗∗∗!!!!!∗∗∗"

13.33.1.5 #define ENDL "\n"

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

13.33.1.7 #define LOG_OUTFILE stdout

13.33.1.8 #define logdbg(*_fmt, ...*) if(debugFlag)logmsg(_fmt,##__VA_ARGS__)

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

13.33.1.10 #define logmsg(*_fmt, ...*) _LOG(WHERESTR _fmt ENDL, WHEREARG,##__VA_ARGS__)

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

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

13.33.1.13 #define RESETCOLOR "\033[0m"

13.33.1.14 #define TK_MAX_ERROR_LEN 128

13.33.1.15 #define TK_MAX_ERRORS 16

13.33.1.16 #define TK_STORE_ERROR(*_fmt, ...*) if(TK_errorCount < TK_MAX_ERROR↩ S)snprintf(TK_errorlog[TK_errorCount],TK_MAX_ERROR_LEN, _fmt,##__VA_ARGS__); ++TK_errorCount

13.33.1.17 #define TK_STORE_WARNING(*_fmt, ...*) if(TK_warnCount < TK_MAX_ERROR↩ S)snprintf(TK_warnlog[TK_warnCount],TK_MAX_ERROR_LEN, _fmt,##__VA_ARGS__); ++TK_warnCount

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

13.33.1.19 #define WHEREARG __FILE__, __LINE__

13.33.1.20 #define WHEREERR ERRORCOLOR "∗∗∗ERROR∗∗∗\n [%s:%d] " RESETCOLOR

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

13.33.1.22 #define WHERETCHK "[%s:%d] T=%.2f s: "

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

13.33.2 Function Documentation

13.33.2.1 void _TKchklog (FILE ∗ , const char ∗ , ...)

13.33.2.2 int logerr_check ( )

13.33.3 Variable Documentation

13.33.3.1 int debugFlag

13.33.3.2 int tcheck

13.33.3.3 clock_t timer_clk

13.33.3.4 unsigned TK_errorCount

13.33.3.5 char TK_errorlog[TK_MAX_ERRORS][TK_MAX_ERROR_LEN]

13.33.3.6 unsigned TK_warnCount

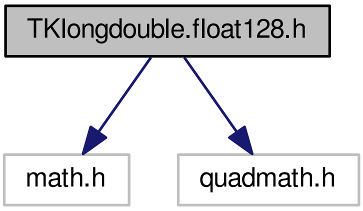
13.33.3.7 char TK_warnlog[TK_MAX_ERRORS][TK_MAX_ERROR_LEN]

13.33.3.8 int writeResiduals

13.34 TKlongdouble.float128.h File Reference

```
#include <math.h>
#include <quadmath.h>
```

Include dependency graph for TKlongdouble.float128.h:



Macros

- #define USE_BUILTIN_LONGDOUBLE
- #define LONGDOUBLE_IS_FLOAT128
- #define LONGDOUBLE_ONE 1.0Q
- #define longdouble(a) a##Q
- #define FMT_LD "Q"
- #define LD_PI M_PIq
- #define cosl cosq
- #define sinl sinq
- #define floorl floorq
- #define fabsl fabsq

Typedefs

- typedef __float128 longdouble

Functions

- longdouble parse_longdouble (const char ∗str)
- int ld_printf (const char ∗__format,...)
- int ld_fprintf (FILE ∗__stream, const char ∗__format,...)
- int ld_sprintf (char ∗__str, const char ∗__format,...)

13.34.1 Macro Definition Documentation

13.34.1.1 #define cosl cosq

13.34.1.2 #define fabsl fabsq

13.34.1.3 #define floorl floorq

13.34.1.4 #define FMT_LD "Q"

13.34.1.5 #define LD_PI M_PIq

13.34.1.6 #define longdouble(*a*) a##Q

13.34.1.7 #define LONGDOUBLE_IS_FLOAT128

13.34.1.8 #define LONGDOUBLE_ONE 1.0Q

13.34.1.9 #define sinl sinq

13.34.1.10 #define USE_BUILTIN_LONGDOUBLE

13.34.2 Typedef Documentation

13.34.2.1 typedef __float128 longdouble

13.34.3 Function Documentation

13.34.3.1 int ld_fprintf (FILE ∗ *__stream,* const char ∗ *__format,* *...*)

13.34.3.2 int ld_printf (const char ∗ *__format,* *...*)

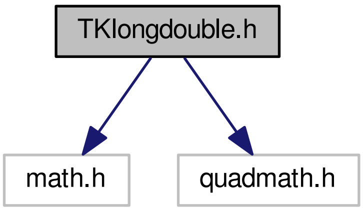
13.34.3.3 int ld_sprintf (char ∗ *__str,* const char ∗ *__format,* *...*)

13.34.3.4 longdouble parse_longdouble (const char ∗ *str*)

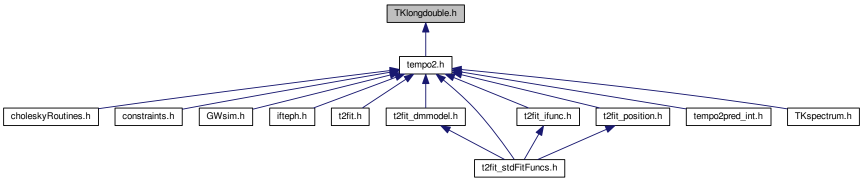
13.35 TKlongdouble.h File Reference

```
#include <math.h>
#include <quadmath.h>
```

Include dependency graph for TKlongdouble.h:



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



Macros

- #define USE_BUILTIN_LONGDOUBLE
- #define LONGDOUBLE_IS_FLOAT128
- #define LONGDOUBLE_ONE 1.0Q
- #define longdouble(a) a##Q
- #define FMT_LD "Q"
- #define LD_PI M_PIq
- #define cosl cosq
- #define sinl sinq
- #define floorl floorq
- #define fabsl fabsq

Typedefs

- typedef __float128 longdouble

Functions

- longdouble parse_longdouble (const char ∗str)
- int ld_printf (const char ∗__format,...)
- int ld_fprintf (FILE ∗__stream, const char ∗__format,...)
- int ld_sprintf (char ∗__str, const char ∗__format,...)

13.35.1 Macro Definition Documentation

13.35.1.1 #define cosl cosq

13.35.1.2 #define fabsl fabsq

13.35.1.3 #define floorl floorq

13.35.1.4 #define FMT_LD "Q"

13.35.1.5 #define LD_PI M_PIq

13.35.1.6 #define longdouble(*a*) a##Q

13.35.1.7 #define LONGDOUBLE_IS_FLOAT128

13.35.1.8 #define LONGDOUBLE_ONE 1.0Q

13.35.1.9 #define sinl sinq

13.35.1.10 #define USE_BUILTIN_LONGDOUBLE

13.35.2 Typedef Documentation

13.35.2.1 typedef __float128 longdouble

13.35.3 Function Documentation

13.35.3.1 int ld_fprintf (FILE ∗ *__stream,* const char ∗ *__format, ...*)

13.35.3.2 int ld_printf (const char ∗ __*format*, ...)

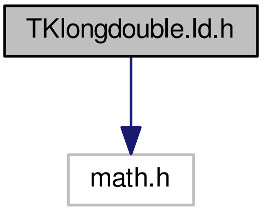
13.35.3.3 int ld_sprintf (char ∗ __*str*, const char ∗ __*format*, ...)

13.35.3.4 longdouble parse_longdouble (const char ∗ *str*)

13.36 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)

13.36.1 Macro Definition Documentation

13.36.1.1 #define ld_fprintf fprintf

13.36.1.2 #define LD_PI M_PI

13.36.1.3 #define ld_printf printf

13.36.1.4 #define ld_sprintf sprintf

13.36.1.5 #define **longdouble**(*a*) a##L

13.36.1.6 #define LONGDOUBLE_IS_IEEE754

13.36.1.7 #define LONGDOUBLE_ONE 1.0L

13.36.1.8 #define USE_BUILTIN_LONGDOUBLE

13.36.2 Typedef Documentation

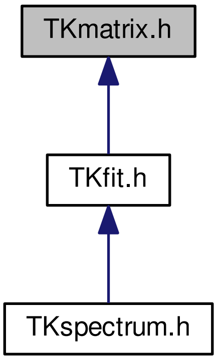
13.36.2.1 typedef long double **longdouble**

13.36.3 Function Documentation

13.36.3.1 **longdouble** parse_longdouble (const char ∗ *str*)

13.37 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)

13.37.1 Function Documentation

13.37.1.1 void free_2df (float ∗∗ *uinv*)

13.37.1.2 void free_blas (double ∗∗ *matrix*)

13.37.1.3 void free_uinv (double ∗∗ *uinv*)

13.37.1.4 int get_blas_cols (double ∗∗ *uinv*)

13.37.1.5 int get_blas_rows (double ∗∗ *uinv*)

13.37.1.6 float∗∗ malloc_2df (int *rows,* int *cols*)

13.37.1.7 double∗∗ malloc_blas (int *n,* int *m*)

13.37.1.8 double∗∗ malloc_uinv (int *n*)

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

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

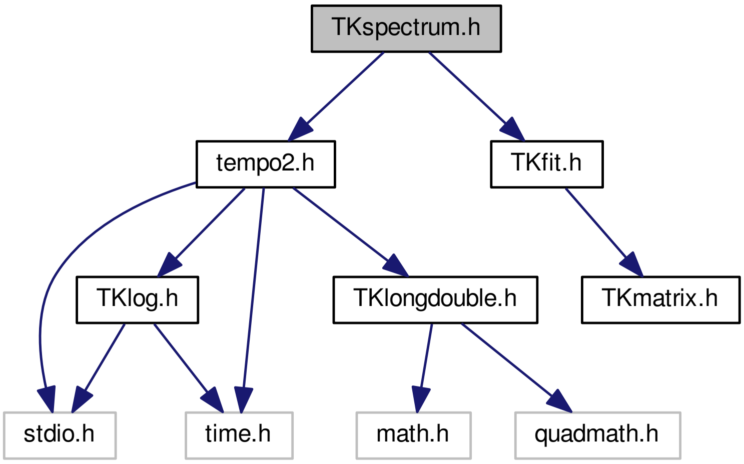
13.37.1.11 void TKmultMatrixVec (double ∗∗ *idcm,* double ∗ *b,* int *ndata,* int *ndata2,* double ∗ *bout*)

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

13.38 TKspectrum.h File Reference

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

Include dependency graph for TKspectrum.h:



Classes

- struct complexVal

Macros

- #define ABS(x) ((x) < 0 ? -(x) : (x))
- #define MAX(x, y) ((x) > (y) ? (x) : (y))
- #define MIN(x, y) ((x) < (y) ? (x) : (y))

Typedefs

- typedef struct complexVal complexVal

Functions

- void readin (pulsar psr)
- void getprtj (int n)
- void indexx8 (int n, double ∗arrin, int ∗indx)
- void getweights (int n, double ∗wt)
- void fit4 (int ∗nfit, double ∗p4, double ∗cov4, int ndostats, double ∗chidf, double ∗avewt)
- void mat20 (double sam[21][21], double a[21][21], int n, double ∗determ, int ∗nbad)
- void sineFunc (double x, double ∗v, int ma)
- void TKsortit (double ∗x, double ∗y, int n)
- void TKaveragePts (double ∗x, double ∗y, int n, int width, double ∗meanX, double ∗meanY, int ∗nMean)
- void TKcmonot (int n, double x[ ], double y[ ], double yd[ ][4])
- void TKspline_interpolate (int n, double ∗x, double ∗y, double yd[ ][4], double ∗interpX, double ∗interpY, int nInterp)
- void TKinterpolateSplineSmoothFixedXPts (double ∗inX, double ∗inY, int inN, double ∗interpX, double ∗interpY, int nInterp)
- void TKhann (double ∗x, double ∗y, int n, double ∗ox, double ∗oy, int ∗on, int width)
- void TKfirstDifference (double ∗x, double ∗y, int n)
- void TK_fitSine (double ∗x, double ∗y, double ∗e, int n, int wErr, double ∗outX, double ∗outY, int ∗outN)
- void TKlomb_d (double ∗x, double ∗y, int n, double ofac, double hifac, double ∗ox, double ∗oy, int ∗outN, double ∗var)
- int TK_fft (short int dir, long n, double ∗x, double ∗y)
- void TK_dft (double ∗x, double ∗y, int n, double ∗outX, double ∗outY, int ∗outN, double ∗outY_re, double ∗outY_im)
- void TK_weightLS (double ∗x, double ∗y, double ∗sig, int n, double ∗outX, double ∗outY, int ∗outN, double ∗outY_re, double ∗outY_im)
- void TK_fitSinusoids (double ∗x, double ∗y, double ∗sig, int n, double ∗outX, double ∗outY, int ∗outN)
- void fitMeanSineFunc (double x, double ∗v, int nfit, pulsar ∗psr, int ival, int ipsr)
- void fitCosSineFunc (double x, double ∗v, int nfit, pulsar ∗psr, int ival, int ipsr)
- int calcSpectraErr (double ∗∗uinv, double ∗resx, double ∗resy, int nres, double ∗specX, double ∗specY, double ∗specE, int nfit)
- double TKspectrum (double ∗x, double ∗y, double ∗e, int n, int averageTime, int smoothWidth, int smooth↩Type, int fitSpline, int preWhite, int specType, double ofac, double hifac, int specOut, double ∗outX, double ∗outY, int ∗nout, int calcWhite, int output, double ∗outY_re, double ∗outY_im)
- void TKboxcar (double ∗x, double ∗y, int n, double ∗ox, double ∗oy, int ∗on, int width)
- void TKcalcSigmaz (pulsar psr, int weights, double ∗ret_tau, double ∗ret_szbias, double ∗ret_e1, double ∗ret_e2, int ∗ret_nval, double mintau)
- int calcSpectra (double ∗∗uinv, double ∗resx, double ∗resy, int nres, double ∗specX, double ∗specY, int nfit)
- int calcSpectra_ri (double ∗∗uinv, double ∗resx, double ∗resy, int nres, double ∗specX, double ∗specY_R, double ∗specY_I, int nfit, pulsar ∗psr)

- int calcSpectra_ri_T (double ∗∗uinv, double ∗resx, double ∗resy, int nres, double ∗specX, double ∗specY_R, double ∗specY_I, int nfit, double T, char useCM, pulsar ∗psr)
- void fitMeanSineFunc_IFUNC (double x, double ∗v, int nfit, pulsar ∗psr, int ival, int ipsr)
- void fitCosSineFunc (double x, double ∗v, int nfit, pulsar ∗psr, int ival)

Variables

- double GLOBAL_OMEGA
- bool verbose_calc_spectra

13.38.1 Macro Definition Documentation

13.38.1.1 #define ABS(*x*) ((x) < 0 ? -(x) : (x))

13.38.1.2 #define MAX(*x,* *y*) ((x) > (y) ? (x) : (y))

13.38.1.3 #define MIN(*x,* *y*) ((x) < (y) ? (x) : (y))

13.38.2 Typedef Documentation

13.38.2.1 typedef struct **complexVal** **complexVal**

13.38.3 Function Documentation

13.38.3.1 int calcSpectra (double ∗∗ *uinv,* double ∗ *resx,* double ∗ *resy,* int *nres,* double ∗ *specX,* double ∗ *specY,* int *nfit*)

13.38.3.2 int calcSpectra_ri (double ∗∗ *uinv,* double ∗ *resx,* double ∗ *resy,* int *nres,* double ∗ *specX,* double ∗ *specY_R,* double ∗ *specY_I,* int *nfit,* **pulsar** ∗ *psr*)

13.38.3.3 int calcSpectra_ri_T (double ∗∗ *uinv,* double ∗ *resx,* double ∗ *resy,* int *nres,* double ∗ *specX,* double ∗ *specY_R,* double ∗ *specY_I,* int *nfit,* double *T,* char *useCM,* **pulsar** ∗ *psr*)

13.38.3.4 int calcSpectraErr (double ∗∗ *uinv,* double ∗ *resx,* double ∗ *resy,* int *nres,* double ∗ *specX,* double ∗ *specY,* double ∗ *specE,* int *nfit*)

13.38.3.5 void fit4 (int ∗ *nfit,* double ∗ *p4,* double ∗ *cov4,* int *ndostats,* double ∗ *chidf,* double ∗ *avewt*)

13.38.3.6 void fitCosSineFunc (double *x,* double ∗ *v,* int *nfit,* **pulsar** ∗ *psr,* int *ival,* int *ipsr*)

13.38.3.7 void fitCosSineFunc (double *x,* double ∗ *v,* int *nfit,* **pulsar** ∗ *psr,* int *ival*)

13.38.3.8 void fitMeanSineFunc (double *x,* double ∗ *v,* int *nfit,* **pulsar** ∗ *psr,* int *ival,* int *ipsr*)

13.38.3.9 void fitMeanSineFunc_IFUNC (double *x,* double ∗ *v,* int *nfit,* **pulsar** ∗ *psr,* int *ival,* int *ipsr*)

13.38.3.10 void getprtj (int *n*)

13.38.3.11 void getweights (int *n,* double ∗ *wt*)

13.38.3.12 void indexx8 (int *n,* double ∗ *arrin,* int ∗ *indx*)

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

13.38.3.14 void readin (**pulsar** *psr*)

13.38.3.15 void sineFunc (double *x,* double ∗ *v,* int *ma*)

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

13.38.3.17 int TK_fft (short int *dir,* long *n,* double ∗ *x,* double ∗ *y*)

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

13.38.3.19 void TK_fitSinusoids (double ∗ *x,* double ∗ *y,* double ∗ *sig,* int *n,* double ∗ *outX,* double ∗ *outY,* int ∗ *outN*)

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

13.38.3.21 void TKaveragePts (double ∗ *x,* double ∗ *y,* int *n,* int *width,* double ∗ *meanX,* double ∗ *meanY,* int ∗ *nMean*)

13.38.3.22 void TKboxcar (double ∗ *x,* double ∗ *y,* int *n,* double ∗ *ox,* double ∗ *oy,* int ∗ *on,* int *width*)

13.38.3.23 void TKcalcSigmaz (pulsar *psr,* int *weights,* double ∗ *ret_tau,* double ∗ *ret_szbias,* double ∗ *ret_e1,* double ∗ *ret_e2,* int ∗ *ret_nval,* double *mintau*)

13.38.3.24 void TKcmonot (int *n,* double *x[ ],* double *y[ ],* double *yd[ ][4]*)

13.38.3.25 void TKfirstDifference (double ∗ *x,* double ∗ *y,* int *n*)

13.38.3.26 void TKhann (double ∗ *x,* double ∗ *y,* int *n,* double ∗ *ox,* double ∗ *oy,* int ∗ *on,* int *width*)

13.38.3.27 void TKinterpolateSplineSmoothFixedXPts (double ∗ *inX,* double ∗ *inY,* int *inN,* double ∗ *interpX,* double ∗ *interpY,* int *nInterp*)

13.38.3.28 void TKlomb_d (double ∗ *x,* double ∗ *y,* int *n,* double *ofac,* double *hifac,* double ∗ *ox,* double ∗ *oy,* int ∗ *outN,* double ∗ *var*)

13.38.3.29 void TKsortit (double ∗ *x,* double ∗ *y,* int *n*)

13.38.3.30 double TKspectrum (double ∗ *x,* double ∗ *y,* double ∗ *e,* int *n,* int *averageTime,* int *smoothWidth,* int *smoothType,* int *fitSpline,* int *preWhite,* int *specType,* double *ofac,* double *hifac,* int *specOut,* double ∗ *outX,* double ∗ *outY,* int ∗ *nout,* int *calcWhite,* int *output,* double ∗ *outY_re,* double ∗ *outY_im*)

13.38.3.31 void TKspline_interpolate (int *n,* double ∗ *x,* double ∗ *y,* double *yd[ ][4],* double ∗ *interpX,* double ∗ *interpY,* int *nInterp*)

13.38.4 Variable Documentation

13.38.4.1 double GLOBAL_OMEGA

13.38.4.2 bool verbose_calc_spectra

13.39 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)

13.39.1 Function Documentation

13.39.1.1 void TKbacksubstitution_svd (**longdouble** ∗∗ ***V,*** **longdouble** ∗ ***w,*** **longdouble** ∗∗ ***U,*** **longdouble** ∗ ***b,*** **longdouble** ∗ ***x,*** int ***n,*** int ***nf***)

13.39.1.2 void TKbidiagonal (**longdouble** ∗∗ ***a,*** **longdouble** ∗ ***anorm,*** int ***ndata,*** int ***nfit,*** **longdouble** ∗∗ ***v,*** **longdouble** ∗ ***w,*** **longdouble** ∗∗ ***u,*** **longdouble** ∗ ***rv1***)

13.39.1.3 **longdouble** TKpythag (**longdouble** ***a,*** **longdouble** ***b***)

13.39.1.4 void TKsingularValueDecomposition_lsq (**longdouble** ∗∗ ***designMatrix,*** int ***n,*** int ***nf,*** **longdouble** ∗∗ ***v,*** **longdouble** ∗ ***w,*** **longdouble** ∗∗ ***u***)

Index

_DARWIN_USE_64_BIT_INODE
    config.h, 77
_LOG
    TKlog.h, 135
_TKchklog
    TKlog.h, 136

ABS
    TKspectrum.h, 144
ACCEL_LSQ
    T2accel.h, 91
ACCEL_MULTMATRIX
    T2accel.h, 91
ACCEL_UINV
    T2accel.h, 91
aSize
    parameter, 46
AU_DIST
    tempo2.h, 109
AULTSC
    tempo2.h, 109
accel_lsq_qr
    T2accel.h, 91
accel_multMatrix
    T2accel.h, 91
accel_multMatrixVec
    T2accel.h, 91
accel_uinv
    T2accel.h, 91
across_g
    gwSrc, 35
    gwgeneralSrc, 33
across_im_g
    gwSrc, 35
    gwgeneralSrc, 33
addTNGlobalEQ
    pulsar, 53
addedNoise
    observation, 39
allocateMemory
    tempo2.h, 120
aplus_g
    gwSrc, 35
    gwgeneralSrc, 33
aplus_im_g
    gwSrc, 35
    gwgeneralSrc, 33
asl_g
    gwgeneralSrc, 33
asl_im_g
    gwgeneralSrc, 33
ast_g
    gwgeneralSrc, 33
ast_im_g
    gwgeneralSrc, 33
au
    jpl_eph_data, 36
auto_constraints
    pulsar, 53
autoConstraints
    tempo2.h, 120
autosetDMCM
    constraints.h, 79
AverageEpochWidth
    pulsar, 53
AverageFlag
    pulsar, 53
AverageResiduals
    pulsar, 53
averagebat
    observation, 39
averageerr
    observation, 39
averageres
    observation, 39
avx_g
    gwgeneralSrc, 33
avx_im_g
    gwgeneralSrc, 33
avy_g
    gwgeneralSrc, 33
avy_im_g
    gwgeneralSrc, 33

BIG_G
    tempo2.h, 109
BOLDCOLOR
    TKlog.h, 135
BTJmodel
    tempo2.h, 120
BTXmodel
    tempo2.h, 120
BTmodel
    tempo2.h, 120
bat
    observation, 39
batCorr
    observation, 39
bbat
    observation, 39

binary_frequency
  T1Polyco, 67
binary_phase
  T1Polyco, 67
binaryModel
  pulsar, 53
bootStrap
  pulsar, 53
bootstrap
  tempo2.h, 120

c_fileptr
  read_fortran.h, 89
c_fileptr2
  read_fortran2.h, 90
CONSTRAINTfuncs
  constraints.h, 80
CVSdisplayVersion
  tempo2.h, 120
cache
  jpl_eph_data, 36
calcRMS
  tempo2.h, 120
calcShapiro
  pulsar, 53
calcSpectra
  TKspectrum.h, 144
calcSpectra_ri
  TKspectrum.h, 144
calcSpectra_ri_T
  TKspectrum.h, 144
calcSpectraErr
  TKspectrum.h, 144
calculate_bclt
  tempo2.h, 120
calculateResidualGW
  GWsim.h, 83
calculateResidualgeneralGW
  GWsim.h, 83
cgw_angpol
  pulsar, 53
cgw_cosinc
  pulsar, 53
cgw_h0
  pulsar, 53
cgw_mc
  pulsar, 53
Cheby
  tempo2pred.h, 127
cheby
  ChebyModel, 28
  T2Predictor, 69
Cheby2D, 27
  coeff, 27
  nx, 27
  ny, 27
Cheby2D_Construct
  tempo2pred_int.h, 129
Cheby2D_Construct_x_Derivative
  tempo2pred_int.h, 129
Cheby2D_Test
  tempo2pred_int.h, 129
ChebyModel, 27
  cheby, 28
  dispersion_constant, 28
  freq_end, 28
  freq_start, 28
  frequency_cheby, 28
  mjd_end, 28
  mjd_start, 28
  psrname, 28
  sitename, 28
ChebyModel_Construct
  tempo2pred_int.h, 129
ChebyModel_Copy
  tempo2pred_int.h, 129
ChebyModel_Destroy
  tempo2pred_int.h, 130
ChebyModel_GetFrequency
  tempo2pred_int.h, 130
ChebyModel_GetPhase
  tempo2pred_int.h, 130
ChebyModel_Init
  tempo2pred_int.h, 130
ChebyModel_Read
  tempo2pred_int.h, 130
ChebyModel_Test
  tempo2pred_int.h, 130
ChebyModel_Write
  tempo2pred_int.h, 130
ChebyModelSet, 29
  nsegments, 29
  segments, 29
ChebyModelSet_Construct
  tempo2pred_int.h, 130
ChebyModelSet_Destroy
  tempo2pred_int.h, 130
ChebyModelSet_GetFrequency
  tempo2pred_int.h, 130
ChebyModelSet_GetNearest
  tempo2pred_int.h, 130
ChebyModelSet_GetPhase
  tempo2pred_int.h, 130
ChebyModelSet_Init
  tempo2pred_int.h, 130
ChebyModelSet_Insert
  tempo2pred_int.h, 130
ChebyModelSet_Keep
  tempo2pred_int.h, 130
ChebyModelSet_OutOfRange
  tempo2pred.h, 128
ChebyModelSet_Read
  tempo2pred_int.h, 130
ChebyModelSet_Test
  tempo2pred_int.h, 130
ChebyModelSet_Write
  tempo2pred_int.h, 130

cholesky.h, 73
cholesky_covarFunc2matrix, 73
cholesky_dmModel, 73
cholesky_dmModelCovarParam, 73
cholesky_ecm, 73
cholesky_formUinv, 73
cholesky_powerlawModel, 73
cholesky_powerlawModel_withBeta, 73
cholesky_readFromCovarianceFunction, 73
cholesky_covarFunc2matrix
cholesky.h, 73
TKcholesky.h, 131
cholesky_dmModel
cholesky.h, 73
TKcholesky.h, 131
cholesky_dmModelCovarParam
cholesky.h, 73
TKcholesky.h, 131
cholesky_ecm
cholesky.h, 73
TKcholesky.h, 131
cholesky_formUinv
cholesky.h, 73
TKcholesky.h, 131
cholesky_powerlawModel
cholesky.h, 73
TKcholesky.h, 132
cholesky_powerlawModel_withBeta
cholesky.h, 73
TKcholesky.h, 132
cholesky_readFromCovarianceFunction
cholesky.h, 73
TKcholesky.h, 132
choleskyRoutines.h, 74
EXPSMOOTH, 76
FCALPHA, 76
FCFINAL, 76
NFIT, 76
T2calculateCholesky, 75
T2calculateCovarFunc, 75
T2calculateDailyCovariance, 75
T2calculateSpectra, 75
T2cholDecomposition, 75
T2cubicFit, 75
T2findSmoothCurve, 75
T2fitSpectra, 75
T2get_covFunc_automatic, 75
T2getHighFreqRes, 75
T2getWhiteNoiseLevel, 75
T2getWhiteRes, 75
T2guess_vals, 75
T2interpolate, 75
T2obtainTimingResiduals, 75
T2writeCovarFuncModel, 76
UPW, 76
WNLEVEL, 76
clk_offsE
pulsar, 53
clk_offsT
pulsar, 53
clk_offsV
pulsar, 54
clkOffsN
pulsar, 54
clock
pulsar, 54
clock_correction, 29
correction, 30
corrects_to, 30
clock_name
observatory, 45
clockCorr
observation, 39
clockFromOverride
pulsar, 54
close_file
read_fortran.h, 89
close_file2
read_fortran2.h, 90
code
observatory, 45
coeff
Cheby2D, 27
T1Polyco, 67
comment
storePrecision, 66
complexVal, 30
imag, 30
real, 30
TKspectrum.h, 144
compute_tropospheric_delays
tempo2.h, 120
computeConstraintWeights
constraints.h, 79
config.h, 76
_DARWIN_USE_64_BIT_INODE, 77
F77_FUNC, 77
F77_FUNC_, 77
HAVE_BLAS, 77
HAVE_DLERROR, 77
HAVE_DLFCN_H, 77
HAVE_FFTW3, 77
HAVE_INTTYPES_H, 77
HAVE_LAPACK, 77
HAVE_LIBDL, 77
HAVE_LIBDLLOADER, 77
HAVE_LIBM, 77
HAVE_MEMORY_H, 77
HAVE_PGPLOT, 77
HAVE_PTHREAD, 77
HAVE_STDINT_H, 77
HAVE_STDLIB_H, 77
HAVE_STRING_H, 77
HAVE_STRINGS_H, 78
HAVE_SYS_STAT_H, 78
HAVE_SYS_TYPES_H, 78

HAVE_UNISTD_H, 78
LT_OBJDIR, 78
PACKAGE, 78
PACKAGE_BUGREPORT, 78
PACKAGE_NAME, 78
PACKAGE_STRING, 78
PACKAGE_TARNAME, 78
PACKAGE_URL, 78
PACKAGE_VERSION, 78
STDC_HEADERS, 78
TEMPO2_ARCH, 78
VERSION, 78
X_DISPLAY_MISSING, 78
consFunc_dmmodel_cw
constraints.h, 79
consFunc_dmmodel_cw_year
constraints.h, 79
consFunc_dmmodel_dm1
constraints.h, 79
consFunc_dmmodel_mean
constraints.h, 80
consFunc_ifunc
constraints.h, 80
consFunc_ifunc_year
constraints.h, 80
consFunc_qifunc_c_year
constraints.h, 80
consFunc_qifunc_p_year
constraints.h, 80
consFunc_quad_ifunc_c
constraints.h, 80
consFunc_quad_ifunc_p
constraints.h, 80
consFunc_tel_dx
constraints.h, 80
consFunc_tel_dy
constraints.h, 80
consFunc_tel_dz
constraints.h, 80
constraint
tempo2.h, 116
constraint_LAST
tempo2.h, 117
constraint_dmmodel_cw_0
tempo2.h, 116
constraint_dmmodel_cw_1
tempo2.h, 116
constraint_dmmodel_cw_2
tempo2.h, 116
constraint_dmmodel_cw_3
tempo2.h, 116
constraint_dmmodel_cw_px
tempo2.h, 116
constraint_dmmodel_cw_year_cos
tempo2.h, 116
constraint_dmmodel_cw_year_cos2
tempo2.h, 116
constraint_dmmodel_cw_year_sin
tempo2.h, 116
constraint_dmmodel_cw_year_sin2
tempo2.h, 116
constraint_dmmodel_cw_year_xcos
tempo2.h, 116
constraint_dmmodel_cw_year_xsin
tempo2.h, 116
constraint_dmmodel_dm1
tempo2.h, 116
constraint_dmmodel_mean
tempo2.h, 116
constraint_ifunc_0
tempo2.h, 116
constraint_ifunc_1
tempo2.h, 116
constraint_ifunc_2
tempo2.h, 116
constraint_ifunc_year_cos
tempo2.h, 116
constraint_ifunc_year_cos2
tempo2.h, 116
constraint_ifunc_year_sin
tempo2.h, 116
constraint_ifunc_year_sin2
tempo2.h, 116
constraint_ifunc_year_xcos
tempo2.h, 116
constraint_ifunc_year_xsin
tempo2.h, 116
constraint_label
tempo2.h, 115
constraint_qifunc_c_year_cos
tempo2.h, 117
constraint_qifunc_c_year_cos2
tempo2.h, 117
constraint_qifunc_c_year_sin
tempo2.h, 117
constraint_qifunc_c_year_sin2
tempo2.h, 117
constraint_qifunc_c_year_xcos
tempo2.h, 117
constraint_qifunc_c_year_xsin
tempo2.h, 117
constraint_qifunc_p_year_cos
tempo2.h, 117
constraint_qifunc_p_year_cos2
tempo2.h, 117
constraint_qifunc_p_year_sin
tempo2.h, 116
constraint_qifunc_p_year_sin2
tempo2.h, 117
constraint_qifunc_p_year_xcos
tempo2.h, 117
constraint_qifunc_p_year_xsin
tempo2.h, 117
constraint_quad_ifunc_c_0
tempo2.h, 116
constraint_quad_ifunc_c_1

tempo2.h, 116
constraint_quad_ifunc_c_2
tempo2.h, 116
constraint_quad_ifunc_p_0
tempo2.h, 116
constraint_quad_ifunc_p_1
tempo2.h, 116
constraint_quad_ifunc_p_2
tempo2.h, 116
constraint_tel_dx_0
tempo2.h, 116
constraint_tel_dx_1
tempo2.h, 116
constraint_tel_dx_2
tempo2.h, 116
constraint_tel_dy_0
tempo2.h, 116
constraint_tel_dy_1
tempo2.h, 116
constraint_tel_dy_2
tempo2.h, 116
constraint_tel_dz_0
tempo2.h, 116
constraint_tel_dz_1
tempo2.h, 116
constraint_tel_dz_2
tempo2.h, 116
constraintCounters
FitInfo, 32
constraintDerivFunc
tempo2.h, 115
constraintDerivs
FitInfo, 32
constraintIndex
FitInfo, 32
constraints
pulsar, 54
constraints.h, 78
autosetDMCM, 79
CONSTRAINTfuncs, 80
computeConstraintWeights, 79
consFunc_dmmodel_cw, 79
consFunc_dmmodel_cw_year, 79
consFunc_dmmodel_dm1, 79
consFunc_dmmodel_mean, 80
consFunc_ifunc, 80
consFunc_ifunc_year, 80
consFunc_qifunc_c_year, 80
consFunc_qifunc_p_year, 80
consFunc_quad_ifunc_c, 80
consFunc_quad_ifunc_p, 80
consFunc_tel_dx, 80
consFunc_tel_dy, 80
consFunc_tel_dz, 80
get_constraint_name, 80
standardConstraintFunctions, 80
copyPSR
tempo2.h, 120
copyParam
tempo2.h, 120
correctTroposphere
pulsar, 54
correction
clock_correction, 30
correctionTT_TB
observation, 40
correctionTT_Teph
observation, 40
correctionUT1
observation, 40
correctionsTT
observation, 39
corrects_to
clock_correction, 30
cosl
TKlongdouble.float128.h, 137
TKlongdouble.h, 139
covar
pulsar, 54
covarFuncFile
tempo2.h, 124
curr_cache_loc
jpl_eph_data, 36

DDGRmodel
tempo2.h, 120
DDHmodel
tempo2.h, 120
DDKmodel
tempo2.h, 120
DDSmodel
tempo2.h, 120
DDmodel
tempo2.h, 120
DEPRECATED
TKlog.h, 135
DLL_FUNC
jpleph.h, 87
DM_CONST
tempo2.h, 109
DM_CONST_SI
tempo2.h, 109
dadt
GWsim.h, 83
data
DynamicArray, 30
date_string
T1Polyco, 67
dcmFile
tempo2.h, 124
debugFlag
TKlog.h, 136
decjStrPost
pulsar, 54
decjStrPre
pulsar, 54
decsim

pulsar, 54
dedt
GWsim.h, 83
defineClockCorrectionSequence
tempo2.h, 120
delayCorr
observation, 40
deleteFileName
pulsar, 54
deleted
observation, 40
destroyMemory
tempo2.h, 120
destroyOne
tempo2.h, 120
dilateFreq
pulsar, 54
dispersion_constant
ChebyModel, 28
displayCVSversion
tempo2.h, 124
displayMsg
tempo2.h, 120
displayParameters
tempo2.h, 120
dist_bin
gwSrc, 35
gwgeneralSrc, 33
dm
T1Polyco, 67
dm_delays
tempo2.h, 120
dmOffset
pulsar, 55
dmoffsCM
pulsar, 54
dmoffsCM_error
pulsar, 54
dmoffsCM_mjd
pulsar, 54
dmoffsCM_weight
pulsar, 54
dmoffsCMnum
pulsar, 54
dmoffsDM
pulsar, 55
dmoffsDM_error
pulsar, 55
dmoffsDM_mjd
pulsar, 55
dmoffsDM_weight
pulsar, 55
dmoffsDMnum
pulsar, 55
dms_turn
tempo2.h, 121
tempo2Util.h, 131
doFit
tempo2.h, 121
doFitAll
tempo2.h, 121
doFitDCM
tempo2.h, 121
doFitGlobal
tempo2.h, 121
documentation/1_USER_GUIDE.md, 80
documentation/2_developers.md, 80
documentation/3_DEVELOPER_GUIDE.md, 80
documentation/4_directories.md, 80
doppler
T1Polyco, 67
dotProduct
GWsim.h, 83
dotproduct
tempo2.h, 121
dtdt
GWsim.h, 83
DynamicArray, 30
data, 30
elem_size, 30
nalloced, 30
nelem, 31
DynamicArray_free
dynarr.h, 81
DynamicArray_init
dynarr.h, 81
DynamicArray_push_back
dynarr.h, 81
DynamicArray_resize
dynarr.h, 82
dynarr.h, 81
DynamicArray_free, 81
DynamicArray_init, 81
DynamicArray_push_back, 81
DynamicArray_resize, 82

ECLIPTIC_OBLIQUITY
tempo2.h, 124
ECLIPTIC_OBLIQUITY_VAL
tempo2.h, 109
ELL1Hmodel
tempo2.h, 121
ELL1model
tempo2.h, 121
ENDERR
TKlog.h, 135
ENDL
TKlog.h, 135
ERRORCOLOR
TKlog.h, 135
EXPSMOOTH
choleskyRoutines.h, 76
earth_ssb
observation, 40
earthMoonBary_earth
observation, 40
earthMoonBary_ssb

observation, 40
eccRes
GWsim.h, 83
eccResWithEnergy
GWsim.h, 83
eclCoord
pulsar, 55
efac
observation, 40
einsteinRate
observation, 40
elem_size
DynamicArray, 30
emrat
jpl_eph_data, 37
eopc04_file
pulsar, 55
ephem_end
jpl_eph_data, 37
ephem_start
jpl_eph_data, 37
ephem_step
jpl_eph_data, 37
ephemeris
pulsar, 55
ephemeris_version
jpl_eph_data, 37
equ2ecl
tempo2.h, 121
equad
observation, 40
err
parameter, 46

F77_FUNC
config.h, 77
F77_FUNC_
config.h, 77
FB90_TIMEEPH
tempo2.h, 109
FCALPHA
choleskyRoutines.h, 76
FCFINAL
choleskyRoutines.h, 76
FITfuncs
tempo2.h, 121
FMT_LD
TKlongdouble.float128.h, 137
TKlongdouble.h, 139
fabsl
TKlongdouble.float128.h, 137
TKlongdouble.h, 139
Fe
GWsim.h, 84
fileName
TabulatedFunction, 70
filterStr
pulsar, 55
Findphi
GWsim.h, 84
fit4
TKspectrum.h, 144
fitChisq
pulsar, 55
fitCosSineFunc
TKspectrum.h, 144
fitFlag
parameter, 46
fitFunc
pulsar, 55
FitInfo, 31
constraintCounters, 32
constraintDerivs, 32
constraintIndex, 32
nConstraints, 32
nParams, 32
paramCounters, 32
paramDerivs, 32
paramIndex, 32
tempo2.h, 115
updateFunctions, 32
fitJump
pulsar, 55
fitMeanSineFunc
TKspectrum.h, 144
fitMeanSineFunc_IFUNC
TKspectrum.h, 144
fitMode
pulsar, 55
fitNfree
pulsar, 55
fitParamGlobalI
pulsar, 55
fitParamGlobalK
pulsar, 55
fitParamI
pulsar, 55
fitParamK
pulsar, 56
fitinfo
pulsar, 55
fixedFormat
pulsar, 56
fjumpID
pulsar, 56
flagID
observation, 40
flagVal
observation, 41
floorl
TKlongdouble.float128.h, 137
TKlongdouble.h, 139
fname
observation, 41
forceGlobalFit
tempo2.h, 124
formBats

tempo2.h, 121
formBatsAll
tempo2.h, 121
formResiduals
tempo2.h, 121
fortran_mod
tempo2.h, 121
fortran_nint
tempo2.h, 121
fortran_nlong
tempo2.h, 121
free_2df
TKmatrix.h, 142
free_blas
TKmatrix.h, 142
free_uinv
TKmatrix.h, 142
freq
observation, 41
freq_end
ChebyModel, 28
freq_start
ChebyModel, 28
freqSSB
observation, 41
frequency_cheby
ChebyModel, 28
frequency_obs
T1Polyco, 67
frequency_psr_0
T1Polyco, 67

GLOBAL_OMEGA
TKspectrum.h, 145
GM
tempo2.h, 109
GM_C3
tempo2.h, 109
GMJ_C3
tempo2.h, 109
GMN_C3
tempo2.h, 109
GMS_C3
tempo2.h, 109
GMU_C3
tempo2.h, 109
GMV_C3
tempo2.h, 110
GWanisotropicbackground
GWsim.h, 84
GWbackground
GWsim.h, 84
GWbackground_read
GWsim.h, 84
GWbackground_write
GWsim.h, 84
GWdipolebackground
GWsim.h, 84
GWgeneralanisotropicbackground
GWsim.h, 84
GWgeneralbackground
GWsim.h, 84
GWgeneralbackground_read
GWsim.h, 84
GWgeneralbackground_write
GWsim.h, 84
GWsim.h, 82
calculateResidualGW, 83
calculateResidualgeneralGW, 83
dadt, 83
dedt, 83
dotProduct, 83
dtdt, 83
eccRes, 83
eccResWithEnergy, 83
Fe, 84
Findphi, 84
GWanisotropicbackground, 84
GWbackground, 84
GWbackground_read, 84
GWbackground_write, 84
GWdipolebackground, 84
GWgeneralanisotropicbackground, 84
GWgeneralbackground, 84
GWgeneralbackground_read, 84
GWgeneralbackground_write, 84
gwSrc, 83
gwgenSpec, 83
gwgeneralSrc, 83
matrixMult, 84
psrangle, 84
Rs, 84
setupGW, 84
setupPulsar_GWsim, 84
setupgeneralGW, 84
sphharm, 84
genrand_int32
T2toolkit.h, 100
genrand_real1
T2toolkit.h, 100
get_EOP
tempo2.h, 121
get_OneobsCoord
tempo2.h, 121
get_blas_cols
TKmatrix.h, 142
get_blas_rows
TKmatrix.h, 142
get_constraint_name
constraints.h, 80
get_obsCoord
tempo2.h, 121
get_obsCoord_IAU2000B
tempo2.h, 121
getCholeskyMatrix
tempo2.h, 121
getClockCorrections

tempo2.h, 121
getCorrection
tempo2.h, 121
getCorrectionTT
tempo2.h, 121
getInputs
tempo2.h, 121
getObservatory
tempo2.h, 122
getParamDeriv
tempo2.h, 122
getParameterValue
tempo2.h, 122
getprtj
TKspectrum.h, 144
getweights
TKspectrum.h, 144
globalNfit
pulsar, 56
globalNoConstrain
pulsar, 56
gwSrc, 34
across_g, 35
across_im_g, 35
aplus_g, 35
aplus_im_g, 35
dist_bin, 35
GWsim.h, 83
h, 35
h_im, 35
inc_bin, 35
kg, 35
omega_g, 35
phase_g, 35
phi_bin, 35
phi_g, 35
phi_polar_g, 35
theta_bin, 35
theta_g, 35
gwb_decj
pulsar, 56
gwb_epoch
pulsar, 56
gwb_geom_c
pulsar, 56
gwb_geom_p
pulsar, 56
gwb_raj
pulsar, 56
gwb_width
pulsar, 56
gwecc_dec
pulsar, 56
gwecc_distance
pulsar, 56
gwecc_e
pulsar, 56
gwecc_epoch
pulsar, 56
gwecc_inc
pulsar, 56
gwecc_m1
pulsar, 56
gwecc_m2
pulsar, 56
gwecc_nodes_orientation
pulsar, 56
gwecc_orbital_period
pulsar, 56
gwecc_psrdist
pulsar, 56
gwecc_pulsarTermOn
pulsar, 56
gwecc_ra
pulsar, 56
gwecc_redshift
pulsar, 56
gwecc_theta_0
pulsar, 57
gwecc_theta_nodes
pulsar, 57
gwgenSpec, 34
GWsim.h, 83
sl_alpha, 34
sl_amp, 34
st_alpha, 34
st_amp, 34
tensor_alpha, 34
tensor_amp, 34
vl_alpha, 34
vl_amp, 34
gwgeneralSrc, 32
across_g, 33
across_im_g, 33
aplus_g, 33
aplus_im_g, 33
asl_g, 33
asl_im_g, 33
ast_g, 33
ast_im_g, 33
avx_g, 33
avx_im_g, 33
avy_g, 33
avy_im_g, 33
dist_bin, 33
GWsim.h, 83
h, 33
h_im, 33
inc_bin, 33
kg, 33
omega_g, 33
phase_g, 33
phi_bin, 33
phi_g, 33
phi_polar_g, 33
theta_bin, 33

theta_g, 33
gwm_decj
pulsar, 57
gwm_dphase
pulsar, 57
gwm_epoch
pulsar, 57
gwm_phi
pulsar, 57
gwm_raj
pulsar, 57
gwsrc_across_i
pulsar, 57
gwsrc_across_i_e
pulsar, 57
gwsrc_across_r
pulsar, 57
gwsrc_across_r_e
pulsar, 57
gwsrc_aplus_i
pulsar, 57
gwsrc_aplus_i_e
pulsar, 57
gwsrc_aplus_r
pulsar, 57
gwsrc_aplus_r_e
pulsar, 57
gwsrc_dec
pulsar, 57
gwsrc_epoch
pulsar, 57
gwsrc_psrdist
pulsar, 57
gwsrc_ra
pulsar, 57

h
gwSrc, 35
gwgeneralSrc, 33
h_im
gwSrc, 35
gwgeneralSrc, 33
HAVE_BLAS
config.h, 77
HAVE_DLERROR
config.h, 77
HAVE_DLFCN_H
config.h, 77
HAVE_FFTW3
config.h, 77
HAVE_GWSIM_H
tempo2.h, 110
HAVE_INTTYPES_H
config.h, 77
HAVE_LAPACK
config.h, 77
HAVE_LIBDL
config.h, 77
HAVE_LIBDLLOADER
config.h, 77
HAVE_LIBM
config.h, 77
HAVE_MEMORY_H
config.h, 77
HAVE_PGPLOT
config.h, 77
HAVE_PTHREAD
config.h, 77
HAVE_STDINT_H
config.h, 77
HAVE_STDLIB_H
config.h, 77
HAVE_STRING_H
config.h, 77
HAVE_STRINGS_H
config.h, 78
HAVE_SYS_STAT_H
config.h, 78
HAVE_SYS_TYPES_H
config.h, 78
HAVE_UNISTD_H
config.h, 78
header_line
TabulatedFunction, 70
height_grs80
observatory, 45
hms_turn
tempo2.h, 122
tempo2Util.h, 131

IF99_TIMEEPH
tempo2.h, 110
IFTE_DeltaT
ifteph.h, 86
IFTE_DeltaTDot
ifteph.h, 86
IFTE_JD0
ifteph.h, 85
IFTE_K
ifteph.h, 85
IFTE_KM1
ifteph.h, 85
IFTE_LC
ifteph.h, 85
IFTE_MJD0
ifteph.h, 86
IFTE_TEPH0
ifteph.h, 86
IFTE_close_file
ifteph.h, 86
IFTE_get_DeltaT_DeltaTDot
ifteph.h, 86
IFTE_get_vE
ifteph.h, 86
IFTE_get_vE_vEDot
ifteph.h, 86
IFTE_get_vEDot
ifteph.h, 86

IFTE_init
  ifteph.h, 86
IFTEPH_FILE
  tempo2.h, 110
id_residual
  tempo2.h, 122
ifile
  jpl_eph_data, 37
ifteph.h, 84
  IFTE_DeltaT, 86
  IFTE_DeltaTDot, 86
  IFTE_JD0, 85
  IFTE_K, 85
  IFTE_KM1, 85
  IFTE_LC, 85
  IFTE_MJD0, 86
  IFTE_TEPH0, 86
  IFTE_close_file, 86
  IFTE_get_DeltaT_DeltaTDot, 86
  IFTE_get_vE, 86
  IFTE_get_vE_vEDot, 86
  IFTE_get_vEDot, 86
  IFTE_init, 86
ifunc
  t2fit_ifunc.h, 96
ifunc_weights
  pulsar, 57
ifuncE
  pulsar, 57
ifuncN
  pulsar, 57
ifuncT
  pulsar, 57
ifuncV
  pulsar, 57
iinfo
  jpl_eph_data, 37
imag
  complexVal, 30
inc_bin
  gwSrc, 35
  gwgeneralSrc, 33
indexx8
  TKspectrum.h, 144
init_genrand
  T2toolkit.h, 100
initialise
  tempo2.h, 122
initialiseOne
  tempo2.h, 122
interpolation_info, 35
  np, 36
  nv, 36
  pc, 36
  twot, 36
  vc, 36
ipm
  pulsar, 57
ipt
  jpl_eph_data, 37

JPL_EPHEM_AU_IN_KM
  jpleph.h, 87
JPL_EPHEM_EARTH_MOON_RATIO
  jpleph.h, 87
JPL_EPHEM_END_JD
  jpleph.h, 87
JPL_EPHEM_EPHEMERIS_VERSION
  jpleph.h, 87
JPL_EPHEM_KERNEL_NCOEFF
  jpleph.h, 87
JPL_EPHEM_KERNEL_RECORD_SIZE
  jpleph.h, 87
JPL_EPHEM_KERNEL_SIZE
  jpleph.h, 87
JPL_EPHEM_KERNEL_SWAP_BYTES
  jpleph.h, 87
JPL_EPHEM_N_CONSTANTS
  jpleph.h, 87
JPL_EPHEM_START_JD
  jpleph.h, 87
JPL_EPHEM_STEP
  jpleph.h, 87
JPL_EPHEMERIS
  pulsar, 58
JPL_HEADER_SIZE
  jpl_int.h, 86
JPLlong
  jpl_int.h, 86
JVmodel
  tempo2.h, 122
jboFormat
  pulsar, 57
jpl_close_ephemeris
  jpleph.h, 88
jpl_eph_data, 36
  au, 36
  cache, 36
  curr_cache_loc, 36
  emrat, 37
  ephem_end, 37
  ephem_start, 37
  ephem_step, 37
  ephemeris_version, 37
  ifile, 37
  iinfo, 37
  ipt, 37
  kernel_size, 37
  ncoeff, 37
  ncon, 37
  pvsun, 37
  recsize, 37
  swap_bytes, 37
jpl_get_double
  jpleph.h, 88
jpl_get_long
  jpleph.h, 88

jpl_init_ephemeris
  jpleph.h, 88
jpl_int.h, 86
  JPL_HEADER_SIZE, 86
  JPLlong, 86
  MAX_KERNEL_SIZE, 86
jpl_pleph
  jpleph.h, 88
jpl_state
  jpleph.h, 88
jpleph.h, 86
  DLL_FUNC, 87
  JPL_EPHEM_AU_IN_KM, 87
  JPL_EPHEM_EARTH_MOON_RATIO, 87
  JPL_EPHEM_END_JD, 87
  JPL_EPHEM_EPHEMERIS_VERSION, 87
  JPL_EPHEM_KERNEL_NCOEFF, 87
  JPL_EPHEM_KERNEL_RECORD_SIZE, 87
  JPL_EPHEM_KERNEL_SIZE, 87
  JPL_EPHEM_KERNEL_SWAP_BYTES, 87
  JPL_EPHEM_N_CONSTANTS, 87
  JPL_EPHEM_START_JD, 87
  JPL_EPHEM_STEP, 87
  jpl_close_ephemeris, 88
  jpl_get_double, 88
  jpl_get_long, 88
  jpl_init_ephemeris, 88
  jpl_pleph, 88
  jpl_state, 88
  make_sub_ephem, 88
jump
  observation, 41
jumpStr
  pulsar, 58
jumpVal
  pulsar, 58
jumpValErr
  pulsar, 58
jupiter_earth
  observation, 41

kernel_size
  jpl_eph_data, 37
kg
  gwSrc, 35
  gwgeneralSrc, 33
kind
  T2Predictor, 69

LD_PI
  TKlongdouble.float128.h, 137
  TKlongdouble.h, 139
  TKlongdouble.ld.h, 140
LEAPSECOND_FILE
  tempo2.h, 110
LOG_OUTFILE
  TKlog.h, 135
LONGDOUBLE_IS_FLOAT128
  TKlongdouble.float128.h, 138
  TKlongdouble.h, 139
LONGDOUBLE_IS_IEEE754
  TKlongdouble.ld.h, 141
LONGDOUBLE_ONE
  TKlongdouble.float128.h, 138
  TKlongdouble.h, 139
  TKlongdouble.ld.h, 141
LT_OBJDIR
  config.h, 78
label
  parameter, 46
  tempo2.h, 117
latitude_grs80
  observatory, 45
ld_fprintf
  TKlongdouble.float128.h, 138
  TKlongdouble.h, 139
  TKlongdouble.ld.h, 140
ld_printf
  TKlongdouble.float128.h, 138
  TKlongdouble.h, 139
  TKlongdouble.ld.h, 140
ld_sprintf
  TKlongdouble.float128.h, 138
  TKlongdouble.h, 140
  TKlongdouble.ld.h, 140
libt2toolkit API, 25
libtempo2 External API, 26
linkFrom
  parameter, 46
linkTo
  parameter, 46
log10rms
  T1Polyco, 67
logdbg
  TKlog.h, 135
logerr
  TKlog.h, 135
logerr_check
  TKlog.h, 136
logicFlag
  tempo2.h, 122
logmsg
  TKlog.h, 135
logtchk
  TKlog.h, 135
logwarn
  TKlog.h, 135
longdouble
  TKlongdouble.float128.h, 137, 138
  TKlongdouble.h, 139
  TKlongdouble.ld.h, 141
longitude_grs80
  observatory, 45
lookup_observatory_alias
  tempo2.h, 122

MASYR2RADS
  tempo2.h, 110

MAX
    TKspectrum.h, 144
MAX_BPJ_JUMPS
    tempo2.h, 110
MAX_CLK_CORR
    tempo2.h, 110
MAX_CLKCORR
    tempo2.h, 110
MAX_COEFF
    tempo2.h, 110
MAX_COMPANIONS
    tempo2.h, 110
MAX_DM_DERIVATIVES
    tempo2.h, 110
MAX_DMX
    tempo2.h, 110
MAX_FILELEN
    tempo2.h, 111
MAX_FIT
    tempo2.h, 111
MAX_FLAG_LEN
    tempo2.h, 111
MAX_FLAGS
    tempo2.h, 111
MAX_FREQ_DERIVATIVES
    tempo2.h, 111
MAX_IFUNC
    tempo2.h, 111
MAX_JUMPS
    tempo2.h, 111
MAX_KERNEL_SIZE
    jpl_int.h, 86
MAX_LEAPSEC
    tempo2.h, 111
MAX_MSG
    tempo2.h, 111
MAX_OBSN
    tempo2.h, 124
MAX_OBSN_VAL
    tempo2.h, 111
MAX_PARAMS
    tempo2.h, 111
MAX_PSR
    tempo2.h, 125
MAX_PSR_VAL
    tempo2.h, 111
MAX_QUAD
    tempo2.h, 112
MAX_SITE
    tempo2.h, 112
MAX_STOREPRECISION
    tempo2.h, 112
MAX_STRLEN
    tempo2.h, 112
MAX_T2EFAC
    tempo2.h, 112
MAX_T2EQUAD
    tempo2.h, 112
MAX_TEL_CLK_OFFS
    tempo2.h, 112
MAX_TEL_DX
    tempo2.h, 112
MAX_TEL_DY
    tempo2.h, 112
MAX_TEL_DZ
    tempo2.h, 112
MAX_TNBN
    tempo2.h, 112
MAX_TNDMEv
    tempo2.h, 112
MAX_TNECORR
    tempo2.h, 112
MAX_TNEF
    tempo2.h, 113
MAX_TNEQ
    tempo2.h, 113
MAX_TNGN
    tempo2.h, 113
MAX_TNSQ
    tempo2.h, 113
MAX_TOFFSET
    tempo2.h, 113
MAX_WHITE
    tempo2.h, 113
MIN
    TKspectrum.h, 144
MSSmodel
    tempo2.h, 122
make_sub_ephem
    jpleph.h, 88
malloc_2df
    TKmatrix.h, 142
malloc_blas
    TKmatrix.h, 142
malloc_uinv
    TKmatrix.h, 142
mat20
    TKspectrum.h, 144
matrixMult
    GWsim.h, 84
minPrec
    storePrecision, 66
mjd_end
    ChebyModel, 28
mjd_mid
    T1Polyco, 67
mjd_start
    ChebyModel, 28
modelset
    T2Predictor, 69

nCompanion
    pulsar, 58
nConstraints
    FitInfo, 32
nDMEvents
    pulsar, 58

NE_SW_DEFAULT
- tempo2.h, 113

NEWFIT
- tempo2.h, 125

NFIT
- choleskyRoutines.h, 76

nFit
- pulsar, 58

nFlags
- observation, 41

nGlobal
- pulsar, 58

nJumps
- pulsar, 58

nLinkFrom
- parameter, 46

nLinkTo
- parameter, 46

nParam
- pulsar, 59

nParams
- FitInfo, 32

nPhaseJump
- pulsar, 59

nQuad
- pulsar, 59

nStorePrecision
- pulsar, 59

nT2efac
- pulsar, 59

nT2equad
- pulsar, 59

nTNBandNoise
- pulsar, 59

nTNECORR
- pulsar, 59

nTNEF
- pulsar, 59

nTNEQ
- pulsar, 59

nTNGroupNoise
- pulsar, 59

nTNSQ
- pulsar, 59

nTNShapeletEvents
- pulsar, 59

nTelDX
- pulsar, 59

nTelDY
- pulsar, 59

nTelDZ
- pulsar, 59

nToffset
- pulsar, 59

nWhite
- pulsar, 59

nWhite_dm
- pulsar, 59

nalloced
- DynamicArray, 30

name
- observatory, 45
- pulsar, 58

nclock_correction
- observation, 41

ncoeff
- jpl_eph_data, 37
- T1Polyco, 67

ncon
- jpl_eph_data, 37

nconstraints
- pulsar, 58

ndmx
- pulsar, 58

ne_sw
- pulsar, 58

nelem
- DynamicArray, 31

neptune_earth
- observation, 41

nits
- pulsar, 58

noWarnings
- pulsar, 59

nobs
- pulsar, 59

NonePredType
- tempo2pred.h, 127

np
- interpolation_info, 36

nphase
- observation, 41

nsegments
- ChebyModelSet, 29
- T1PolycoSet, 68

nutations
- observation, 41

nv
- interpolation_info, 36

nx
- Cheby2D, 27

ny
- Cheby2D, 27

OBLQ
- tempo2.h, 113

OBSSYS_FILE
- tempo2.h, 113

obsNjump
- observation, 41

observation, 37
- addedNoise, 39
- averagebat, 39
- averageerr, 39
- averageres, 39
- bat, 39
- batCorr, 39

bbat, 39
clockCorr, 39
correctionTT_TB, 40
correctionTT_Teph, 40
correctionUT1, 40
correctionsTT, 39
delayCorr, 40
deleted, 40
earth_ssb, 40
earthMoonBary_earth, 40
earthMoonBary_ssb, 40
efac, 40
einsteinRate, 40
equad, 40
flagID, 40
flagVal, 41
fname, 41
freq, 41
freqSSB, 41
jump, 41
jupiter_earth, 41
nFlags, 41
nclock_correction, 41
neptune_earth, 41
nphase, 41
nutations, 41
obsNjump, 41
observatory_earth, 41
origErr, 41
origsat, 41
pet, 42
phase, 42
phaseOffset, 42
planet_ssb, 42
prefitResidual, 42
psrPos, 42
pulseN, 42
residual, 42
roemer, 42
sat, 42
sat_day, 42
sat_sec, 42
saturn_earth, 42
shapiroDelayJupiter, 42
shapiroDelayNeptune, 42
shapiroDelaySaturn, 43
shapiroDelaySun, 43
shapiroDelayUranus, 43
shapiroDelayVenus, 43
shklovskii, 43
siteVel, 43
sun_earth, 43
sun_ssb, 43
TNDMErr, 43
TNDMSignal, 44
TNGroupErr, 44
TNGroupSignal, 44
TNRedErr, 44
TNRedSignal, 44
tdis1, 43
tdis2, 43
telID, 43
tempo2.h, 115
toaDMErr, 44
toaErr, 44
torb, 44
tropsphericDelay, 44
uranus_earth, 44
venus_earth, 44
zenith, 44

observatory, 45
- clock_name, 45
- code, 45
- height_grs80, 45
- latitude_grs80, 45
- longitude_grs80, 45
- name, 45
- x, 45
- y, 45
- z, 45

observatory_earth
- observation, 41

obsn
- pulsar, 59

offset
- pulsar, 60

offset_e
- pulsar, 60

omega_g
- gwSrc, 35
- gwgeneralSrc, 33

open_file
- read_fortran.h, 89

open_file2
- read_fortran2.h, 90

origErr
- observation, 41

origsat
- observation, 41

outputTMatrix
- pulsar, 60

PACKAGE
- config.h, 78

PACKAGE_BUGREPORT
- config.h, 78

PACKAGE_NAME
- config.h, 78

PACKAGE_STRING
- config.h, 78

PACKAGE_TARNAME
- config.h, 78

PACKAGE_URL
- config.h, 78

PACKAGE_VERSION
- config.h, 78

PCM

tempo2.h, 113
param
pulsar, 60
param_JUMP
tempo2.h, 120
param_LAST
tempo2.h, 120
param_ZERO
tempo2.h, 120
param_a0
tempo2.h, 118
param_a1
tempo2.h, 117
param_a1dot
tempo2.h, 118
param_a2dot
tempo2.h, 118
param_afac
tempo2.h, 119
param_b0
tempo2.h, 118
param_bp
tempo2.h, 118
param_bpja1
tempo2.h, 118
param_bpjec
tempo2.h, 118
param_bpjep
tempo2.h, 118
param_bpjom
tempo2.h, 118
param_bpjpb
tempo2.h, 118
param_bpjph
tempo2.h, 118
param_bpp
tempo2.h, 118
param_brake
tempo2.h, 120
param_cgw
tempo2.h, 119
param_clk_offs
tempo2.h, 119
param_daop
tempo2.h, 119
param_decj
tempo2.h, 117
param_df1
tempo2.h, 120
param_dm
tempo2.h, 117
param_dm_cos1yr
tempo2.h, 119
param_dm_sin1yr
tempo2.h, 119
param_dmassplanet
tempo2.h, 119
param_dmepoch
tempo2.h, 117
param_dmmodel
tempo2.h, 119
param_dmx
tempo2.h, 119
param_dmxr1
tempo2.h, 119
param_dmxr2
tempo2.h, 119
param_dr
tempo2.h, 118
param_dshk
tempo2.h, 119
param_dth
tempo2.h, 118
param_dtheta
tempo2.h, 118
param_e2dot
tempo2.h, 117
param_ecc
tempo2.h, 117
param_edot
tempo2.h, 117
param_ephver
tempo2.h, 119
param_eps1
tempo2.h, 118
param_eps1dot
tempo2.h, 119
param_eps2
tempo2.h, 118
param_eps2dot
tempo2.h, 119
param_f
tempo2.h, 117
param_fb
tempo2.h, 117
param_fd
tempo2.h, 118
param_fddc
tempo2.h, 118
param_fddi
tempo2.h, 118
param_finish
tempo2.h, 118
param_gamma
tempo2.h, 118
param_glep
tempo2.h, 118
param_glf0
tempo2.h, 118
param_glf0d
tempo2.h, 118
param_glf1
tempo2.h, 118
param_glf2
tempo2.h, 118
param_glph

tempo2.h, 118
param_gltd
tempo2.h, 118
param_gwb_amp
tempo2.h, 119
param_gwecc
tempo2.h, 119
param_gwm_amp
tempo2.h, 119
param_gwsingle
tempo2.h, 119
param_h3
tempo2.h, 119
param_h4
tempo2.h, 119
param_ifunc
tempo2.h, 119
param_iperharm
tempo2.h, 119
param_kin
tempo2.h, 118
param_kom
tempo2.h, 118
param_label
tempo2.h, 115
param_m2
tempo2.h, 118
param_mtot
tempo2.h, 118
param_nharm
tempo2.h, 119
param_om
tempo2.h, 117
param_om2dot
tempo2.h, 118
param_omdot
tempo2.h, 118
param_orbpx
tempo2.h, 118
param_pb
tempo2.h, 117
param_pbdot
tempo2.h, 117
param_pepoch
tempo2.h, 117
param_pmdec
tempo2.h, 117
param_pmra
tempo2.h, 117
param_pmrv
tempo2.h, 117
param_posepoch
tempo2.h, 117
param_px
tempo2.h, 117
param_quad_ifunc_c
tempo2.h, 119
param_quad_ifunc_p
tempo2.h, 119
param_quad_om
tempo2.h, 119
param_raj
tempo2.h, 117
param_shapmax
tempo2.h, 118
param_sini
tempo2.h, 117
param_start
tempo2.h, 118
param_stateSwitchT
tempo2.h, 120
param_stig
tempo2.h, 119
param_t0
tempo2.h, 117
param_tasc
tempo2.h, 118
param_tel_dx
tempo2.h, 119
param_tel_dy
tempo2.h, 119
param_tel_dz
tempo2.h, 119
param_tel_vx
tempo2.h, 119
param_tel_vy
tempo2.h, 119
param_tel_vz
tempo2.h, 119
param_tel_x0
tempo2.h, 119
param_tel_y0
tempo2.h, 119
param_tel_z0
tempo2.h, 119
param_telEpoch
tempo2.h, 119
param_telx
tempo2.h, 119
param_tely
tempo2.h, 119
param_telz
tempo2.h, 119
param_track
tempo2.h, 118
param_tres
tempo2.h, 119
param_tspan
tempo2.h, 118
param_tzrfrq
tempo2.h, 118
param_tzrmjd
tempo2.h, 118
param_wave_dm
tempo2.h, 119
param_wave_om

tempo2.h, 118
param_waveepoch
tempo2.h, 119
param_waveepoch_dm
tempo2.h, 119
param_xomdot
tempo2.h, 118
param_xpbdot
tempo2.h, 117
paramCounters
FitInfo, 32
paramDerivFunc
tempo2.h, 115
paramDerivs
FitInfo, 32
paramIndex
FitInfo, 32
paramSet
parameter, 46
paramUpdateFunc
tempo2.h, 115
parameter, 45
aSize, 46
err, 46
fitFlag, 46
label, 46
linkFrom, 46
linkTo, 46
nLinkFrom, 46
nLinkTo, 46
paramSet, 46
prefit, 46
prefitErr, 47
shortlabel, 47
tempo2.h, 115
val, 47
parse_longdouble
TKlongdouble.float128.h, 138
TKlongdouble.h, 140
TKlongdouble.ld.h, 141
passStr
pulsar, 60
pc
interpolation_info, 36
pet
observation, 42
phase
observation, 42
phase_g
gwSrc, 35
gwgeneralSrc, 33
phaseJump
pulsar, 60
phaseJumpDir
pulsar, 60
phaseJumpID
pulsar, 60
phaseOffset
observation, 42
phi_bin
gwSrc, 35
gwgeneralSrc, 33
phi_g
gwSrc, 35
gwgeneralSrc, 33
phi_polar_g
gwSrc, 35
gwgeneralSrc, 33
planet_ssb
observation, 42
planetShapiro
pulsar, 60
polyco
tempo2.h, 122
posPulsar
pulsar, 60
preProcess
tempo2.h, 122
preProcessSimple
tempo2.h, 122
preProcessSimple1
tempo2.h, 122
preProcessSimple2
tempo2.h, 122
preProcessSimple3
tempo2.h, 122
prefit
parameter, 46
prefitErr
parameter, 47
prefitResidual
observation, 42
processFlag
tempo2.h, 122
processSimultaneous
tempo2.h, 122
psrPos
observation, 42
psrangle
GWsim.h, 84
psrname
ChebyModel, 28
T1Polyco, 67
pulsar, 47
addTNGlobalEQ, 53
auto_constraints, 53
AverageEpochWidth, 53
AverageFlag, 53
AverageResiduals, 53
binaryModel, 53
bootStrap, 53
calcShapiro, 53
cgw_angpol, 53
cgw_cosinc, 53
cgw_h0, 53
cgw_mc, 53

clk_offsE, 53
clk_offsT, 53
clk_offsV, 54
clkOffsN, 54
clock, 54
clockFromOverride, 54
constraints, 54
correctTroposphere, 54
covar, 54
decjStrPost, 54
decjStrPre, 54
decsim, 54
deleteFileName, 54
dilateFreq, 54
dmOffset, 55
dmoffsCM, 54
dmoffsCM_error, 54
dmoffsCM_mjd, 54
dmoffsCM_weight, 54
dmoffsCMnum, 54
dmoffsDM, 55
dmoffsDM_error, 55
dmoffsDM_mjd, 55
dmoffsDM_weight, 55
dmoffsDMnum, 55
eclCoord, 55
eopc04_file, 55
ephemeris, 55
filterStr, 55
fitChisq, 55
fitFunc, 55
fitJump, 55
fitMode, 55
fitNfree, 55
fitParamGlobalI, 55
fitParamGlobalK, 55
fitParamI, 55
fitParamK, 56
fitinfo, 55
fixedFormat, 56
fjumpID, 56
globalNfit, 56
globalNoConstrain, 56
gwb_decj, 56
gwb_epoch, 56
gwb_geom_c, 56
gwb_geom_p, 56
gwb_raj, 56
gwb_width, 56
gwecc_dec, 56
gwecc_distance, 56
gwecc_e, 56
gwecc_epoch, 56
gwecc_inc, 56
gwecc_m1, 56
gwecc_m2, 56
gwecc_nodes_orientation, 56
gwecc_orbital_period, 56
gwecc_psrdist, 56
gwecc_pulsarTermOn, 56
gwecc_ra, 56
gwecc_redshift, 56
gwecc_theta_0, 57
gwecc_theta_nodes, 57
gwm_decj, 57
gwm_dphase, 57
gwm_epoch, 57
gwm_phi, 57
gwm_raj, 57
gwsrc_across_i, 57
gwsrc_across_i_e, 57
gwsrc_across_r, 57
gwsrc_across_r_e, 57
gwsrc_aplus_i, 57
gwsrc_aplus_i_e, 57
gwsrc_aplus_r, 57
gwsrc_aplus_r_e, 57
gwsrc_dec, 57
gwsrc_epoch, 57
gwsrc_psrdist, 57
gwsrc_ra, 57
ifunc_weights, 57
ifuncE, 57
ifuncN, 57
ifuncT, 57
ifuncV, 57
ipm, 57
JPL_EPHEMERIS, 58
jboFormat, 57
jumpStr, 58
jumpVal, 58
jumpValErr, 58
nCompanion, 58
nDMEvents, 58
nFit, 58
nGlobal, 58
nJumps, 58
nParam, 59
nPhaseJump, 59
nQuad, 59
nStorePrecision, 59
nT2efac, 59
nT2equad, 59
nTNBandNoise, 59
nTNECORR, 59
nTNEF, 59
nTNEQ, 59
nTNGroupNoise, 59
nTNSQ, 59
nTNShapeletEvents, 59
nTelDX, 59
nTelDY, 59
nTelDZ, 59
nToffset, 59
nWhite, 59
nWhite_dm, 59

name, 58
nconstraints, 58
ndmx, 58
ne_sw, 58
nits, 58
noWarnings, 59
nobs, 59
obsn, 59
offset, 60
offset_e, 60
outputTMatrix, 60
param, 60
passStr, 60
phaseJump, 60
phaseJumpDir, 60
phaseJumpID, 60
planetShapiro, 60
posPulsar, 60
quad_across_i, 60
quad_across_i_e, 60
quad_across_r, 60
quad_across_r_e, 60
quad_aplus_i, 60
quad_aplus_i_e, 60
quad_aplus_r, 61
quad_aplus_r_e, 61
quad_ifunc_c_DEC, 61
quad_ifunc_c_RA, 61
quad_ifunc_geom_c, 61
quad_ifunc_geom_p, 61
quad_ifunc_p_DEC, 61
quad_ifunc_p_RA, 61
quad_ifuncE_c, 61
quad_ifuncE_p, 61
quad_ifuncN_c, 61
quad_ifuncN_p, 61
quad_ifuncT_c, 61
quad_ifuncT_p, 61
quad_ifuncV_c, 61
quad_ifuncV_p, 61
quadDEC, 61
quadEpoch, 61
quadRA, 61
rajStrPost, 61
rajStrPre, 61
rasim, 61
rescaleErrChisq, 61
rmsPost, 61
rmsPre, 61
robust, 61
setTelVelX, 61
setTelVelY, 62
setTelVelZ, 62
setUnits, 62
simflag, 62
sorted, 62
storePrec, 62
swm, 62
t2cMethod, 62
T2efacFlagID, 62
T2efacFlagVal, 62
T2efacVal, 62
T2equadFlagID, 62
T2equadFlagVal, 62
T2equadVal, 62
T2globalEfac, 62
TNBandDMAmp, 63
TNBandDMC, 63
TNBandDMGam, 63
TNBandNoiseAmp, 63
TNBandNoiseC, 63
TNBandNoiseGam, 63
TNBandNoiseHF, 63
TNBandNoiseLF, 63
TNDMAmp, 63
TNDMC, 63
TNDMCoeffs, 63
TNDMEvAmp, 63
TNDMEvGam, 63
TNDMEvLength, 63
TNDMEvLin, 63
TNDMEvOff, 63
TNDMEvQuad, 64
TNDMEvStart, 64
TNDMGam, 64
TNECORRFlagID, 64
TNECORRFlagVal, 64
TNECORRVal, 64
TNEFFlagID, 64
TNEFFlagVal, 64
TNEFVal, 64
TNEQFlagID, 64
TNEQFlagVal, 64
TNEQVal, 64
TNGlobalEF, 64
TNGlobalEQ, 64
TNGroupNoiseAmp, 64
TNGroupNoiseC, 64
TNGroupNoiseFlagID, 64
TNGroupNoiseFlagVal, 64
TNGroupNoiseGam, 64
TNRedAmp, 64
TNRedC, 64
TNRedCoeffs, 64
TNRedCorner, 64
TNRedFLow, 64
TNRedGam, 64
TNSQFlagID, 65
TNSQFlagVal, 65
TNSQVal, 65
TNShapeletEvFScale, 64
TNShapeletEvN, 64
TNShapeletEvPos, 64
TNShapeletEvWidth, 65
TNsubtractDM, 65
TNsubtractRed, 65

tOffset, 65
tOffset_f1, 65
tOffset_f2, 65
tOffset_t1, 65
tOffset_t2, 65
tOffsetFlags, 65
tOffsetSite, 65
telDX_e, 62
telDX_t, 62
telDX_v, 62
telDX_vel, 62
telDX_vel_e, 62
telDY_e, 62
telDY_t, 62
telDY_v, 63
telDY_vel, 63
telDY_vel_e, 63
telDZ_e, 63
telDZ_t, 63
telDZ_v, 63
telDZ_vel, 63
telDZ_vel_e, 63
tempo1, 63
tempo2.h, 115
timeEphemeris, 63
ToAextraCovar, 65
tzrsite, 65
units, 65
useCalceph, 65
useTNOrth, 65
velPulsar, 65
wave_cos, 65
wave_cos_dm, 65
wave_cos_dm_err, 65
wave_cos_err, 66
wave_sine, 66
wave_sine_dm, 66
wave_sine_dm_err, 66
wave_sine_err, 66
waveScale, 66
whiteNoiseModelFile, 66

pulseN
observation, 42

pvsun
jpl_eph_data, 37

quad_across_i
pulsar, 60

quad_across_i_e
pulsar, 60

quad_across_r
pulsar, 60

quad_across_r_e
pulsar, 60

quad_aplus_i
pulsar, 60

quad_aplus_i_e
pulsar, 60

quad_aplus_r
pulsar, 61

quad_aplus_r_e
pulsar, 61

quad_ifunc_c_DEC
pulsar, 61

quad_ifunc_c_RA
pulsar, 61

quad_ifunc_geom_c
pulsar, 61

quad_ifunc_geom_p
pulsar, 61

quad_ifunc_p_DEC
pulsar, 61

quad_ifunc_p_RA
pulsar, 61

quad_ifuncE_c
pulsar, 61

quad_ifuncE_p
pulsar, 61

quad_ifuncN_c
pulsar, 61

quad_ifuncN_p
pulsar, 61

quad_ifuncT_c
pulsar, 61

quad_ifuncT_p
pulsar, 61

quad_ifuncV_c
pulsar, 61

quad_ifuncV_p
pulsar, 61

quadDEC
pulsar, 61

quadEpoch
pulsar, 61

quadRA
pulsar, 61

README.md, 90

RESETCOLOR
TKlog.h, 135

rajStrPost
pulsar, 61

rajStrPre
pulsar, 61

rasim
pulsar, 61

read_char
read_fortran.h, 89

read_character
read_fortran.h, 89

read_character2
read_fortran2.h, 90

read_double
read_fortran.h, 89

read_double2
read_fortran2.h, 90

read_float
read_fortran.h, 89

read_float2
    read_fortran2.h, 90
read_fortran.h, 88
    c_fileptr, 89
    close_file, 89
    open_file, 89
    read_char, 89
    read_character, 89
    read_double, 89
    read_float, 89
    read_int, 89
    read_record_int, 89
    swapByte, 89
read_fortran2.h, 89
    c_fileptr2, 90
    close_file2, 90
    open_file2, 90
    read_character2, 90
    read_double2, 90
    read_float2, 90
    read_int2, 90
    read_record_int2, 90
    swapByte2, 90
read_int
    read_fortran.h, 89
read_int2
    read_fortran2.h, 90
read_record_int
    read_fortran.h, 89
read_record_int2
    read_fortran2.h, 90
readEphemeris
    tempo2.h, 122
readEphemeris_calceph
    tempo2.h, 122
readJBO_bat
    tempo2.h, 122
readObsFile
    tempo2.h, 122
readOneEphemeris
    tempo2.h, 122
readParfile
    tempo2.h, 122
readParfileGlobal
    tempo2.h, 123
readSimpleParfile
    tempo2.h, 123
readTimfile
    tempo2.h, 123
readin
    TKspectrum.h, 144
real
    complexVal, 30
recordPrecision
    tempo2.h, 123
recsize
    jpl_eph_data, 37
reference_phase
    T1Polyco, 67
rescaleErrChisq
    pulsar, 61
residual
    observation, 42
rmsPost
    pulsar, 61
rmsPre
    pulsar, 61
robust
    pulsar, 61
roemer
    observation, 42
routine
    storePrecision, 66
Rs
    GWsim.h, 84

SECDAY
    tempo2.h, 113
SECDAYl
    tempo2.h, 113
SI_UNITS
    tempo2.h, 114
SOLAR_MASS
    tempo2.h, 114
SOLAR_RADIUS
    tempo2.h, 114
SPEED_LIGHT
    tempo2.h, 114
STDC_HEADERS
    config.h, 78
samples
    TabulatedFunction, 70
sat
    observation, 42
sat_day
    observation, 42
sat_sec
    observation, 42
saturn_earth
    observation, 42
secularMotion
    tempo2.h, 123
segments
    ChebyModelSet, 29
    T1PolycoSet, 68
setPlugPath
    tempo2.h, 123
setStart
    tempo2.h, 123
setTelVelX
    pulsar, 61
setTelVelY
    pulsar, 62
setTelVelZ
    pulsar, 62
setUnits
    pulsar, 62

setupGW
    GWsim.h, 84
setupParameterFileDefaults
    tempo2.h, 123
setupPulsar_GWsim
    GWsim.h, 84
setupgeneralGW
    GWsim.h, 84
shapiro_delay
    tempo2.h, 123
shapiroDelayJupiter
    observation, 42
shapiroDelayNeptune
    observation, 42
shapiroDelaySaturn
    observation, 43
shapiroDelaySun
    observation, 43
shapiroDelayUranus
    observation, 43
shapiroDelayVenus
    observation, 43
shklovskii
    observation, 43
shortlabel
    parameter, 47
simflag
    pulsar, 62
simplePlot
    tempo2.h, 123
sineFunc
    TKspectrum.h, 144
sinfunc
    t2fit_ifunc.h, 96
sinl
    TKlongdouble.float128.h, 138
    TKlongdouble.h, 139
siteVel
    observation, 43
sitename
    ChebyModel, 28
    T1Polyco, 67
sl_alpha
    gwgenSpec, 34
sl_amp
    gwgenSpec, 34
solarWindModel
    tempo2.h, 123
sortToAs
    tempo2.h, 123
sorted
    pulsar, 62
span
    T1Polyco, 67
sphharm
    GWsim.h, 84
st_alpha
    gwgenSpec, 34
st_amp
    gwgenSpec, 34
standardConstraintFunctions
    constraints.h, 80
storePrec
    pulsar, 62
storePrecision, 66
    comment, 66
    minPrec, 66
    routine, 66
    tempo2.h, 115
sun_earth
    observation, 43
sun_ssb
    observation, 43
swap_bytes
    jpl_eph_data, 37
swapByte
    read_fortran.h, 89
swapByte2
    read_fortran2.h, 90
swm
    pulsar, 62

T1
    tempo2pred.h, 127
t1
    T2Predictor, 69
T1Polyco, 66
    binary_frequency, 67
    binary_phase, 67
    coeff, 67
    date_string, 67
    dm, 67
    doppler, 67
    frequency_obs, 67
    frequency_psr_0, 67
    log10rms, 67
    mjd_mid, 67
    ncoeff, 67
    psrname, 67
    reference_phase, 67
    sitename, 67
    span, 67
    utc_string, 67
T1Polyco_GetFrequency
    tempo2pred_int.h, 130
T1Polyco_GetPhase
    tempo2pred_int.h, 130
T1Polyco_Read
    tempo2pred_int.h, 130
T1Polyco_Write
    tempo2pred_int.h, 130
T1PolycoSet, 68
    nsegments, 68
    segments, 68
T1PolycoSet_Destroy
    tempo2pred_int.h, 130
T1PolycoSet_GetFrequency

tempo2pred_int.h, 130
T1PolycoSet_GetNearest
tempo2pred_int.h, 130
T1PolycoSet_GetPhase
tempo2pred_int.h, 130
T1PolycoSet_Read
tempo2pred_int.h, 131
T1PolycoSet_Write
tempo2pred_int.h, 131
T2_PTAmodel
tempo2.h, 123
T2C_IAU2000B
tempo2.h, 114
T2C_TEMPO
tempo2.h, 114
t2Fit
t2fit.h, 92
t2Fit_buildConstraintsMatrix
t2fit.h, 92
t2Fit_buildDesignMatrix
t2fit.h, 92
t2Fit_fillFitInfo
t2fit.h, 92
t2Fit_fillGlobalFitInfo
t2fit.h, 92
t2Fit_getFitData
t2fit.h, 92
t2Fit_updateParameters
t2fit.h, 92
t2FitFunc_binaryModels
t2fit_stdFitFuncs.h, 98
t2FitFunc_dmmodelCM
t2fit_dmmodel.h, 93
t2FitFunc_dmmodelDM
t2fit_dmmodel.h, 93
t2FitFunc_fitwaves
t2fit_fitwaves.h, 94
t2FitFunc_ifunc
t2fit_ifunc.h, 96
t2fit_stdFitFuncs.h, 98
t2FitFunc_jump
t2fit_stdFitFuncs.h, 98
t2FitFunc_miscDm
t2fit_stdFitFuncs.h, 98
t2FitFunc_planet
t2fit_stdFitFuncs.h, 98
t2FitFunc_sifunc
t2fit_ifunc.h, 96
t2FitFunc_stdDm
t2fit_stdFitFuncs.h, 98
t2FitFunc_stdFreq
t2fit_stdFitFuncs.h, 98
t2FitFunc_stdGlitch
t2fit_glitch.h, 95
t2FitFunc_stdGravWav
t2fit_stdFitFuncs.h, 98
t2FitFunc_stdPosition
t2fit_position.h, 97
t2FitFunc_telPos
t2fit_stdFitFuncs.h, 98
t2FitFunc_zero
t2fit_stdFitFuncs.h, 98
T2Predictor, 68
cheby, 69
kind, 69
modelset, 69
t1, 69
T2Predictor_Copy
tempo2pred.h, 127
T2Predictor_Destroy
tempo2pred.h, 127
T2Predictor_FRead
tempo2pred.h, 127
T2Predictor_FWrite
tempo2pred.h, 127
T2Predictor_GetEndFreq
tempo2pred.h, 127
T2Predictor_GetEndMJD
tempo2pred.h, 127
T2Predictor_GetFrequency
tempo2pred.h, 127
T2Predictor_GetPSRName
tempo2pred.h, 128
T2Predictor_GetPhase
tempo2pred.h, 127
T2Predictor_GetPlan
tempo2pred.h, 127
T2Predictor_GetPlan_Ext
tempo2pred.h, 128
T2Predictor_GetSiteName
tempo2pred.h, 128
T2Predictor_GetStartFreq
tempo2pred.h, 128
T2Predictor_GetStartMJD
tempo2pred.h, 128
T2Predictor_Init
tempo2pred.h, 128
T2Predictor_Insert
tempo2pred.h, 128
T2Predictor_Keep
tempo2pred.h, 128
T2Predictor_Kind
tempo2pred.h, 128
T2Predictor_Read
tempo2pred.h, 128
T2Predictor_Write
tempo2pred.h, 128
T2PredictorKind
tempo2pred.h, 127
t2UpdateFunc_binaryModels
t2fit_stdFitFuncs.h, 98
t2UpdateFunc_dmmodelCM
t2fit_dmmodel.h, 93
t2UpdateFunc_dmmodelDM
t2fit_dmmodel.h, 94
t2UpdateFunc_fitwaves

t2fit_fitwaves.h, 94
t2UpdateFunc_ifunc
t2fit_ifunc.h, 96
t2fit_stdFitFuncs.h, 98
t2UpdateFunc_jump
t2fit_stdFitFuncs.h, 98
t2UpdateFunc_miscDm
t2fit_stdFitFuncs.h, 98
t2UpdateFunc_planet
t2fit_stdFitFuncs.h, 98
t2UpdateFunc_simpleAdd
t2fit_stdFitFuncs.h, 98
t2UpdateFunc_simpleMinus
t2fit_stdFitFuncs.h, 99
t2UpdateFunc_stdFreq
t2fit_stdFitFuncs.h, 99
t2UpdateFunc_stdGlitch
t2fit_glitch.h, 95
t2UpdateFunc_stdGravWav
t2fit_stdFitFuncs.h, 99
t2UpdateFunc_stdPosition
t2fit_position.h, 97
t2UpdateFunc_telPos
t2fit_stdFitFuncs.h, 99
t2UpdateFunc_zero
t2fit_stdFitFuncs.h, 99
T2accel.h, 90
ACCEL_LSQ, 91
ACCEL_MULTMATRIX, 91
ACCEL_UINV, 91
accel_lsq_qr, 91
accel_multMatrix, 91
accel_multMatrixVec, 91
accel_uinv, 91
useT2accel, 91
t2cMethod
pulsar, 62
T2calculateCholesky
choleskyRoutines.h, 75
T2calculateCovarFunc
choleskyRoutines.h, 75
T2calculateDailyCovariance
choleskyRoutines.h, 75
T2calculateSpectra
choleskyRoutines.h, 75
T2cholDecomposition
choleskyRoutines.h, 75
T2cubicFit
choleskyRoutines.h, 75
T2efacFlagID
pulsar, 62
T2efacFlagVal
pulsar, 62
T2efacVal
pulsar, 62
T2equadFlagID
pulsar, 62
T2equadFlagVal
pulsar, 62
T2equadVal
pulsar, 62
T2findSmoothCurve
choleskyRoutines.h, 75
t2fit.h, 91
t2Fit, 92
t2Fit_buildConstraintsMatrix, 92
t2Fit_buildDesignMatrix, 92
t2Fit_fillFitInfo, 92
t2Fit_fillGlobalFitInfo, 92
t2Fit_getFitData, 92
t2Fit_updateParameters, 92
t2fit_dmmodel.h, 92
t2FitFunc_dmmodelCM, 93
t2FitFunc_dmmodelDM, 93
t2UpdateFunc_dmmodelCM, 93
t2UpdateFunc_dmmodelDM, 94
t2fit_fitwaves.h, 94
t2FitFunc_fitwaves, 94
t2UpdateFunc_fitwaves, 94
t2fit_glitch.h, 94
t2FitFunc_stdGlitch, 95
t2UpdateFunc_stdGlitch, 95
t2fit_ifunc.h, 95
ifunc, 96
sinfunc, 96
t2FitFunc_ifunc, 96
t2FitFunc_sifunc, 96
t2UpdateFunc_ifunc, 96
t2fit_position.h, 96
t2FitFunc_stdPosition, 97
t2UpdateFunc_stdPosition, 97
t2fit_stdFitFuncs.h, 97
t2FitFunc_binaryModels, 98
t2FitFunc_ifunc, 98
t2FitFunc_jump, 98
t2FitFunc_miscDm, 98
t2FitFunc_planet, 98
t2FitFunc_stdDm, 98
t2FitFunc_stdFreq, 98
t2FitFunc_stdGravWav, 98
t2FitFunc_telPos, 98
t2FitFunc_zero, 98
t2UpdateFunc_binaryModels, 98
t2UpdateFunc_ifunc, 98
t2UpdateFunc_jump, 98
t2UpdateFunc_miscDm, 98
t2UpdateFunc_planet, 98
t2UpdateFunc_simpleAdd, 98
t2UpdateFunc_simpleMinus, 99
t2UpdateFunc_stdFreq, 99
t2UpdateFunc_stdGravWav, 99
t2UpdateFunc_telPos, 99
t2UpdateFunc_zero, 99
T2fitSpectra
choleskyRoutines.h, 75
T2get_covFunc_automatic

choleskyRoutines.h, 75
T2getHighFreqRes
choleskyRoutines.h, 75
T2getWhiteNoiseLevel
choleskyRoutines.h, 75
T2getWhiteRes
choleskyRoutines.h, 75
T2globalEfac
pulsar, 62
T2guess_vals
choleskyRoutines.h, 75
T2interpolate
choleskyRoutines.h, 75
T2model
tempo2.h, 123
T2obtainTimingResiduals
choleskyRoutines.h, 75
T2toolkit.h, 99
genrand_int32, 100
genrand_real1, 100
init_genrand, 100
TKconvertFloat1, 100
TKconvertFloat2, 100
TKfindMax_d, 100
TKfindMax_f, 100
TKfindMedian_d, 100
TKfindMedian_f, 100
TKfindMin_d, 100
TKfindMin_f, 100
TKfindRMS_d, 100
TKfindRMS_f, 100
TKfindRMSweight_d, 100
TKgaussDev, 100
TKmean_d, 100
TKmean_f, 100
TKranDev, 100
TKrange_d, 100
TKrange_f, 100
TKretMax_d, 100
TKretMax_f, 100
TKretMin_d, 100
TKretMin_f, 101
TKretMin_i, 101
TKsetSeed, 101
TKsign_d, 101
TKsort_2f, 101
TKsort_3d, 101
TKsort_d, 101
TKsort_f, 101
TKvariance_d, 101
TKzeromean_d, 101
T2writeCovarFuncModel
choleskyRoutines.h, 76
TDB_UNITS
tempo2.h, 114
TDBTDT_FILE
tempo2.h, 114
TEMPO2_ARCH
config.h, 78
TEMPO2_ENVIRON
tempo2.h, 125
TEMPO2_ERROR
tempo2.h, 125
TEMPO2_h_HASH
tempo2.h, 114
TEMPO2_h_MAJOR_VER
tempo2.h, 114
TEMPO2_h_MINOR_VER
tempo2.h, 114
TEMPO2_h_VER
tempo2.h, 114
TK_MAX_ERROR_LEN
TKlog.h, 135
TK_MAX_ERRORS
TKlog.h, 135
TK_STORE_ERROR
TKlog.h, 135
TK_STORE_WARNING
TKlog.h, 136
TK_dft
TKspectrum.h, 145
TK_errorCount
TKlog.h, 136
TK_errorlog
TKlog.h, 136
TK_fft
TKspectrum.h, 145
TK_fitSine
TKspectrum.h, 145
TK_fitSinusoids
TKspectrum.h, 145
TK_warnCount
TKlog.h, 136
TK_warnlog
TKlog.h, 136
TK_weightLS
TKspectrum.h, 145
TKaveragePts
TKspectrum.h, 145
TKbacksubstitution_svd
TKsvd.h, 146
TKbidiagonal
TKsvd.h, 146
TKboxcar
TKspectrum.h, 145
TKcalcSigmaz
TKspectrum.h, 145
TKcholesky.h, 131
cholesky_covarFunc2matrix, 131
cholesky_dmModel, 131
cholesky_dmModelCovarParam, 131
cholesky_ecm, 131
cholesky_formUinv, 131
cholesky_powerlawModel, 132
cholesky_powerlawModel_withBeta, 132
cholesky_readFromCovarianceFunction, 132

TKcmonot
TKspectrum.h, 145
TKconstrainedLeastSquares
TKfit.h, 133
TKconvertFloat1
T2toolkit.h, 100
TKconvertFloat2
T2toolkit.h, 100
TKfindMax_d
T2toolkit.h, 100
TKfindMax_f
T2toolkit.h, 100
TKfindMedian_d
T2toolkit.h, 100
TKfindMedian_f
T2toolkit.h, 100
TKfindMin_d
T2toolkit.h, 100
TKfindMin_f
T2toolkit.h, 100
TKfindPoly_d
TKfit.h, 133
TKfindRMS_d
T2toolkit.h, 100
TKfindRMS_f
T2toolkit.h, 100
TKfindRMSweight_d
T2toolkit.h, 100
TKfirstDifference
TKspectrum.h, 145
TKfit.h, 132
TKconstrainedLeastSquares, 133
TKfindPoly_d, 133
TKfitPoly, 133
TKleastSquares, 133
TKleastSquares_svd, 133
TKleastSquares_svd_noErr, 133
TKremovePoly_d, 133
TKremovePoly_f, 133
TKrobustConstrainedLeastSquares, 133
TKrobustLeastSquares, 133
TKfitPoly
TKfit.h, 133
TKgaussDev
T2toolkit.h, 100
TKhann
TKspectrum.h, 145
TKinterpolateSplineSmoothFixedXPts
TKspectrum.h, 145
TKleastSquares
TKfit.h, 133
TKleastSquares_svd
TKfit.h, 133
TKleastSquares_svd_noErr
TKfit.h, 133
TKlog.h, 133
_LOG, 135
_TKchklog, 136
BOLDCOLOR, 135
DEPRECATED, 135
debugFlag, 136
ENDERR, 135
ENDL, 135
ERRORCOLOR, 135
LOG_OUTFILE, 135
logdbg, 135
logerr, 135
logerr_check, 136
logmsg, 135
logtchk, 135
logwarn, 135
RESETCOLOR, 135
TK_MAX_ERROR_LEN, 135
TK_MAX_ERRORS, 135
TK_STORE_ERROR, 135
TK_STORE_WARNING, 136
TK_errorCount, 136
TK_errorlog, 136
TK_warnCount, 136
TK_warnlog, 136
tcheck, 136
timer_clk, 136
WARNCOLOR, 136
WHEREARG, 136
WHEREERR, 136
WHERESTR, 136
WHERETCHK, 136
WHEREWARN, 136
writeResiduals, 136
TKlomb_d
TKspectrum.h, 145
TKlongdouble.float128.h, 136
cosl, 137
FMT_LD, 137
fabsl, 137
floorl, 137
LD_PI, 137
LONGDOUBLE_IS_FLOAT128, 138
LONGDOUBLE_ONE, 138
ld_fprintf, 138
ld_printf, 138
ld_sprintf, 138
longdouble, 137, 138
parse_longdouble, 138
sinl, 138
USE_BUILTIN_LONGDOUBLE, 138
TKlongdouble.h, 138
cosl, 139
FMT_LD, 139
fabsl, 139
floorl, 139
LD_PI, 139
LONGDOUBLE_IS_FLOAT128, 139
LONGDOUBLE_ONE, 139
ld_fprintf, 139
ld_printf, 139

ld_sprintf, 140
longdouble, 139
parse_longdouble, 140
sinl, 139
USE_BUILTIN_LONGDOUBLE, 139
TKlongdouble.ld.h, 140
LD_PI, 140
LONGDOUBLE_IS_IEEE754, 141
LONGDOUBLE_ONE, 141
ld_fprintf, 140
ld_printf, 140
ld_sprintf, 140
longdouble, 141
parse_longdouble, 141
USE_BUILTIN_LONGDOUBLE, 141
TKmatrix.h, 141
free_2df, 142
free_blas, 142
free_uinv, 142
get_blas_cols, 142
get_blas_rows, 142
malloc_2df, 142
malloc_blas, 142
malloc_uinv, 142
TKmultMatrix, 142
TKmultMatrix_sq, 142
TKmultMatrixVec, 142
TKmultMatrixVec_sq, 142
TKmean_d
T2toolkit.h, 100
TKmean_f
T2toolkit.h, 100
TKmultMatrix
TKmatrix.h, 142
TKmultMatrix_sq
TKmatrix.h, 142
TKmultMatrixVec
TKmatrix.h, 142
TKmultMatrixVec_sq
TKmatrix.h, 142
TKpythag
TKsvd.h, 146
TKranDev
T2toolkit.h, 100
TKrange_d
T2toolkit.h, 100
TKrange_f
T2toolkit.h, 100
TKremovePoly_d
TKfit.h, 133
TKremovePoly_f
TKfit.h, 133
TKretMax_d
T2toolkit.h, 100
TKretMax_f
T2toolkit.h, 100
TKretMin_d
T2toolkit.h, 100
TKretMin_f
T2toolkit.h, 101
TKretMin_i
T2toolkit.h, 101
TKrobustConstrainedLeastSquares
TKfit.h, 133
TKrobustLeastSquares
TKfit.h, 133
TKsetSeed
T2toolkit.h, 101
TKsign_d
T2toolkit.h, 101
TKsingularValueDecomposition_lsq
TKsvd.h, 146
TKsort_2f
T2toolkit.h, 101
TKsort_3d
T2toolkit.h, 101
TKsort_d
T2toolkit.h, 101
TKsort_f
T2toolkit.h, 101
TKsortit
TKspectrum.h, 145
TKspectrum
TKspectrum.h, 145
TKspectrum.h, 142
ABS, 144
calcSpectra, 144
calcSpectra_ri, 144
calcSpectra_ri_T, 144
calcSpectraErr, 144
complexVal, 144
fit4, 144
fitCosSineFunc, 144
fitMeanSineFunc, 144
fitMeanSineFunc_IFUNC, 144
GLOBAL_OMEGA, 145
getprtj, 144
getweights, 144
indexx8, 144
MAX, 144
MIN, 144
mat20, 144
readin, 144
sineFunc, 144
TK_dft, 145
TK_fft, 145
TK_fitSine, 145
TK_fitSinusoids, 145
TK_weightLS, 145
TKaveragePts, 145
TKboxcar, 145
TKcalcSigmaz, 145
TKcmonot, 145
TKfirstDifference, 145
TKhann, 145
TKinterpolateSplineSmoothFixedXPts, 145

TKlomb_d, 145
TKsortit, 145
TKspectrum, 145
TKspline_interpolate, 145
verbose_calc_spectra, 145
TKspline_interpolate
TKspectrum.h, 145
TKsvd.h, 145
TKbacksubstitution_svd, 146
TKbidiagonal, 146
TKpythag, 146
TKsingularValueDecomposition_lsq, 146
TKvariance_d
T2toolkit.h, 101
TKzeromean_d
T2toolkit.h, 101
TNBandDMAmp
pulsar, 63
TNBandDMC
pulsar, 63
TNBandDMGam
pulsar, 63
TNBandNoiseAmp
pulsar, 63
TNBandNoiseC
pulsar, 63
TNBandNoiseGam
pulsar, 63
TNBandNoiseHF
pulsar, 63
TNBandNoiseLF
pulsar, 63
TNDMAmp
pulsar, 63
TNDMC
pulsar, 63
TNDMCoeffs
pulsar, 63
TNDMErr
observation, 43
TNDMEvAmp
pulsar, 63
TNDMEvGam
pulsar, 63
TNDMEvLength
pulsar, 63
TNDMEvLin
pulsar, 63
TNDMEvOff
pulsar, 63
TNDMEvQuad
pulsar, 64
TNDMEvStart
pulsar, 64
TNDMGam
pulsar, 64
TNDMSignal
observation, 44
TNECORRFlagID
pulsar, 64
TNECORRFlagVal
pulsar, 64
TNECORRVal
pulsar, 64
TNEFFlagID
pulsar, 64
TNEFFlagVal
pulsar, 64
TNEFVal
pulsar, 64
TNEQFlagID
pulsar, 64
TNEQFlagVal
pulsar, 64
TNEQVal
pulsar, 64
TNGlobalEF
pulsar, 64
TNGlobalEQ
pulsar, 64
TNGroupErr
observation, 44
TNGroupNoiseAmp
pulsar, 64
TNGroupNoiseC
pulsar, 64
TNGroupNoiseFlagID
pulsar, 64
TNGroupNoiseFlagVal
pulsar, 64
TNGroupNoiseGam
pulsar, 64
TNGroupSignal
observation, 44
TNRedAmp
pulsar, 64
TNRedC
pulsar, 64
TNRedCoeffs
pulsar, 64
TNRedCorner
pulsar, 64
TNRedErr
observation, 44
TNRedFLow
pulsar, 64
TNRedGam
pulsar, 64
TNRedSignal
observation, 44
TNSQFlagID
pulsar, 65
TNSQFlagVal
pulsar, 65
TNSQVal
pulsar, 65

TNShapeletEvFScale
pulsar, 64
TNShapeletEvN
pulsar, 64
TNShapeletEvPos
pulsar, 64
TNShapeletEvWidth
pulsar, 65
TNsubtractDM
pulsar, 65
TNsubtractRed
pulsar, 65
tOffset
pulsar, 65
tOffset_f1
pulsar, 65
tOffset_f2
pulsar, 65
tOffset_t1
pulsar, 65
tOffset_t2
pulsar, 65
tOffsetFlags
pulsar, 65
tOffsetSite
pulsar, 65
TSUN
tempo2.h, 114
TabulatedFunction, 70
fileName, 70
header_line, 70
samples, 70
TabulatedFunction_getEndX
tabulatedfunction.h, 102
TabulatedFunction_getStartX
tabulatedfunction.h, 102
TabulatedFunction_getValue
tabulatedfunction.h, 102
TabulatedFunction_load
tabulatedfunction.h, 102
TabulatedFunctionSample, 70
x, 70
y, 70
tabulatedfunction.h, 101
TabulatedFunction_getEndX, 102
TabulatedFunction_getStartX, 102
TabulatedFunction_getValue, 102
TabulatedFunction_load, 102
tai2tt
tempo2.h, 123
tai2ut1
tempo2.h, 123
tcheck
TKlog.h, 136
tdis1
observation, 43
tdis2
observation, 43
telDX_e
pulsar, 62
telDX_t
pulsar, 62
telDX_v
pulsar, 62
telDX_vel
pulsar, 62
telDX_vel_e
pulsar, 62
telDY_e
pulsar, 62
telDY_t
pulsar, 62
telDY_v
pulsar, 63
telDY_vel
pulsar, 63
telDY_vel_e
pulsar, 63
telDZ_e
pulsar, 63
telDZ_t
pulsar, 63
telDZ_v
pulsar, 63
telDZ_vel
pulsar, 63
telDZ_vel_e
pulsar, 63
telID
observation, 43
tempo1
pulsar, 63
tempo2.h, 102
AU_DIST, 109
AULTSC, 109
allocateMemory, 120
autoConstraints, 120
BIG_G, 109
BTJmodel, 120
BTXmodel, 120
BTmodel, 120
bootstrap, 120
CVSdisplayVersion, 120
calcRMS, 120
calculate_bclt, 120
compute_tropospheric_delays, 120
constraint, 116
constraint_LAST, 117
constraint_dmmodel_cw_0, 116
constraint_dmmodel_cw_1, 116
constraint_dmmodel_cw_2, 116
constraint_dmmodel_cw_3, 116
constraint_dmmodel_cw_px, 116
constraint_dmmodel_cw_year_cos, 116
constraint_dmmodel_cw_year_cos2, 116
constraint_dmmodel_cw_year_sin, 116

constraint_dmmodel_cw_year_sin2, 116
constraint_dmmodel_cw_year_xcos, 116
constraint_dmmodel_cw_year_xsin, 116
constraint_dmmodel_dm1, 116
constraint_dmmodel_mean, 116
constraint_ifunc_0, 116
constraint_ifunc_1, 116
constraint_ifunc_2, 116
constraint_ifunc_year_cos, 116
constraint_ifunc_year_cos2, 116
constraint_ifunc_year_sin, 116
constraint_ifunc_year_sin2, 116
constraint_ifunc_year_xcos, 116
constraint_ifunc_year_xsin, 116
constraint_label, 115
constraint_qifunc_c_year_cos, 117
constraint_qifunc_c_year_cos2, 117
constraint_qifunc_c_year_sin, 117
constraint_qifunc_c_year_sin2, 117
constraint_qifunc_c_year_xcos, 117
constraint_qifunc_c_year_xsin, 117
constraint_qifunc_p_year_cos, 117
constraint_qifunc_p_year_cos2, 117
constraint_qifunc_p_year_sin, 116
constraint_qifunc_p_year_sin2, 117
constraint_qifunc_p_year_xcos, 117
constraint_qifunc_p_year_xsin, 117
constraint_quad_ifunc_c_0, 116
constraint_quad_ifunc_c_1, 116
constraint_quad_ifunc_c_2, 116
constraint_quad_ifunc_p_0, 116
constraint_quad_ifunc_p_1, 116
constraint_quad_ifunc_p_2, 116
constraint_tel_dx_0, 116
constraint_tel_dx_1, 116
constraint_tel_dx_2, 116
constraint_tel_dy_0, 116
constraint_tel_dy_1, 116
constraint_tel_dy_2, 116
constraint_tel_dz_0, 116
constraint_tel_dz_1, 116
constraint_tel_dz_2, 116
constraintDerivFunc, 115
copyPSR, 120
copyParam, 120
covarFuncFile, 124
DDGRmodel, 120
DDHmodel, 120
DDKmodel, 120
DDSmodel, 120
DDmodel, 120
DM_CONST, 109
DM_CONST_SI, 109
dcmFile, 124
defineClockCorrectionSequence, 120
destroyMemory, 120
destroyOne, 120
displayCVSversion, 124
displayMsg, 120
displayParameters, 120
dm_delays, 120
dms_turn, 121
doFit, 121
doFitAll, 121
doFitDCM, 121
doFitGlobal, 121
dotproduct, 121
ECLIPTIC_OBLIQUITY, 124
ECLIPTIC_OBLIQUITY_VAL, 109
ELL1Hmodel, 121
ELL1model, 121
equ2ecl, 121
FB90_TIMEEPH, 109
FITfuncs, 121
FitInfo, 115
forceGlobalFit, 124
formBats, 121
formBatsAll, 121
formResiduals, 121
fortran_mod, 121
fortran_nint, 121
fortran_nlong, 121
GM, 109
GM_C3, 109
GMJ_C3, 109
GMN_C3, 109
GMS_C3, 109
GMU_C3, 109
GMV_C3, 110
get_EOP, 121
get_OneobsCoord, 121
get_obsCoord, 121
get_obsCoord_IAU2000B, 121
getCholeskyMatrix, 121
getClockCorrections, 121
getCorrection, 121
getCorrectionTT, 121
getInputs, 121
getObservatory, 122
getParamDeriv, 122
getParameterValue, 122
HAVE_GWSIM_H, 110
hms_turn, 122
IF99_TIMEEPH, 110
IFTEPH_FILE, 110
id_residual, 122
initialise, 122
initialiseOne, 122
JVmodel, 122
LEAPSECOND_FILE, 110
label, 117
logicFlag, 122
lookup_observatory_alias, 122
MASYR2RADS, 110
MAX_BPJ_JUMPS, 110
MAX_CLK_CORR, 110

MAX_CLKCORR, 110
MAX_COEFF, 110
MAX_COMPANIONS, 110
MAX_DM_DERIVATIVES, 110
MAX_DMX, 110
MAX_FILELEN, 111
MAX_FIT, 111
MAX_FLAG_LEN, 111
MAX_FLAGS, 111
MAX_FREQ_DERIVATIVES, 111
MAX_IFUNC, 111
MAX_JUMPS, 111
MAX_LEAPSEC, 111
MAX_MSG, 111
MAX_OBSN, 124
MAX_OBSN_VAL, 111
MAX_PARAMS, 111
MAX_PSR, 125
MAX_PSR_VAL, 111
MAX_QUAD, 112
MAX_SITE, 112
MAX_STOREPRECISION, 112
MAX_STRLEN, 112
MAX_T2EFAC, 112
MAX_T2EQUAD, 112
MAX_TEL_CLK_OFFS, 112
MAX_TEL_DX, 112
MAX_TEL_DY, 112
MAX_TEL_DZ, 112
MAX_TNBN, 112
MAX_TNDMEv, 112
MAX_TNECORR, 112
MAX_TNEF, 113
MAX_TNEQ, 113
MAX_TNGN, 113
MAX_TNSQ, 113
MAX_TOFFSET, 113
MAX_WHITE, 113
MSSmodel, 122
NE_SW_DEFAULT, 113
NEWFIT, 125
OBLQ, 113
OBSSYS_FILE, 113
observation, 115
PCM, 113
param_JUMP, 120
param_LAST, 120
param_ZERO, 120
param_a0, 118
param_a1, 117
param_a1dot, 118
param_a2dot, 118
param_afac, 119
param_b0, 118
param_bp, 118
param_bpja1, 118
param_bpjec, 118
param_bpjep, 118
param_bpjom, 118
param_bpjpb, 118
param_bpjph, 118
param_bpp, 118
param_brake, 120
param_cgw, 119
param_clk_offs, 119
param_daop, 119
param_decj, 117
param_df1, 120
param_dm, 117
param_dm_cos1yr, 119
param_dm_sin1yr, 119
param_dmassplanet, 119
param_dmepoch, 117
param_dmmodel, 119
param_dmx, 119
param_dmxr1, 119
param_dmxr2, 119
param_dr, 118
param_dshk, 119
param_dth, 118
param_dtheta, 118
param_e2dot, 117
param_ecc, 117
param_edot, 117
param_ephver, 119
param_eps1, 118
param_eps1dot, 119
param_eps2, 118
param_eps2dot, 119
param_f, 117
param_fb, 117
param_fd, 118
param_fddc, 118
param_fddi, 118
param_finish, 118
param_gamma, 118
param_glep, 118
param_glf0, 118
param_glf0d, 118
param_glf1, 118
param_glf2, 118
param_glph, 118
param_gltd, 118
param_gwb_amp, 119
param_gwecc, 119
param_gwm_amp, 119
param_gwsingle, 119
param_h3, 119
param_h4, 119
param_ifunc, 119
param_iperharm, 119
param_kin, 118
param_kom, 118
param_label, 115
param_m2, 118
param_mtot, 118

param_nharm, 119
param_om, 117
param_om2dot, 118
param_omdot, 118
param_orbpx, 118
param_pb, 117
param_pbdot, 117
param_pepoch, 117
param_pmdec, 117
param_pmra, 117
param_pmrv, 117
param_posepoch, 117
param_px, 117
param_quad_ifunc_c, 119
param_quad_ifunc_p, 119
param_quad_om, 119
param_raj, 117
param_shapmax, 118
param_sini, 117
param_start, 118
param_stateSwitchT, 120
param_stig, 119
param_t0, 117
param_tasc, 118
param_tel_dx, 119
param_tel_dy, 119
param_tel_dz, 119
param_tel_vx, 119
param_tel_vy, 119
param_tel_vz, 119
param_tel_x0, 119
param_tel_y0, 119
param_tel_z0, 119
param_telEpoch, 119
param_telx, 119
param_tely, 119
param_telz, 119
param_track, 118
param_tres, 119
param_tspan, 118
param_tzrfrq, 118
param_tzrmjd, 118
param_wave_dm, 119
param_wave_om, 118
param_waveepoch, 119
param_waveepoch_dm, 119
param_xomdot, 118
param_xpbdot, 117
paramDerivFunc, 115
paramUpdateFunc, 115
parameter, 115
polyco, 122
preProcess, 122
preProcessSimple, 122
preProcessSimple1, 122
preProcessSimple2, 122
preProcessSimple3, 122
processFlag, 122
processSimultaneous, 122
pulsar, 115
readEphemeris, 122
readEphemeris_calceph, 122
readJBO_bat, 122
readObsFile, 122
readOneEphemeris, 122
readParfile, 122
readParfileGlobal, 123
readSimpleParfile, 123
readTimfile, 123
recordPrecision, 123
SECDAY, 113
SECDAYl, 113
SI_UNITS, 114
SOLAR_MASS, 114
SOLAR_RADIUS, 114
SPEED_LIGHT, 114
secularMotion, 123
setPlugPath, 123
setStart, 123
setupParameterFileDefaults, 123
shapiro_delay, 123
simplePlot, 123
solarWindModel, 123
sortToAs, 123
storePrecision, 115
T2_PTAmodel, 123
T2C_IAU2000B, 114
T2C_TEMPO, 114
T2model, 123
TDB_UNITS, 114
TDBTDT_FILE, 114
TEMPO2_ENVIRON, 125
TEMPO2_ERROR, 125
TEMPO2_h_HASH, 114
TEMPO2_h_MAJOR_VER, 114
TEMPO2_h_MINOR_VER, 114
TEMPO2_h_VER, 114
TSUN, 114
tai2tt, 123
tai2ut1, 123
tempo2_plug_path, 125
tempo2_plug_path_len, 125
tempo2MachineType, 125
textOutput, 123
toa2utc, 123
transform_units, 123
tt2tb, 123
turn_deg, 123
turn_dms, 123
turn_hms, 123
UT1_FILE, 114
updateBT, 123
updateBTJ, 123
updateBTX, 123
updateBatsAll, 123
updateDD, 124

  - updateDDGR, 124
  - updateDDH, 124
  - updateDDK, 124
  - updateDDS, 124
  - updateELL1, 124
  - updateELL1H, 124
  - updateJV, 124
  - updateMSS, 124
  - updateParameters, 124
  - updateT2, 124
  - updateT2_PTA, 124
  - useSelectFile, 124
  - utc2tai, 124
  - vectorPulsar, 124
  - vectorscale, 124
  - vectorsum, 124
  - veryFast, 125
  - writeTim, 124
  - zoom_graphics, 124
- tempo2_plug_path
  - tempo2.h, 125
- tempo2_plug_path_len
  - tempo2.h, 125
- tempo2MachineType
  - tempo2.h, 125
- tempo2Util.h, 131
  - dms_turn, 131
  - hms_turn, 131
  - turn_deg, 131
- tempo2pred.h, 126
  - Cheby, 127
  - ChebyModelSet_OutOfRange, 128
  - NonePredType, 127
  - T1, 127
  - T2Predictor_Copy, 127
  - T2Predictor_Destroy, 127
  - T2Predictor_FRead, 127
  - T2Predictor_FWrite, 127
  - T2Predictor_GetEndFreq, 127
  - T2Predictor_GetEndMJD, 127
  - T2Predictor_GetFrequency, 127
  - T2Predictor_GetPSRName, 128
  - T2Predictor_GetPhase, 127
  - T2Predictor_GetPlan, 127
  - T2Predictor_GetPlan_Ext, 128
  - T2Predictor_GetSiteName, 128
  - T2Predictor_GetStartFreq, 128
  - T2Predictor_GetStartMJD, 128
  - T2Predictor_Init, 128
  - T2Predictor_Insert, 128
  - T2Predictor_Keep, 128
  - T2Predictor_Kind, 128
  - T2Predictor_Read, 128
  - T2Predictor_Write, 128
  - T2PredictorKind, 127
- tempo2pred_int.h, 128
  - Cheby2D_Construct, 129
  - Cheby2D_Construct_x_Derivative, 129
  - Cheby2D_Test, 129
  - ChebyModel_Construct, 129
  - ChebyModel_Copy, 129
  - ChebyModel_Destroy, 130
  - ChebyModel_GetFrequency, 130
  - ChebyModel_GetPhase, 130
  - ChebyModel_Init, 130
  - ChebyModel_Read, 130
  - ChebyModel_Test, 130
  - ChebyModel_Write, 130
  - ChebyModelSet_Construct, 130
  - ChebyModelSet_Destroy, 130
  - ChebyModelSet_GetFrequency, 130
  - ChebyModelSet_GetNearest, 130
  - ChebyModelSet_GetPhase, 130
  - ChebyModelSet_Init, 130
  - ChebyModelSet_Insert, 130
  - ChebyModelSet_Keep, 130
  - ChebyModelSet_Read, 130
  - ChebyModelSet_Test, 130
  - ChebyModelSet_Write, 130
  - T1Polyco_GetFrequency, 130
  - T1Polyco_GetPhase, 130
  - T1Polyco_Read, 130
  - T1Polyco_Write, 130
  - T1PolycoSet_Destroy, 130
  - T1PolycoSet_GetFrequency, 130
  - T1PolycoSet_GetNearest, 130
  - T1PolycoSet_GetPhase, 130
  - T1PolycoSet_Read, 131
  - T1PolycoSet_Write, 131
- tensor_alpha
  - gwgenSpec, 34
- tensor_amp
  - gwgenSpec, 34
- textOutput
  - tempo2.h, 123
- theta_bin
  - gwSrc, 35
  - gwgeneralSrc, 33
- theta_g
  - gwSrc, 35
  - gwgeneralSrc, 33
- timeEphemeris
  - pulsar, 63
- timer_clk
  - TKlog.h, 136
- ToAextraCovar
  - pulsar, 65
- toa2utc
  - tempo2.h, 123
- toaDMErr
  - observation, 44
- toaErr
  - observation, 44
- torb
  - observation, 44
- transform_units

tempo2.h, 123
troposphericDelay
observation, 44
tt2tb
tempo2.h, 123
turn_deg
tempo2.h, 123
tempo2Util.h, 131
turn_dms
tempo2.h, 123
turn_hms
tempo2.h, 123
twot
interpolation_info, 36
tzrsite
pulsar, 65

UPW
choleskyRoutines.h, 76
USE_BUILTIN_LONGDOUBLE
TKlongdouble.float128.h, 138
TKlongdouble.h, 139
TKlongdouble.ld.h, 141
UT1_FILE
tempo2.h, 114
units
pulsar, 65
updateBT
tempo2.h, 123
updateBTJ
tempo2.h, 123
updateBTX
tempo2.h, 123
updateBatsAll
tempo2.h, 123
updateDD
tempo2.h, 124
updateDDGR
tempo2.h, 124
updateDDH
tempo2.h, 124
updateDDK
tempo2.h, 124
updateDDS
tempo2.h, 124
updateELL1
tempo2.h, 124
updateELL1H
tempo2.h, 124
updateFunctions
FitInfo, 32
updateJV
tempo2.h, 124
updateMSS
tempo2.h, 124
updateParameters
tempo2.h, 124
updateT2
tempo2.h, 124
updateT2_PTA
tempo2.h, 124
uranus_earth
observation, 44
useCalceph
pulsar, 65
useSelectFile
tempo2.h, 124
useT2accel
T2accel.h, 91
useTNOrth
pulsar, 65
utc2tai
tempo2.h, 124
utc_string
T1Polyco, 67

VERSION
config.h, 78
val
parameter, 47
vc
interpolation_info, 36
vectorPulsar
tempo2.h, 124
vectorscale
tempo2.h, 124
vectorsum
tempo2.h, 124
velPulsar
pulsar, 65
venus_earth
observation, 44
verbose_calc_spectra
TKspectrum.h, 145
veryFast
tempo2.h, 125
vl_alpha
gwgenSpec, 34
vl_amp
gwgenSpec, 34

WARNCOLOR
TKlog.h, 136
WHEREARG
TKlog.h, 136
WHEREERR
TKlog.h, 136
WHERESTR
TKlog.h, 136
WHERETCHK
TKlog.h, 136
WHEREWARN
TKlog.h, 136
WNLEVEL
choleskyRoutines.h, 76
wave_cos
pulsar, 65
wave_cos_dm

pulsar, 65
wave_cos_dm_err
pulsar, 65
wave_cos_err
pulsar, 66
wave_sine
pulsar, 66
wave_sine_dm
pulsar, 66
wave_sine_dm_err
pulsar, 66
wave_sine_err
pulsar, 66
waveScale
pulsar, 66
whiteNoiseModelFile
pulsar, 66
writeResiduals
TKlog.h, 136
writeTim
tempo2.h, 124

x
observatory, 45
TabulatedFunctionSample, 70
X_DISPLAY_MISSING
config.h, 78

y
observatory, 45
TabulatedFunctionSample, 70

z
observatory, 45
zenith
observation, 44
zoom_graphics
tempo2.h, 124