Fits to the tachocline

Jørgen Christensen-Dalsgaard Stellar Astrophysics Centre;
and Department of Physics and Astronomy, Aarhus University
22 May, 2023

1 Introduction

An important outcome of the helioseismic rotation investigations is to characterize the properties of the tachocline. Here I present some simple experiments to do so, emphasizing results for the artificial data and considering a variety of data sets and inversion methods.

2 Fitting techniques

The goal of the analysis is, given the inferred rotation rate $\Omega(\hat{r}, \theta)/2\pi$, to fit a function $\mathcal{T}(\hat{r}, \theta, \{p_i\})$ in the vicinity of the tachocline, suitably parametrized in terms of the parameters $\{p_i\}$. The fit is carried out as a function of $\hat{r}$ separately for selected values of colatitude θ, such that the parameters are determined as function $p_i(\theta)$. The fit is carried out using the `Python` function `curve_fit` from the package `scipy.optimize`, basing the weights in the fit on the inferred standard deviation of $\Omega/2\pi$. In the fit, the inversion results on the interval $[0.4, 0.95]$ in $\hat{r}$ were typically used.

To investigate the statistics of the fit a simple Monte-Carlo (MC) analysis was used. This was carried out by adding to the inversion results $\Omega/2\pi$, before the fits, Gaussian random errors with the inferred standard deviations $\sigma(\Omega)/2\pi$ of the inversion results, for a suitable number $N_{\rm MC}$ of samples (somewhat limited by the fact that this has so far been run on my laptop). Based on the resulting fits I have calculated the unweighted averages and standard deviations of the fitting parameters. In some analyses I have considered both $N_{\rm MC} = 500$ and 2000 samples, to obtain a reasonable estimate of the distributions. In other cases, scanning a range of inversion parameters, fewer samples were used, testing also the dependence of the inferred standard deviations on $N_{\rm MC}$.

3 Testing the fits for Model 2

As a first example I have considered fitting inversion results for artificial data based on Model 2 for the rotation rate, as set up by Sasha Kosovichev. The rotation rate is shown in Fig. 1. This has the advantage of being characterized by well-defined parameters, from which the selection of fitting parameters $\{p_i\}$ can be made. Here I use the location $\hat{r}_{\rm c}$ of the tachocline, its width w, the rotation rate $\hat{\Omega}_{\rm c}$ below the tachocline and the rotation rates $\hat{\Omega}_1$ and $\hat{\Omega}_2$ just above the tachocline and near the surface. The values for the model rotation rate are listed in Table 1. Figure 2 shows the effects of the rotation rate at the equator of changes to these parameters. Further details on the artificial rotation rate and similar plots at a selection of latitudes are provided in Appendix A.

As a sanity check I have confirmed that fitting the original Model 2 rotation rate, starting with somewhat perturbed parameters, returns the original values. For this test, uniform weights were applied in the fits.

In the following I show a few examples of the results of such fits, generally focusing on the equatorial behaviour. Full results at a selection of latitudes are provided in appendices.

Table 1: Parameters characterizing the Model 2 rotation rate. $\hat{\Omega}_c$, $\hat{\Omega}_1$ and $\hat{\Omega}_2$ correspond to cyclic rotation frequencies $\Omega/2\pi$.

$\hat{r}_c$	w	$\hat{\Omega}_c$	$\hat{\Omega}_1$	$\hat{\Omega}_2$
		(nHz)	(nHz)	(nHz)
0.695	0.05	430	460	475

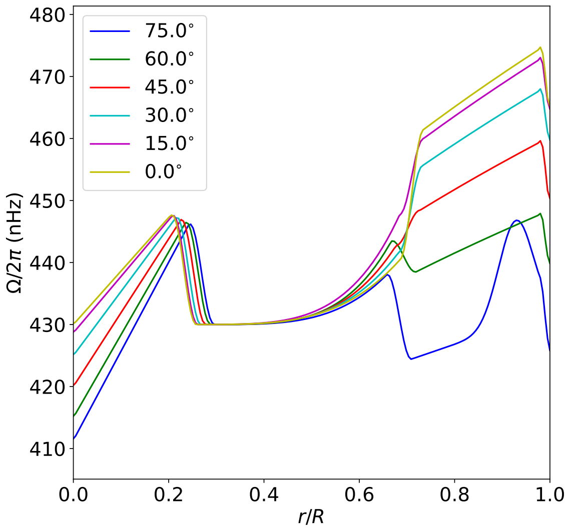


Figure 1: The input Model 2 rotation rate.

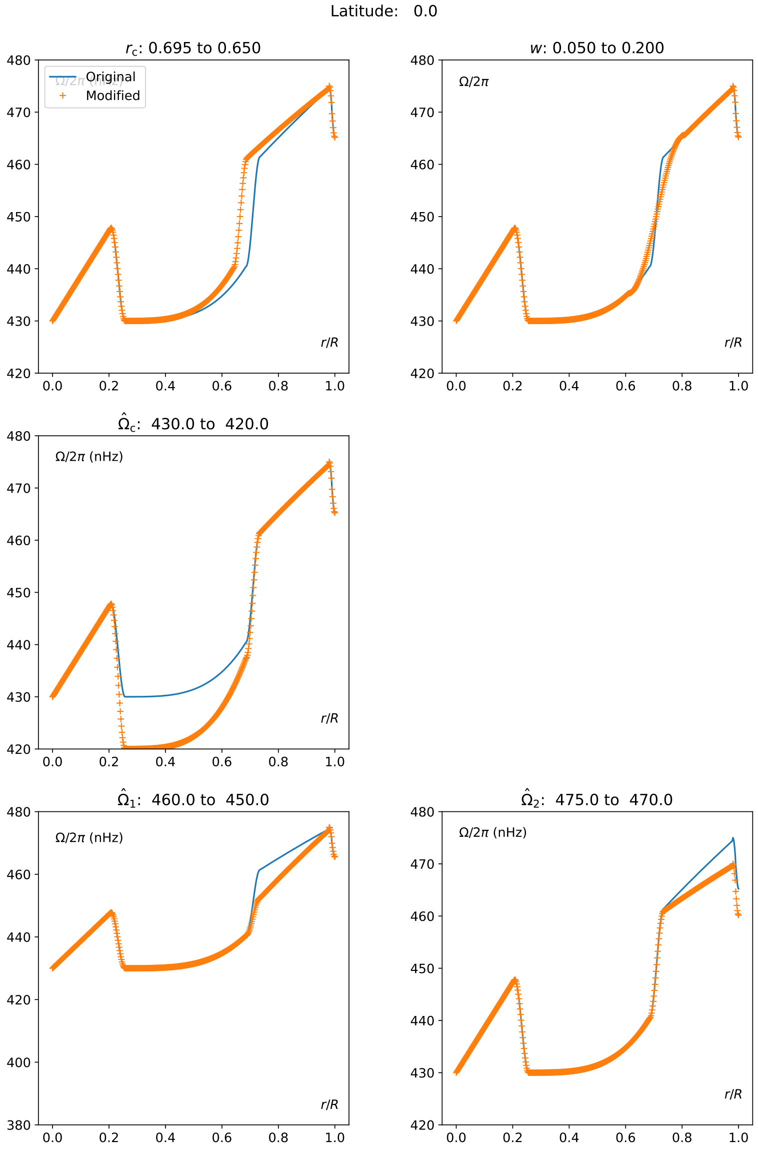


Figure 2: Effects on the equatorial rotation rate of changes to the five parameters $\{p_i\}$ chosen to represent the Model 2 rotation rate. The original values are as provided in Table 1, and the panel headers list the modified and original values.

3.1 RLS inversion of JS a coefficients.

To illustrate the statistical properties of the fit I consider a somewhat randomly selected case from Rachel's RLS inversions of JS a coefficients. Here 2000 MC samples were used. The data used in the inversion were JS a coefficients, with no added errors. Figure 3 shows the inversion results with the fitted function superimposed, while Table 2 lists the inferred parameters with their standard deviations. It is clear that the fit provides a reasonable match to the inferred rotation rate, with parameters that are similar to the input parameters (*cf.* Table 1), although with some variation with latitude.

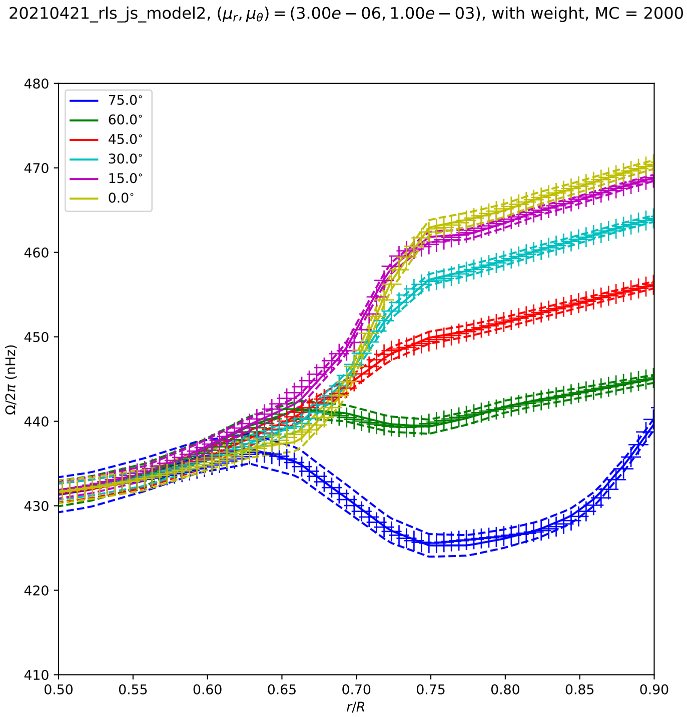


Figure 3: RLS inversion results for Model 2 artificial data, in the form of JS a coefficients, with inversion parameters $\mu_r = 3 \times 10^{-6}, \mu_\theta = 10^{-3}$. The solid lines show the inversion results, whereas the dashed lines indicate the $\pm 1\sigma$ standard deviations. The plusses show the fitted functions, based on the parameters in Table 2. $N_{\rm MC} = 2000$ Monte-Carlo samples were used.

Table 2: Inferred parameters $\{p_i\}$ for the fit illustrated in Fig. 3.

Latitude	$\hat{r}_{\rm c}$	w	$\hat{\Omega}_{\rm c}$	$\hat{\Omega}_1$	$\hat{\Omega}_2$
(Degree)			(nHz)	(nHz)	(nHz)
75.0	0.7031 ± 0.0031	0.1192 ± 0.0110	430.82 ± 0.34	448.88 ± 2.87	458.62 ± 4.11
60.0	0.6971 ± 0.0041	0.0831 ± 0.0080	430.25 ± 0.32	460.14 ± 0.75	474.97 ± 1.20
45.0	0.6959 ± 0.0032	0.0526 ± 0.0200	430.02 ± 0.24	460.11 ± 0.23	475.11 ± 0.50
30.0	0.6899 ± 0.0017	0.1040 ± 0.0570	430.13 ± 0.18	459.66 ± 0.25	475.08 ± 0.46
15.0	0.6972 ± 0.0011	0.0475 ± 0.0120	429.76 ± 0.19	460.31 ± 0.18	474.91 ± 0.16
0.0	0.6945 ± 0.0010	0.0949 ± 0.0080	430.25 ± 0.22	460.17 ± 0.31	474.91 ± 0.19

The statistical properties of the fit are further illustrated in Fig. 4, which shows histograms

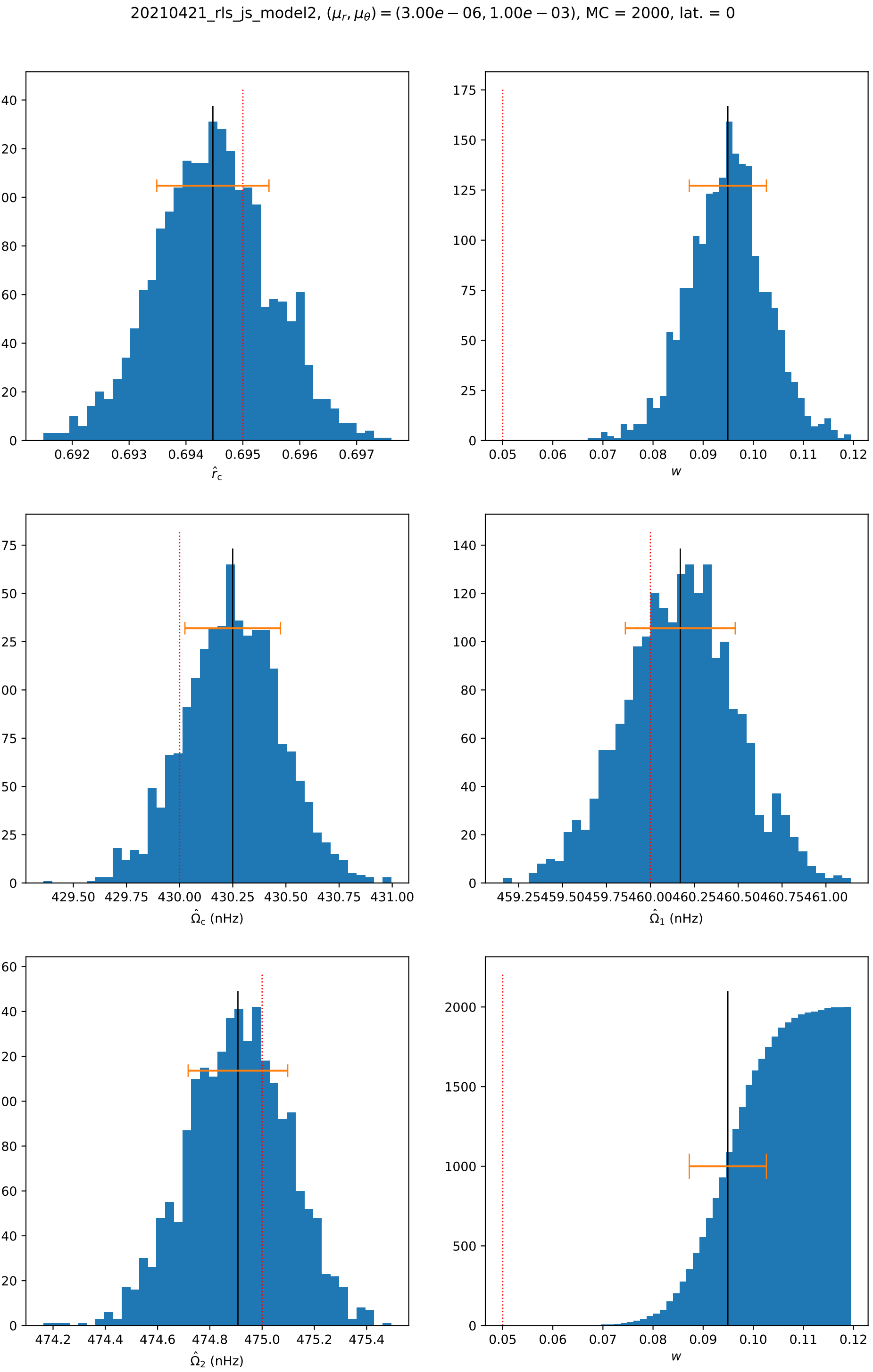


Figure 4: Histograms of the inferred parameters $\{p_i\}$ for the fit illustrated in Fig. 3. The vertical black lines show the means of the distributions, and the horizontal error bars illustrate the standard deviation. The vertical dotted lines show the true values of the parameters. The lower right panel shows the cumulated distribution for the width w.

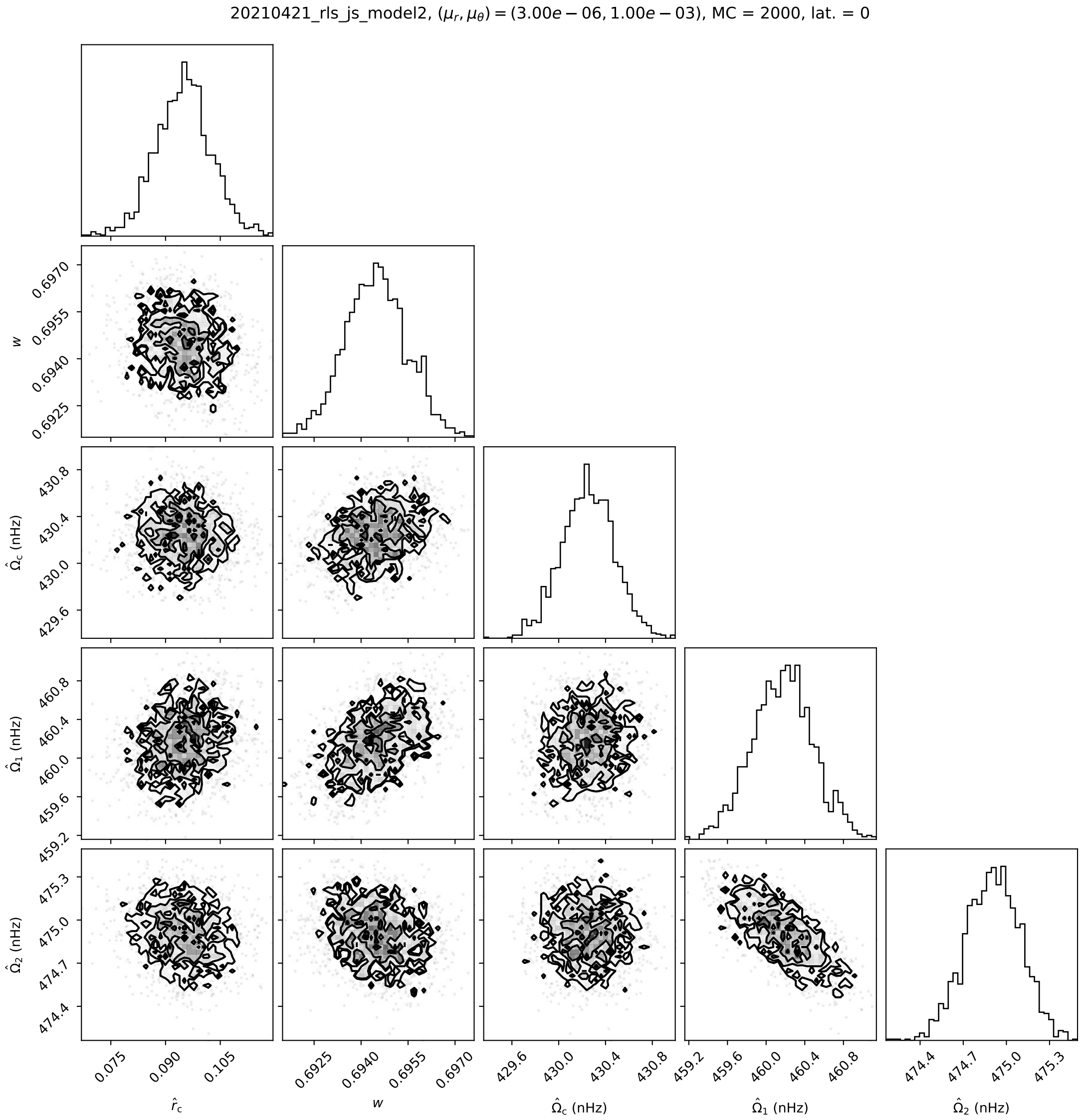


Figure 5: Corner plots of the inferred parameters $\{p_i\}$ for the fit illustrated in Fig. 3.

of the fitted parameters at the equator. Plots for all latitudes are provided in Section B, Figs B1 to B6. Correlations between the parameters at the equator are illustrated in Fig. 5, and in Figs B7 to B12 at all latitudes. There is clearly significant correlation between $\hat{\Omega}_1$ and $\hat{\Omega}_2$, with little correlation between the remaining parameters.

The dependence of the fitting results on the RLS trade-off parameters (μ_r, μ_θ) is shown in Figs 6 and 7, for fits at the equator. The corresponding results at all latitudes are provided in Figs B13 - B24. In principle, results such as these can be used to select the optimal trade-off parameters, as far as characterizing the tachocline is concerned.

Qualitatively the results look reasonable, with a generally increasing departure from the true values, particularly for the width w, with increasing trade-off parameters and hence increasing kernel widths, while the standard deviations decrease. The balance between resolution and error is clearly something that needs to be analysed more carefully.

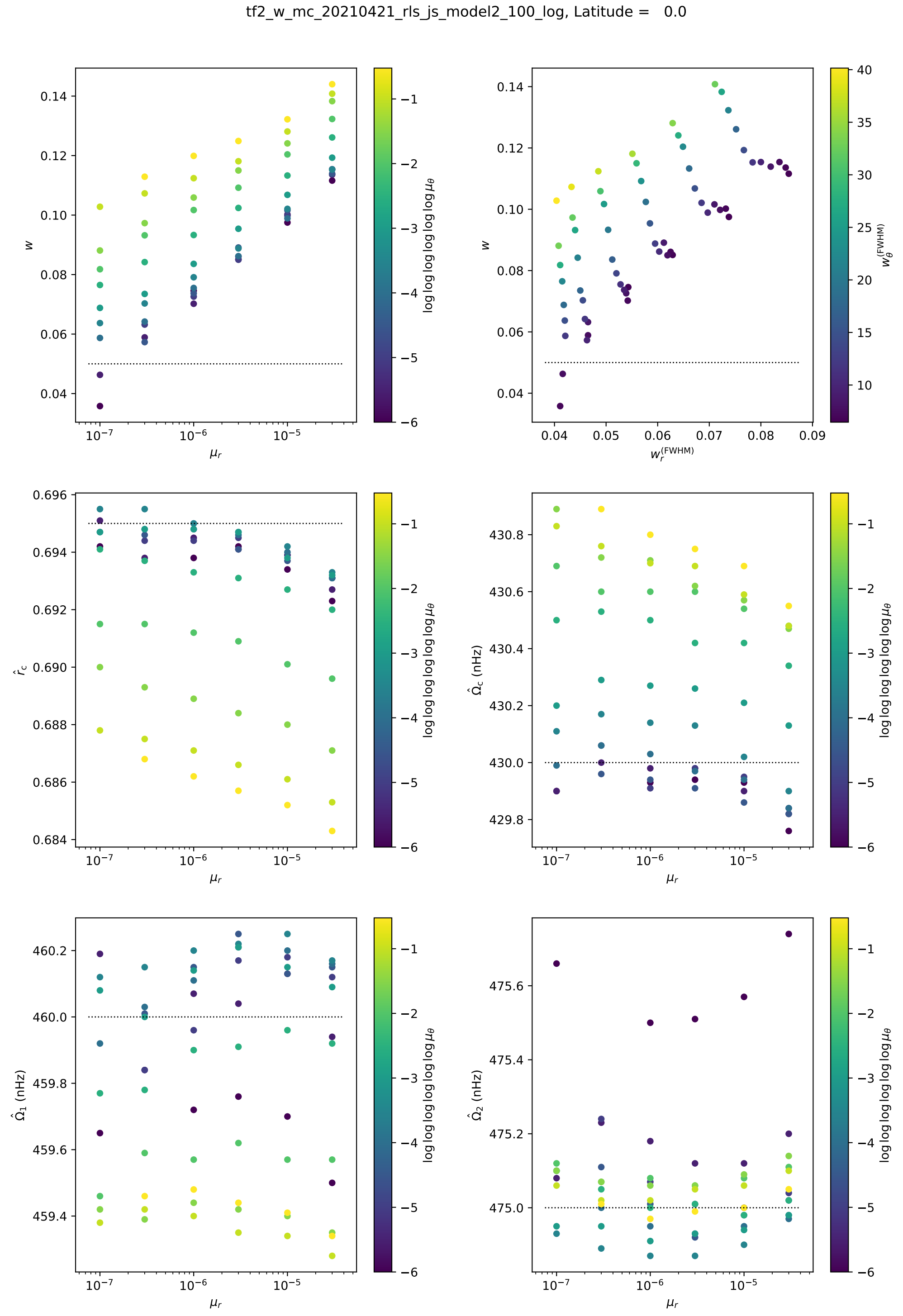


Figure 6: Average results at the equator of fits using 100 MC samples, as functions of the trade-off parameter μ_r and colour-coded with the trade-off parameter μ_θ, for RLS inversions of JS a coefficients. The upper right-hand panel shows results for the width w, against the full width at half maximum $w_r^{(\mathrm{FWHM})}$ in $\hat{r}$ and colour-coded with the full width at half maximum $w_\theta^{(\mathrm{FWHM})}$ in latitude. The horizontal dotted lines show the true values of the parameters of the model (*cf.* Table 1).

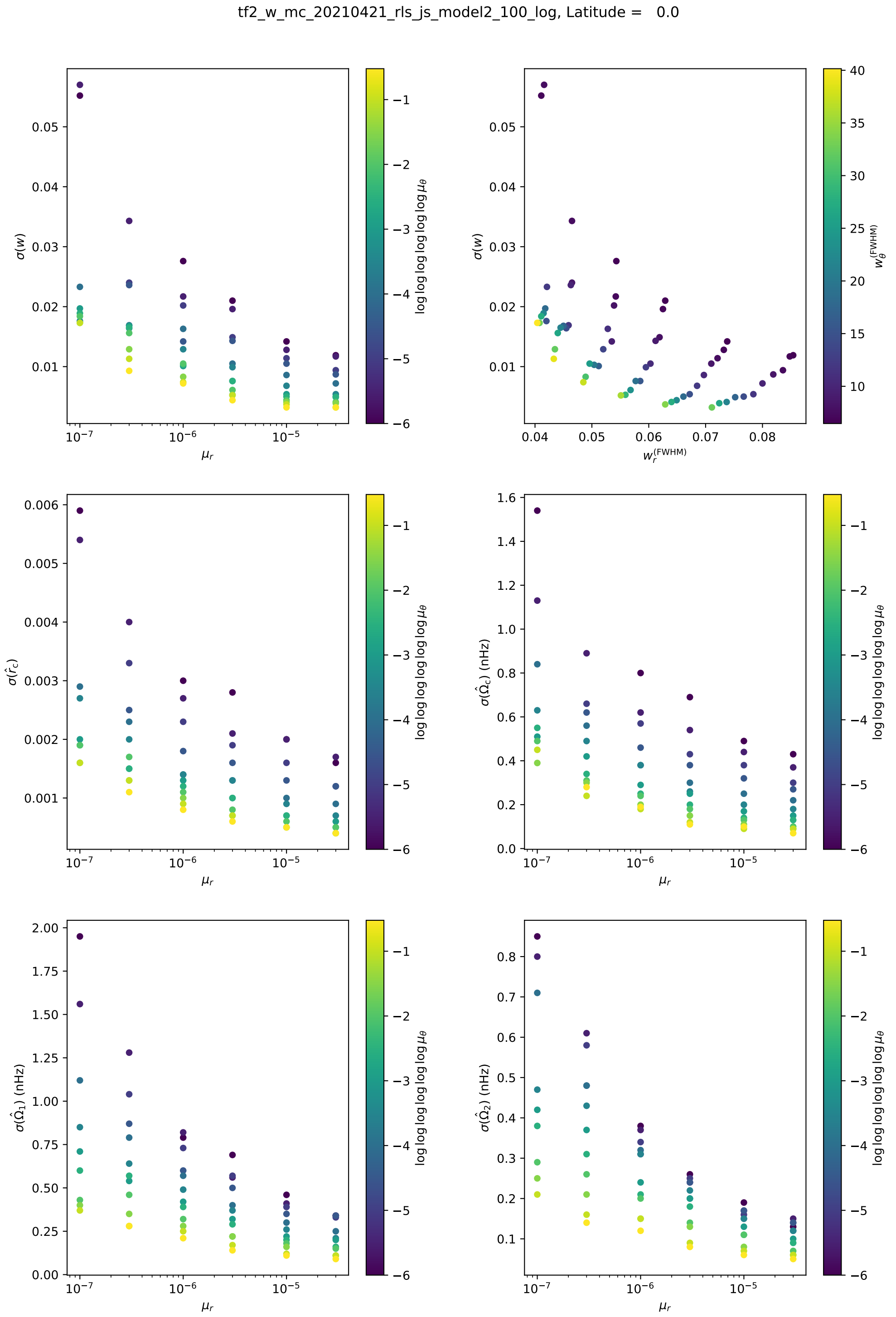


Figure 7: Standard deviations for the fits using 100 MC samples shown in Fig. 6; see caption to that figure for details.

A Properties of Model2 rotation rate

This was set up by Sasha Kosovichev as a basis for well-defined artificial data including some of the properties thought relevant for the Sun, including a tachocline. This makes it useful for testing the fitting of the tachocline properties.

A.1 Code and parameters for calculating the Model 2 rotation rate

Below I include the `Python` code used to compute the parametrized rotation rate, converted and slightly modified from Sasha's `FORTRAN` code. Here I included the following parameters to be varied, also listed in Table 1:

Parameter	variable	true value
$\hat{r}_{\rm c}$	`rc`	0.695
w	`w`	0.05
$\hat{\Omega}_{\rm c}$	`omc`	430 nHz
$\hat{\Omega}_1$	`omin1`	460 nHz
$\hat{\Omega}_2$	`omin2`	475 nHz

Figures A1 - A6 show the effects of variations of these parameters from their original values.

`Python` code:

```
def tachfit2(ri,rc,w,omc,omin1,omin2):
    def rot1(r,theta):
        nr=r.shape[0]
        nt=theta.shape[0]
        rr=np.zeros((nt,nr))
        tt=np.zeros((nt,nr))
        for i in range(nt):
            rr[i,:]=r
        for j in range(nr):
            tt[:,j]=theta
        ct=np.cos(tt)
        ct2=ct**2
        om=rot0(rr,tt)
        dr1=0.05
        r1=0.252-0.04*(0.5-ct2)
        om1=rot0(r1-dr1/2,tt)
        om2=rot0(r1+dr1/2,tt)
#        dr2=0.05
        dr2=w
#        r2=0.695+0.03*(0.5-ct2**2)
        r2=rc+0.03*(0.5-ct2**2)
        om3=rot0(r2-dr2/2,tt)
        om4=rot0(r2+dr2/2,tt)
        for i in range(nt):
            for j in range(nr):
                if r1[i,j]-dr1/2 <= rr[i,j] <= r1[i,j]+dr1/2:
                    om[i,j]=om1[i,j]+(om2[i,j]-om1[i,j])/2.* \
                            (1+np.sin(np.pi*(rr[i,j]-r1[i,j])/dr1))
                if r2[i,j]-dr2/2 <= rr[i,j] <= r2[i,j]+dr2/2:
                    om[i,j]=om3[i,j]+(om4[i,j]-om3[i,j])/2.* \
                            (1+np.sin(np.pi*(rr[i,j]-r2[i,j])/dr2))
        [dr3, r3, r4] = [0.02, 0.99, 1]
        rr3=r3+0*rr
        rr4=r4+0*rr
        ct1=np.cos(8.*tt)
        om5=rot0(rr3,tt)
        om6=rot0(rr4,tt)*0.98
        for i in range(nt):
            for j in range(nr):
                if(r3-dr3/2 <= rr[i,j] <= r4):
                    om[i,j]=om5[i,j]+(om6[i,j]-om5[i,j])/2.* \
                            (1+np.sin(np.pi*(rr[i,j]-r3)/dr3))
                if(r3 <= rr[i,j] <= r4):
                    om[i,j]+=1.5*(0.5-ct1[i,j]**2)
        x=np.abs(r*np.sin(tt))
        y=np.abs(r*np.cos(tt))
        [x0, y0, dx, dy, a] = [0.2, 0.9, 0.1, 0.05, 20]
        om+=a*np.exp(-(x-x0)**2/dx**2)*np.exp(-(y-y0)**2/dy**2)
        return om
    def rot0(rr,tt):
        (nt,nr)=rr.shape
        ct=np.cos(tt)
        ct2=ct**2
        ct1=np.cos(5.*tt)
        om1=420.+20*(0.5-ct2)
#        [om2,  om3,  om4,  om5,  om6] = [450., 430., 445., 460., 475.]
        [om2,  om3,  om4,  om5,  om6] = [450, omc, 445., omin1, omin2]
        r1=0.252-0.04*(0.5-ct2)
        om=om4+0*rr
#        r2=0.695+0.03*(0.5-ct2**2)
        r2=rc+0.03*(0.5-ct2**2)
#        r20=0.695+0.07*(0.5-ct1**2)*(rr-r1)/(r2-r1)
        r20=rc+0.07*(0.5-ct1**2)*(rr-r1)/(r2-r1)
        r3=0.99
        a=(om6-om5)/(np.sqrt(r3)-np.sqrt(r2))
        c=om5-a*np.sqrt(r2)
        z=rr*np.sin(tt)+1.e-6
        r4=1
        for i in range(nt):
            for j in range(nr):
                if rr[i,j] <= r1[i,j]:
                    om[i,j]=om1[i,j]+(om2-om1[i,j])/r1[i,j]*rr[i,j]
                if r1[i,j] <= rr[i,j] <= r2[i,j]:
                    om[i,j]=om3+(om4-om3)/(r20[i,j]-r1[i,j])**4* \
                            (rr[i,j]-r1[i,j])**4
                if r2[i,j] <= rr[i,j] <= r4:
                    om[i,j]=a[i,j]*np.sqrt(z[i,j])+c[i,j]
        return om
    theta=np.array([(90.-latitude)*np.pi/180])
    om=rot1(ri,theta)
    return om[0,:]
```

A.2 Changes for Model 2 at all latitudes

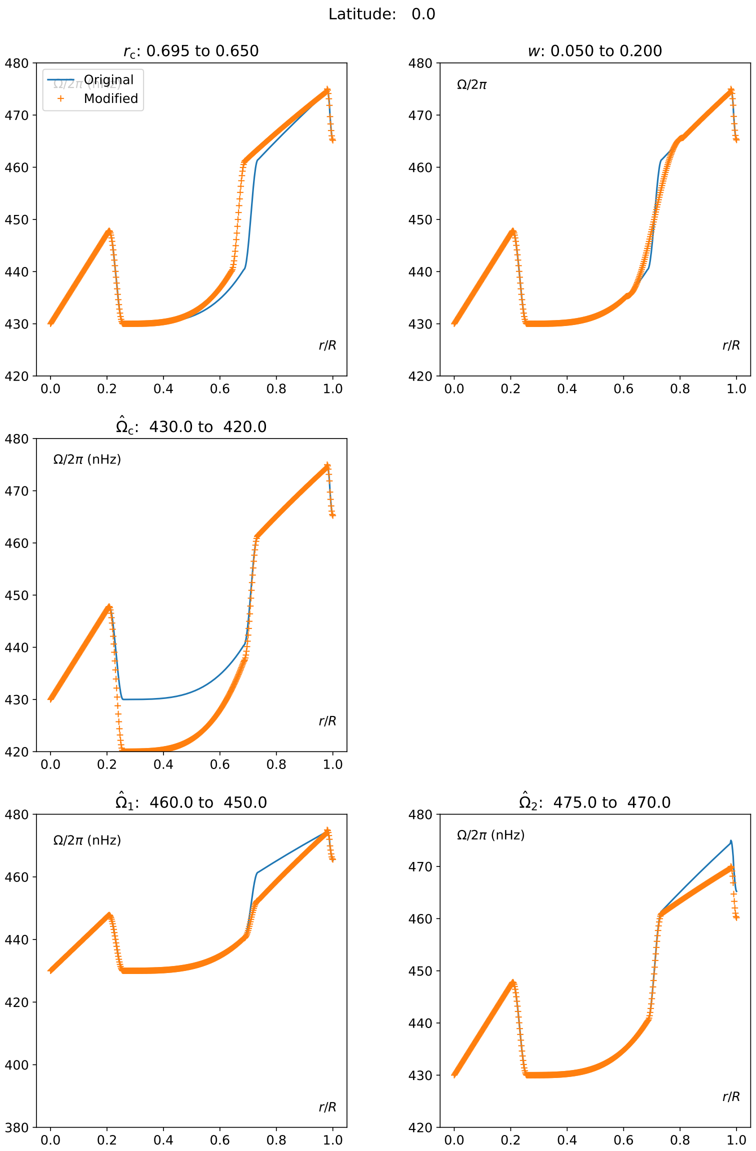


Figure A1: Effects in the equatorial rotation rate of changes to the five parameters $\{p_i\}$ chosen to represent the Model 2 rotation rate (see Fig. 2).

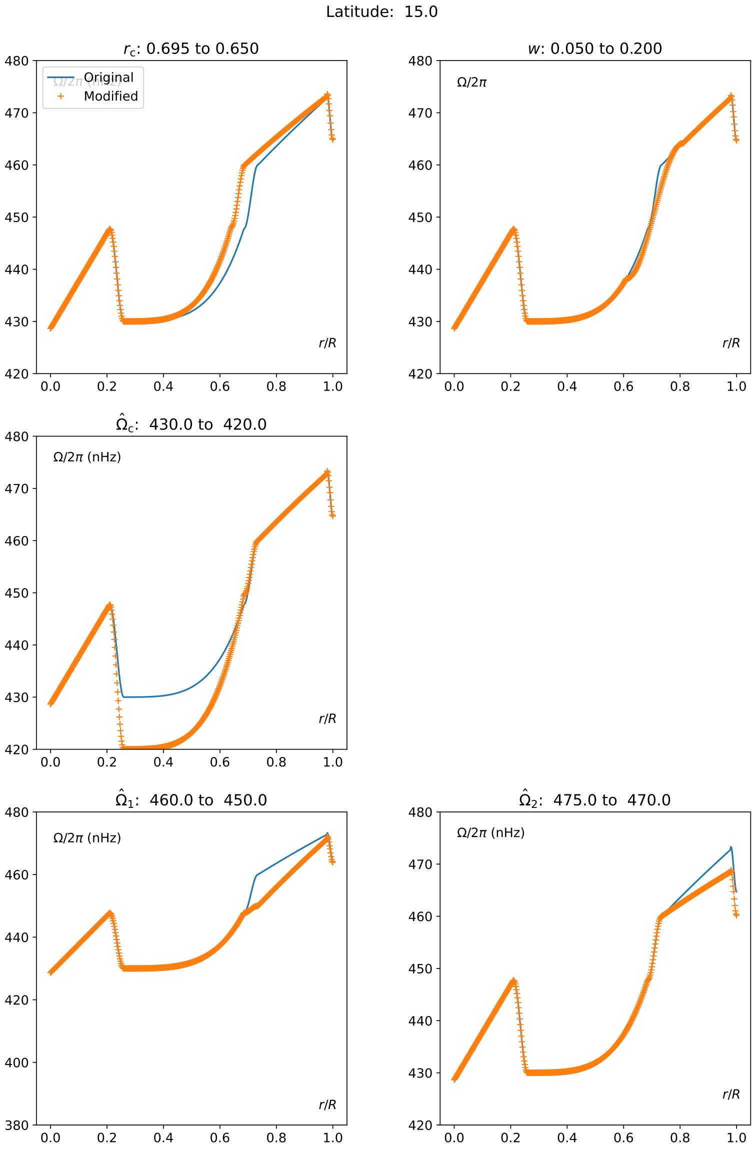


Figure A2: As Fig. A1, but at latitude 15°.

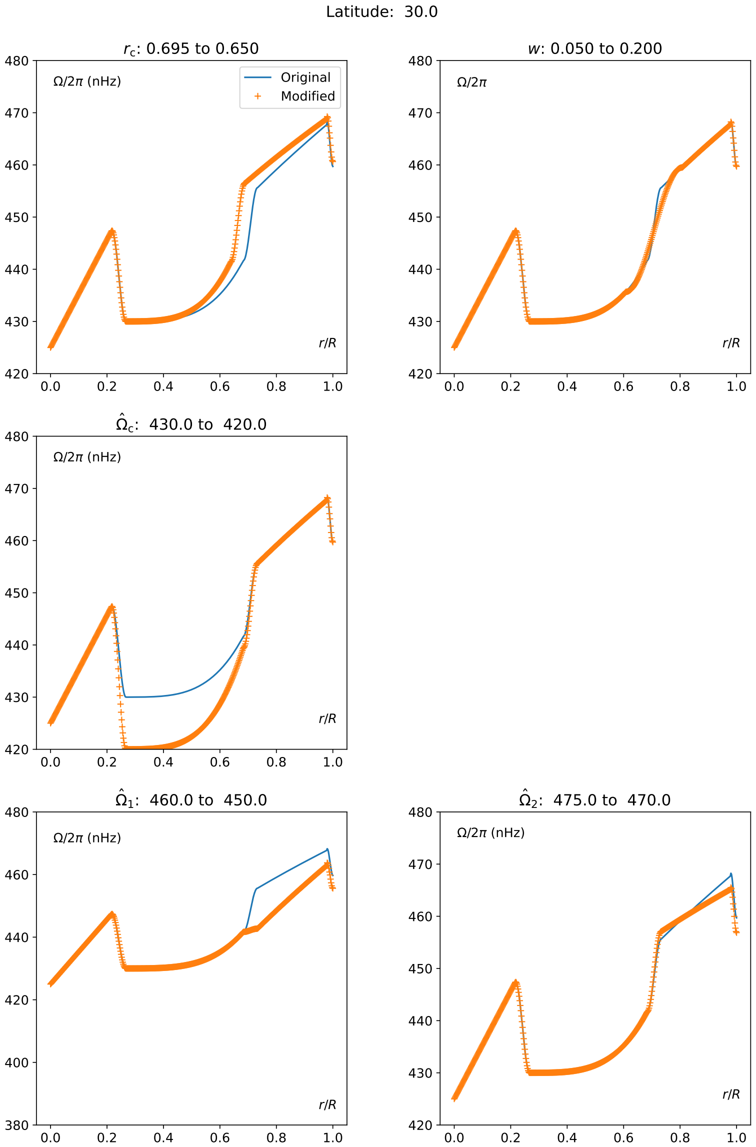


Figure A3: As Fig. A1, but at latitude 30°.

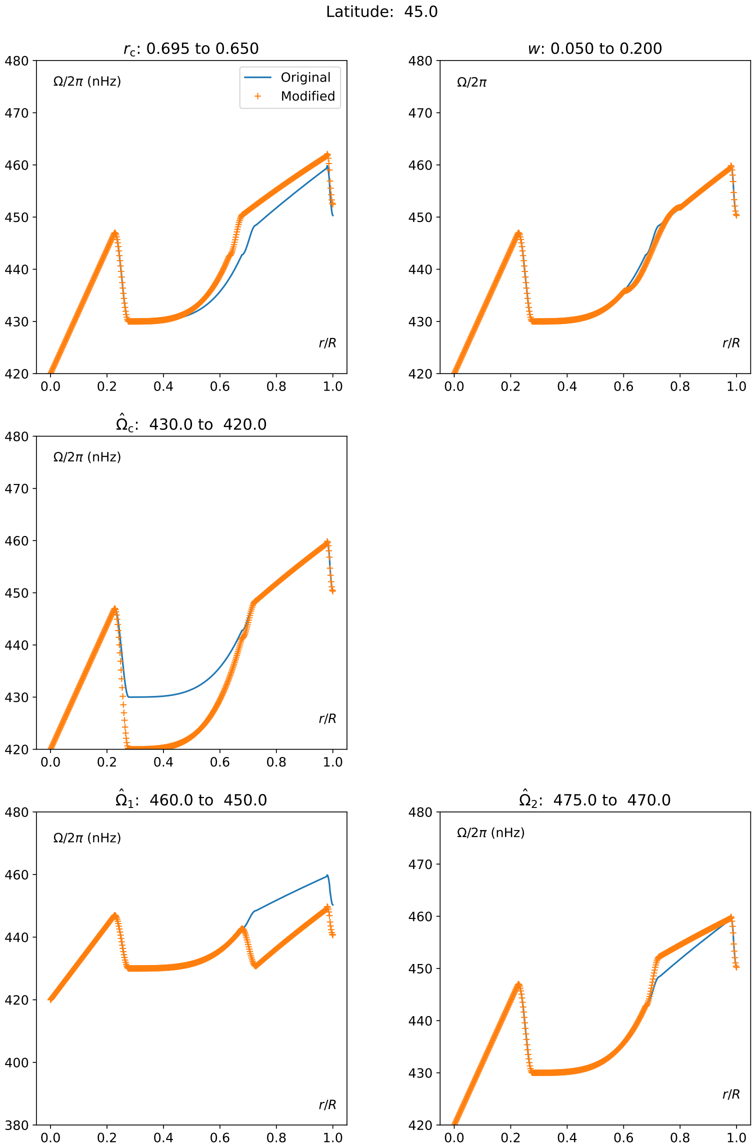


Figure A4: As Fig. A1, but at latitude 45°.

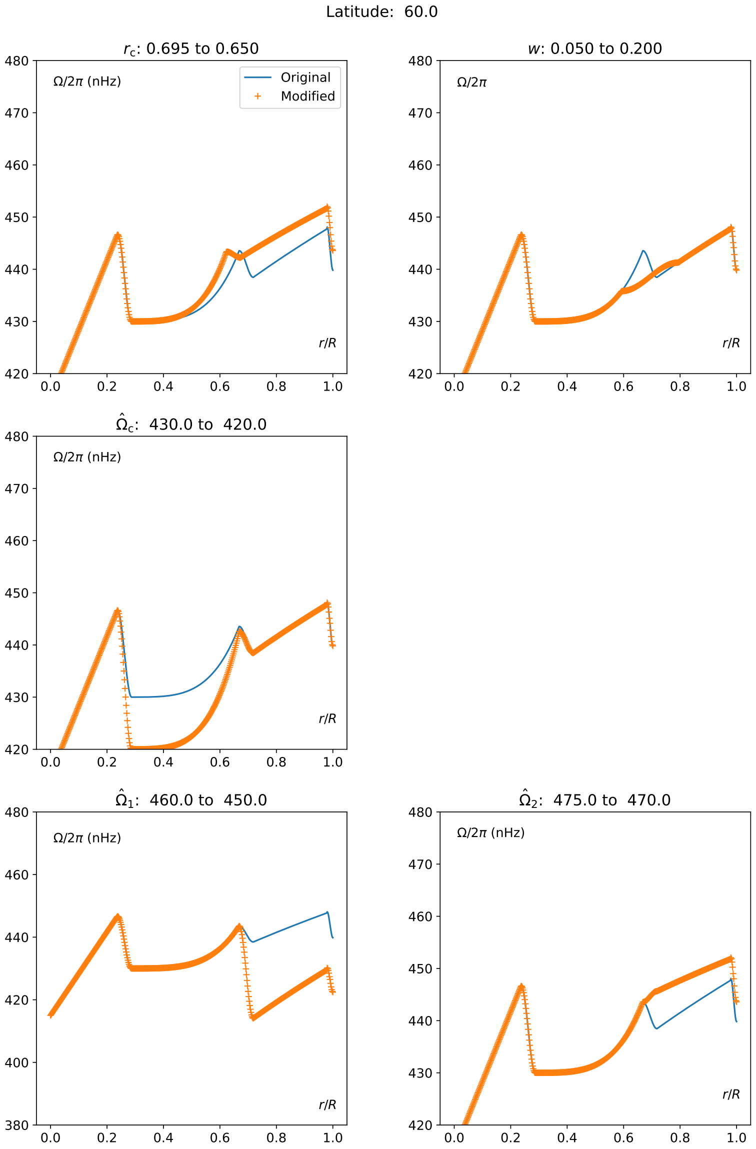
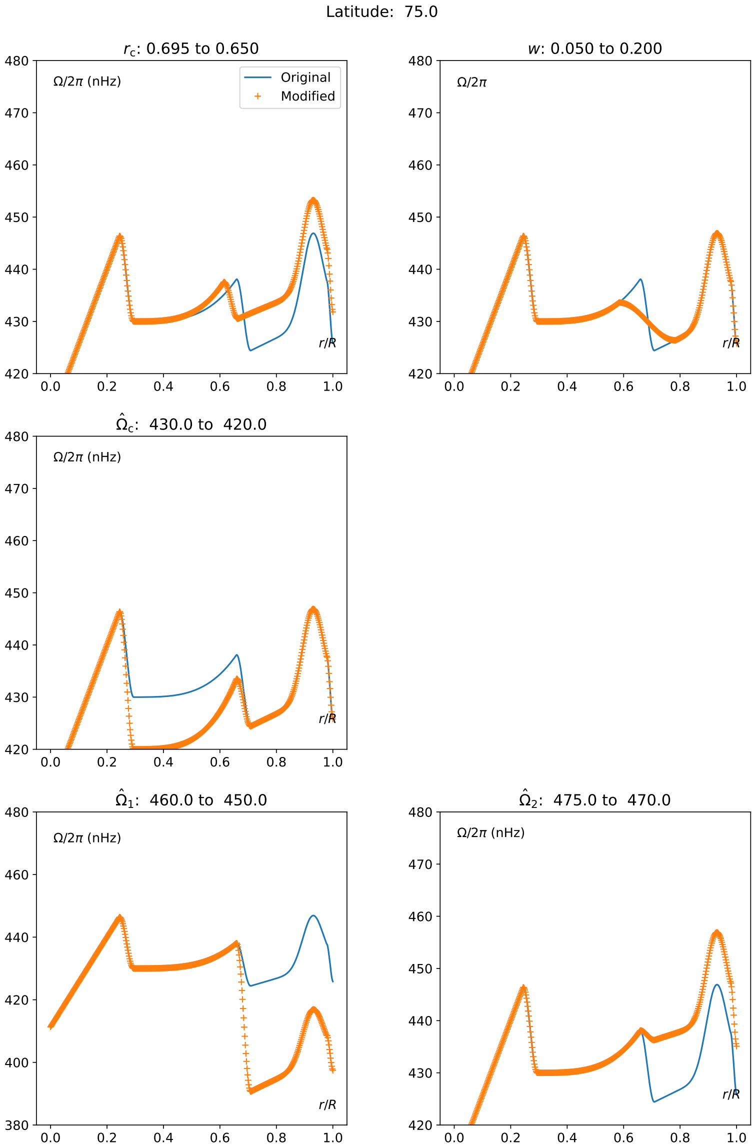


Figure A5: As Fig. A1, but at latitude 60°.



Latitude: 75.0

r_c: 0.695 to 0.650

w: 0.050 to 0.200

$\hat{\Omega}_c$: 430.0 to 420.0

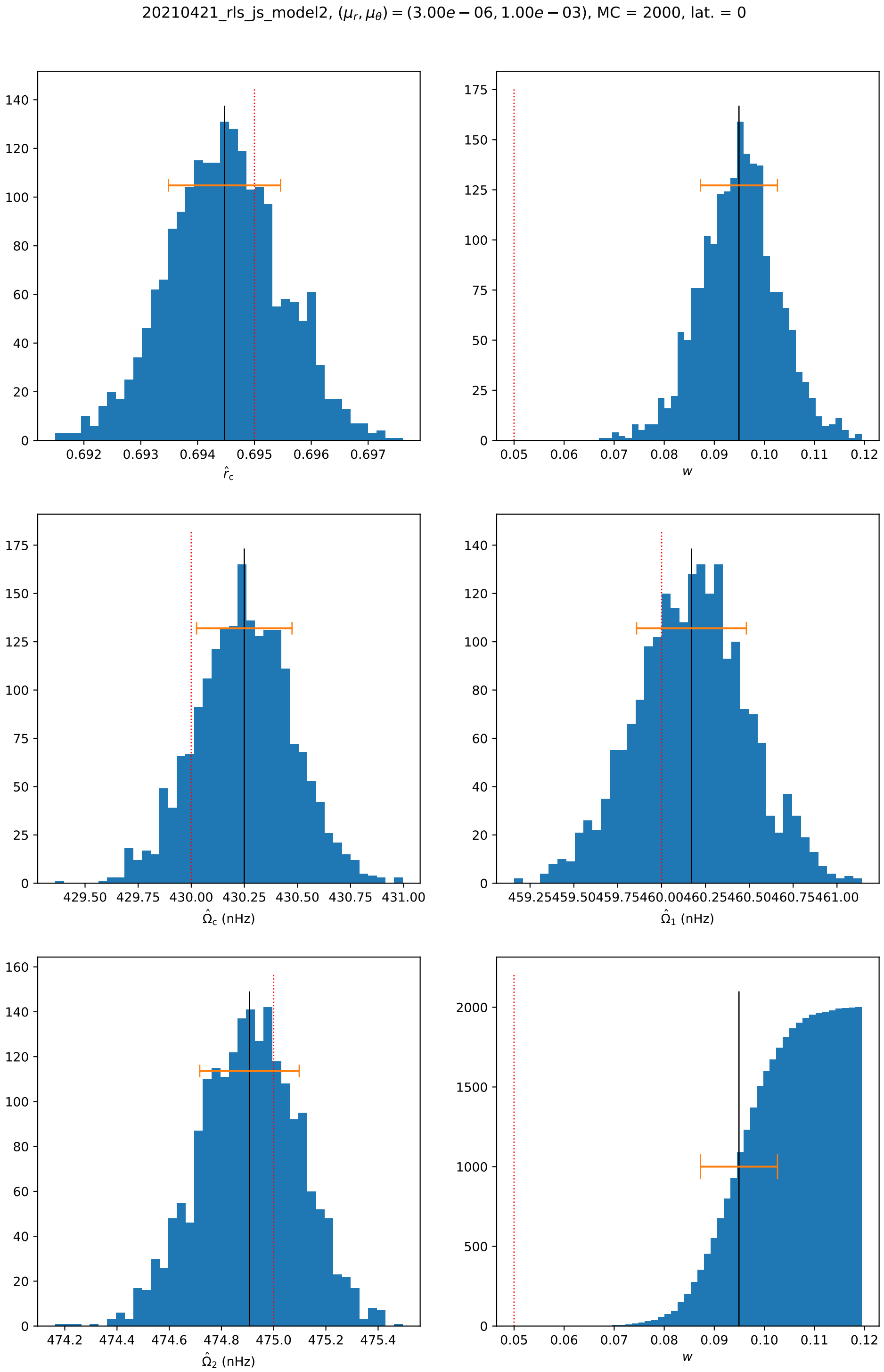
$\hat{\Omega}_1$: 460.0 to 450.0

$\hat{\Omega}_2$: 475.0 to 470.0

Figure A6: As Fig. A1, but at latitude 75°.

B Statistics for the RLS inversions of JS a coefficients for Model 2

.



20210421_rls_js_model2, $(\mu_r, \mu_\theta) = (3.00e-06, 1.00e-03)$, MC = 2000, lat. = 0

Figure B1: Histogram for the fits illustrated in Fig. 3, at the equator. See caption to Fig. 4.

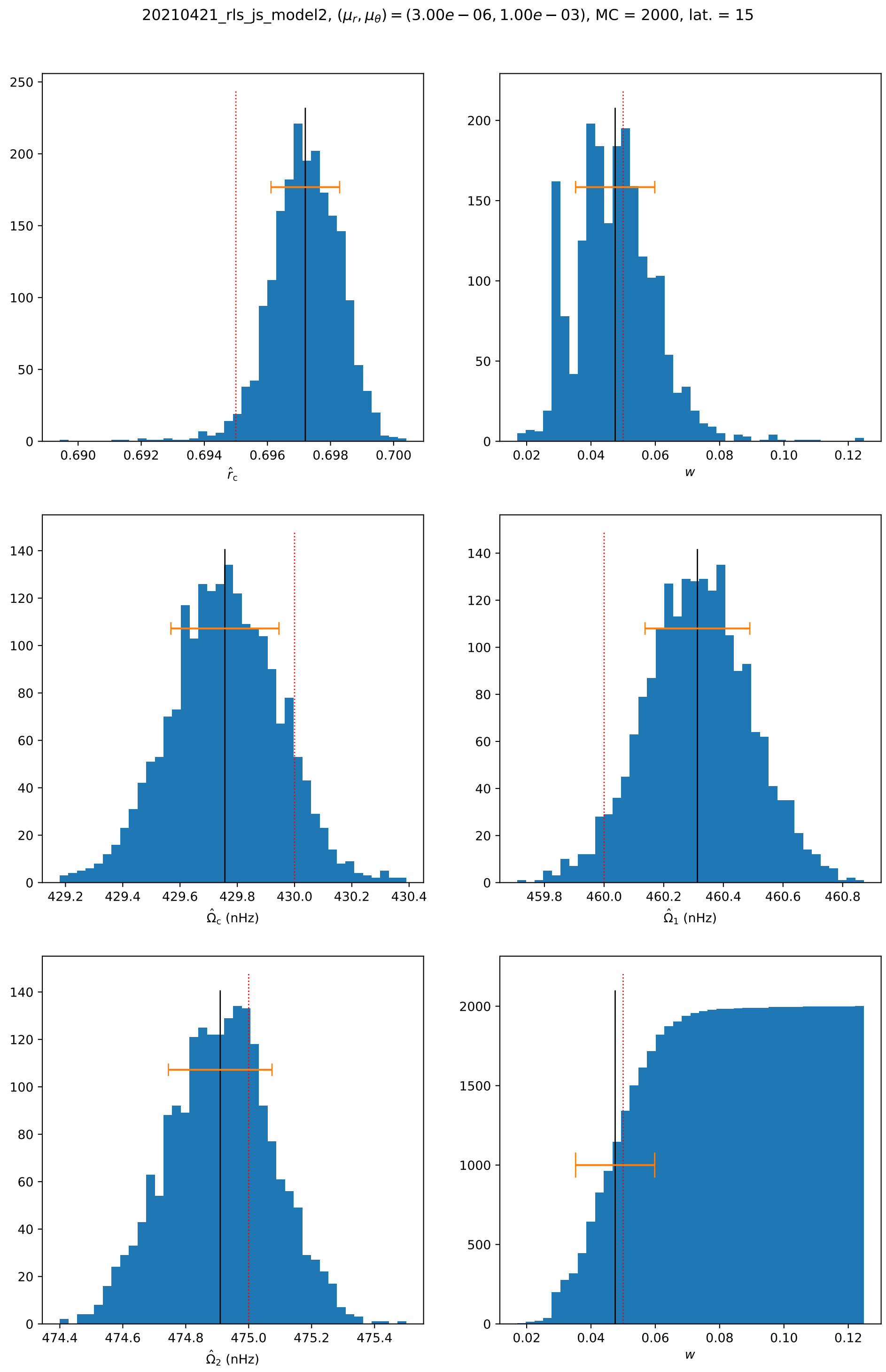


Figure B2: As Fig. B1, but at latitude 15°

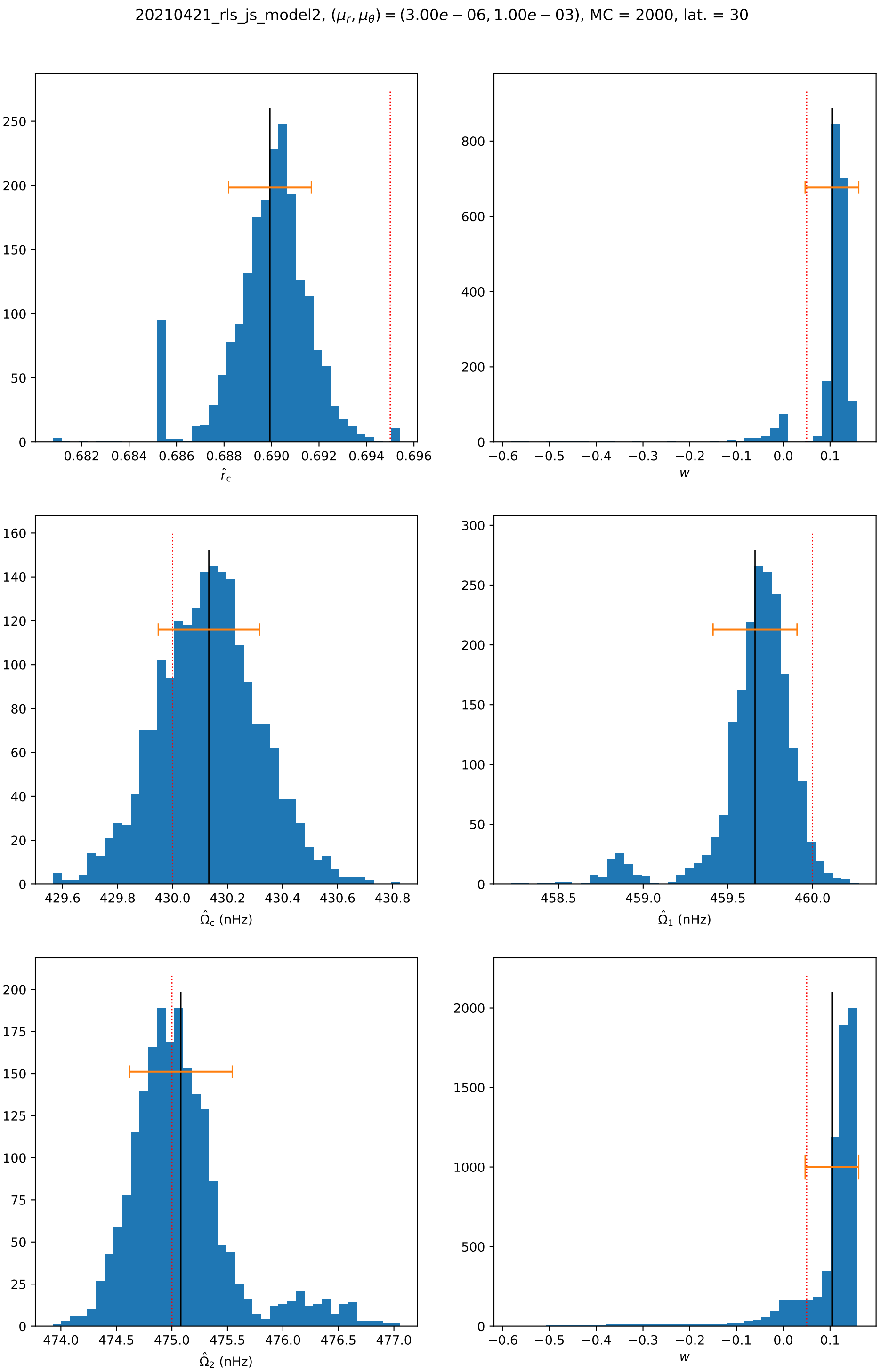


Figure B3: As Fig. B1, but at latitude 30°.

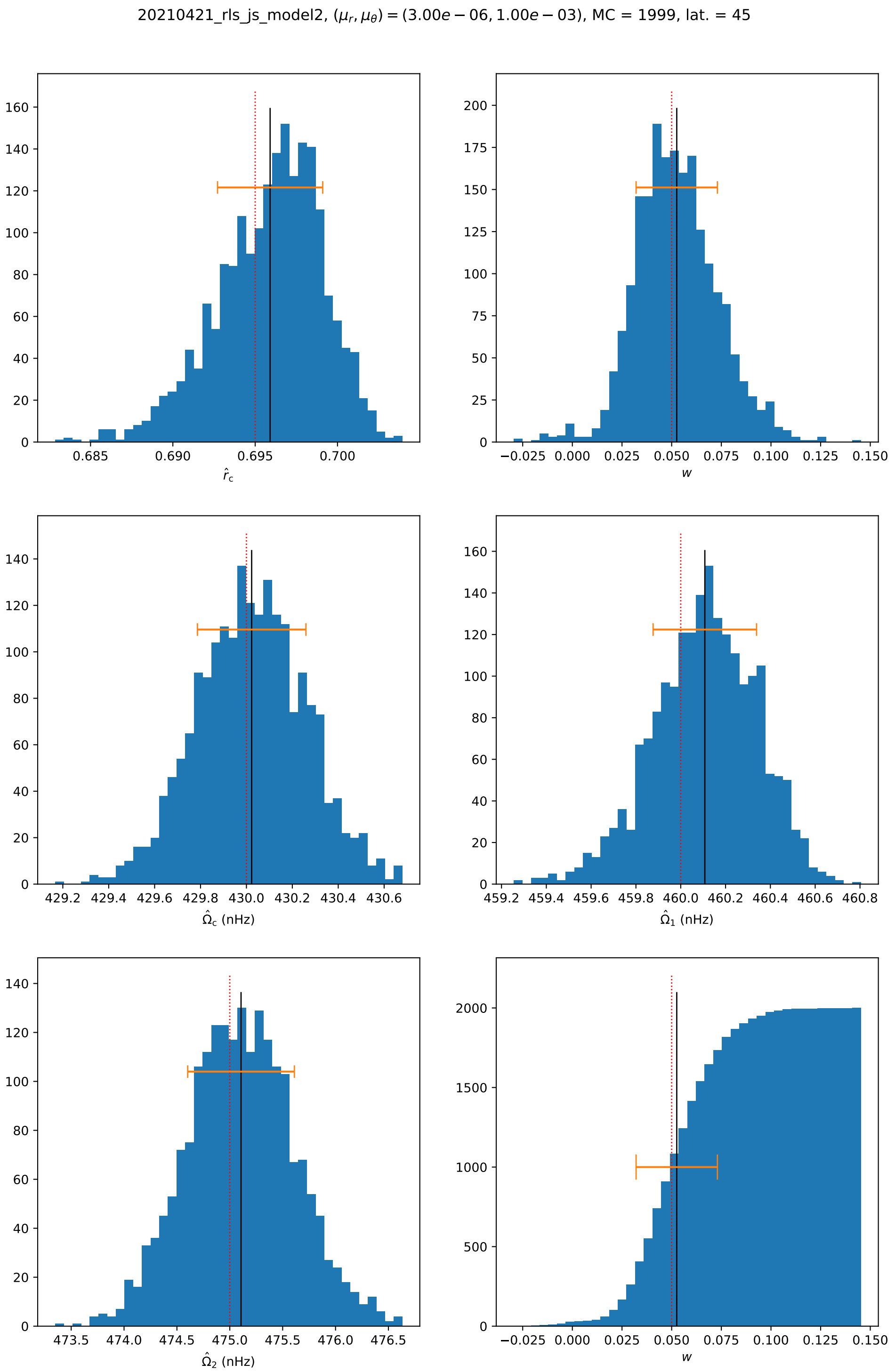


Figure B4: As Fig. B1, but at latitude 45°.

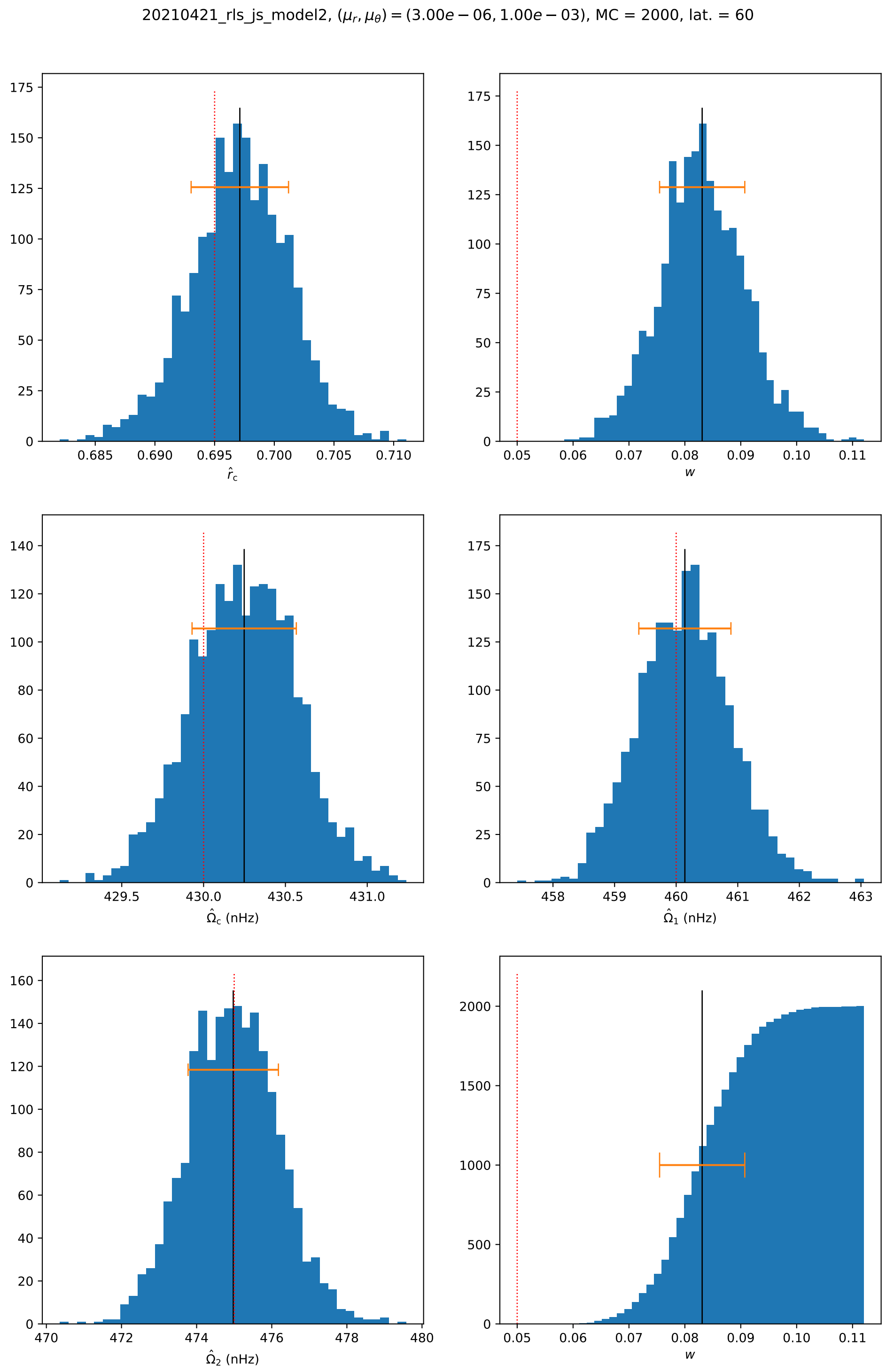


Figure B5: As Fig. B1, but at latitude 60°.

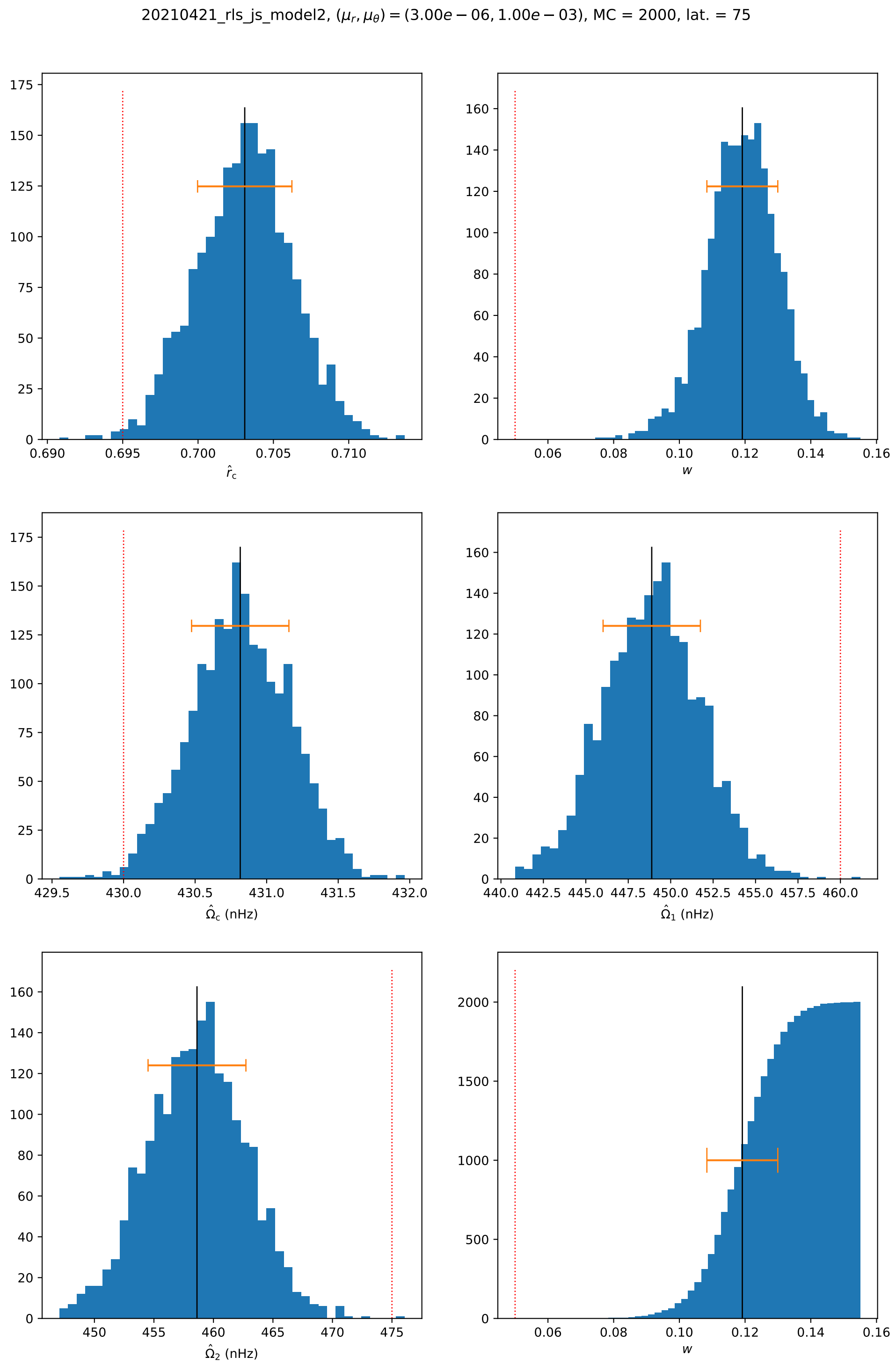


Figure B6: As Fig. B1, but at latitude 75°.

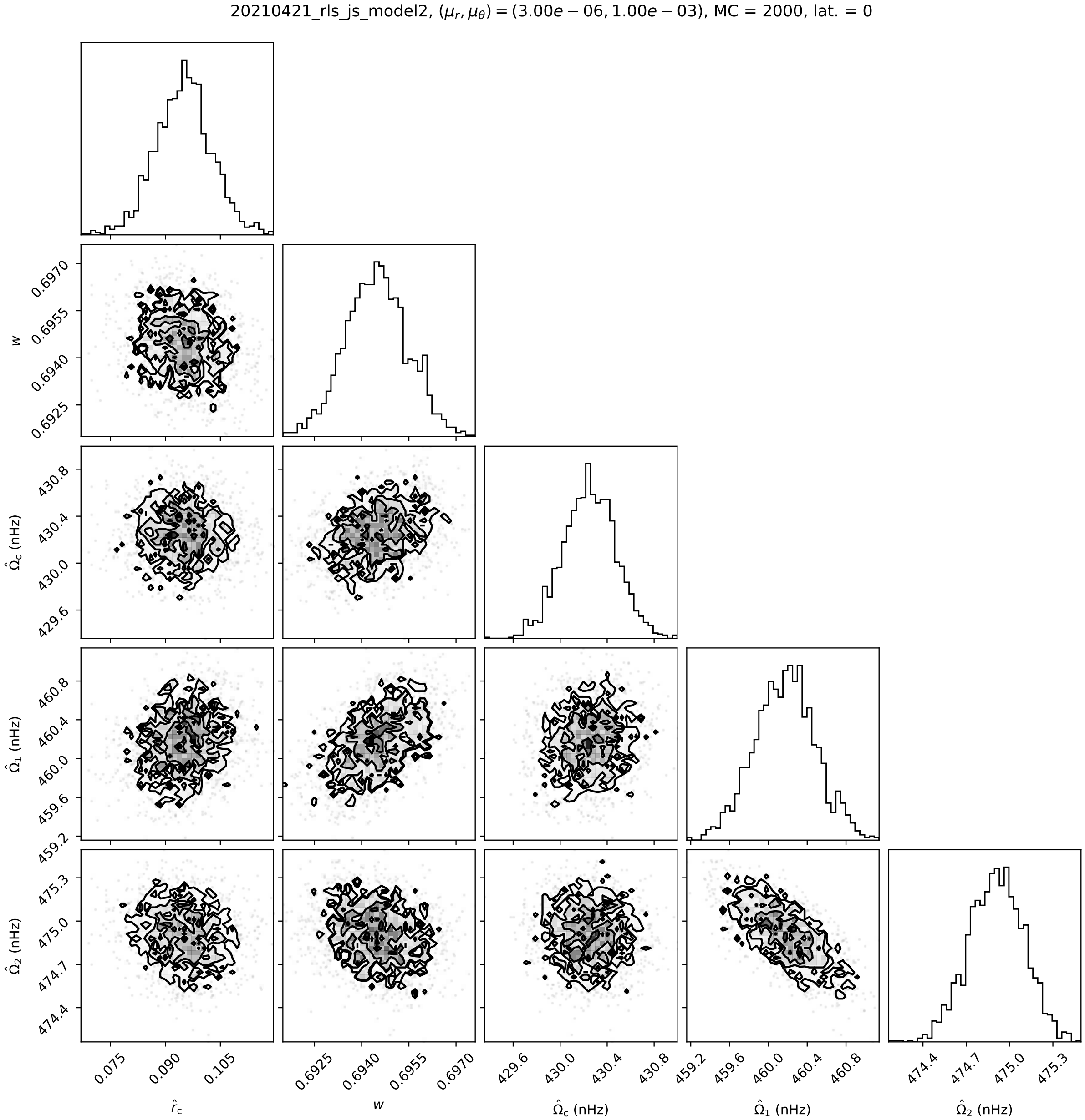


Figure B7: Corner plot for the fits illustrated in Fig. 3, at the equator.

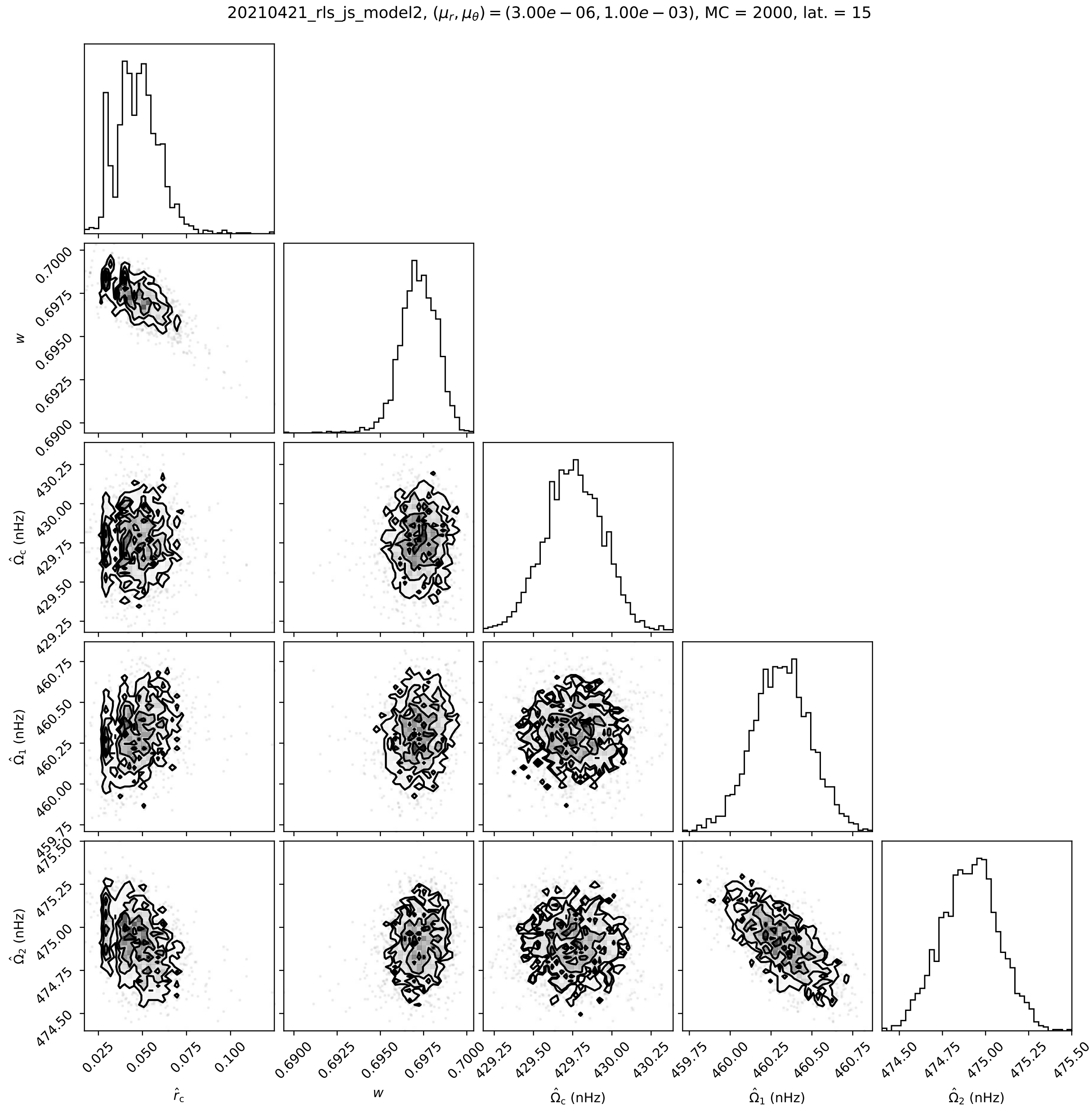


Figure B8: As Fig. B7, but at latitude 15°.

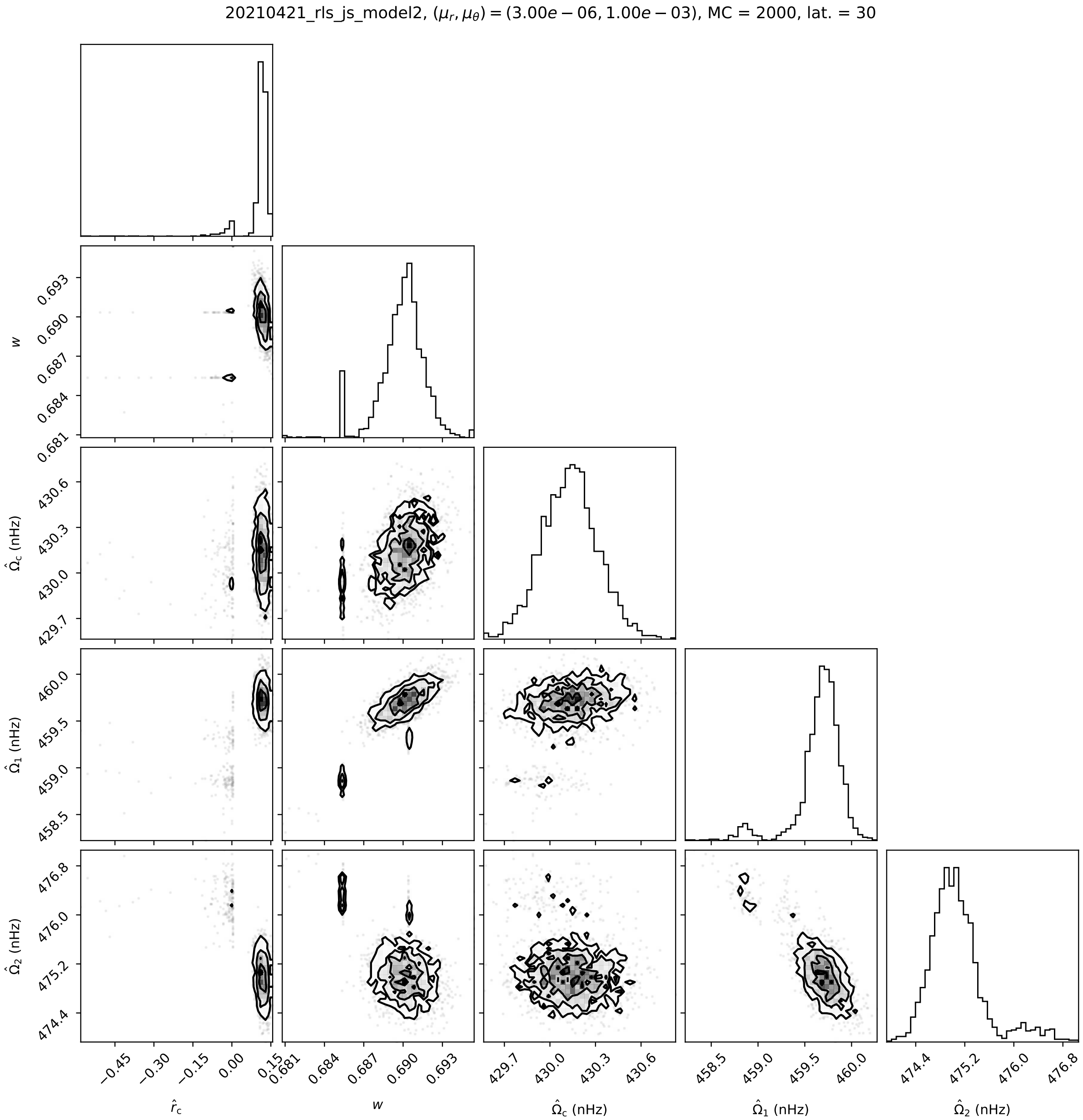


Figure B9: As Fig. B7, but at latitude 30°.

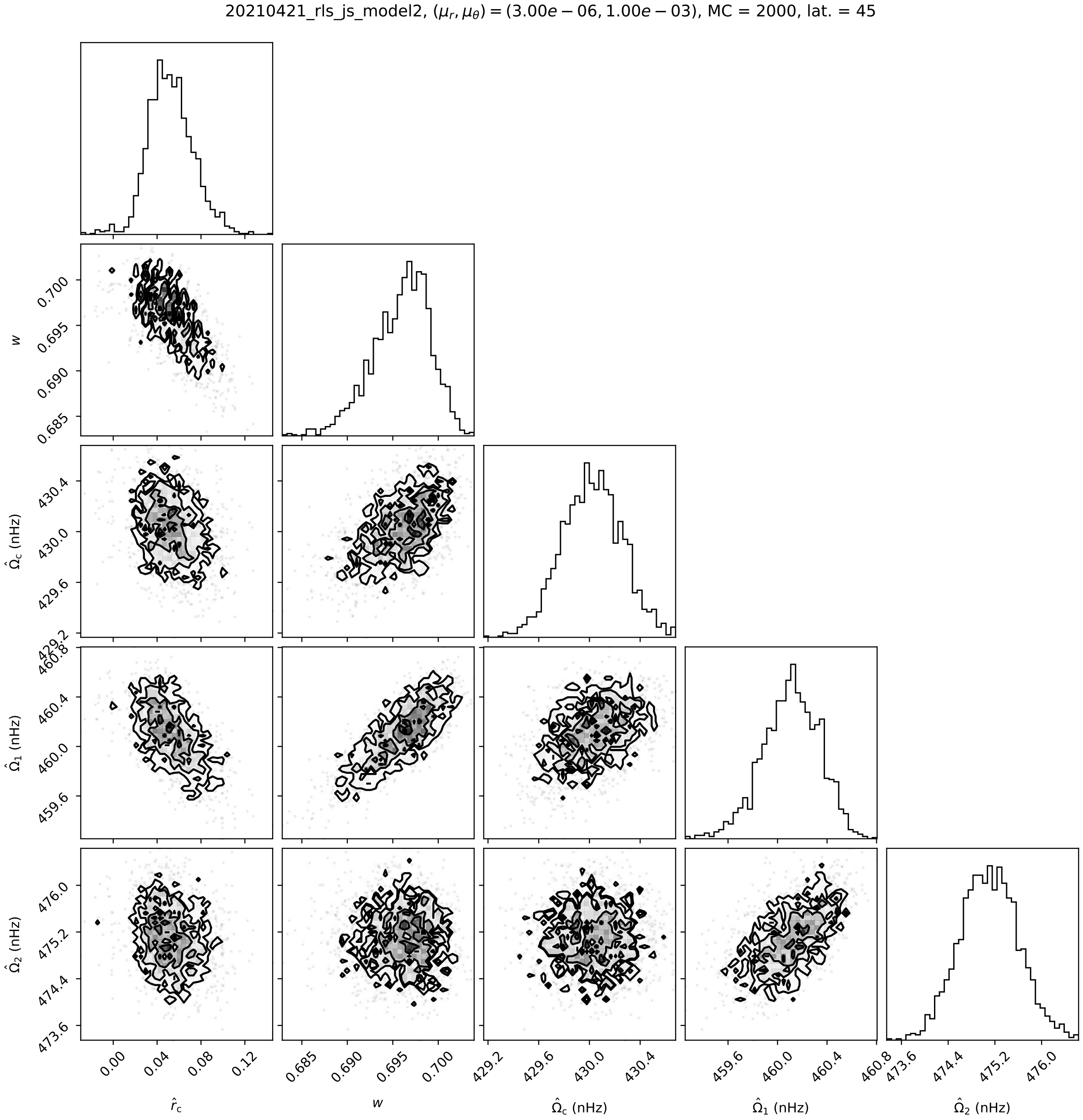


Figure B10: As Fig. B7, but at latitude 45°.

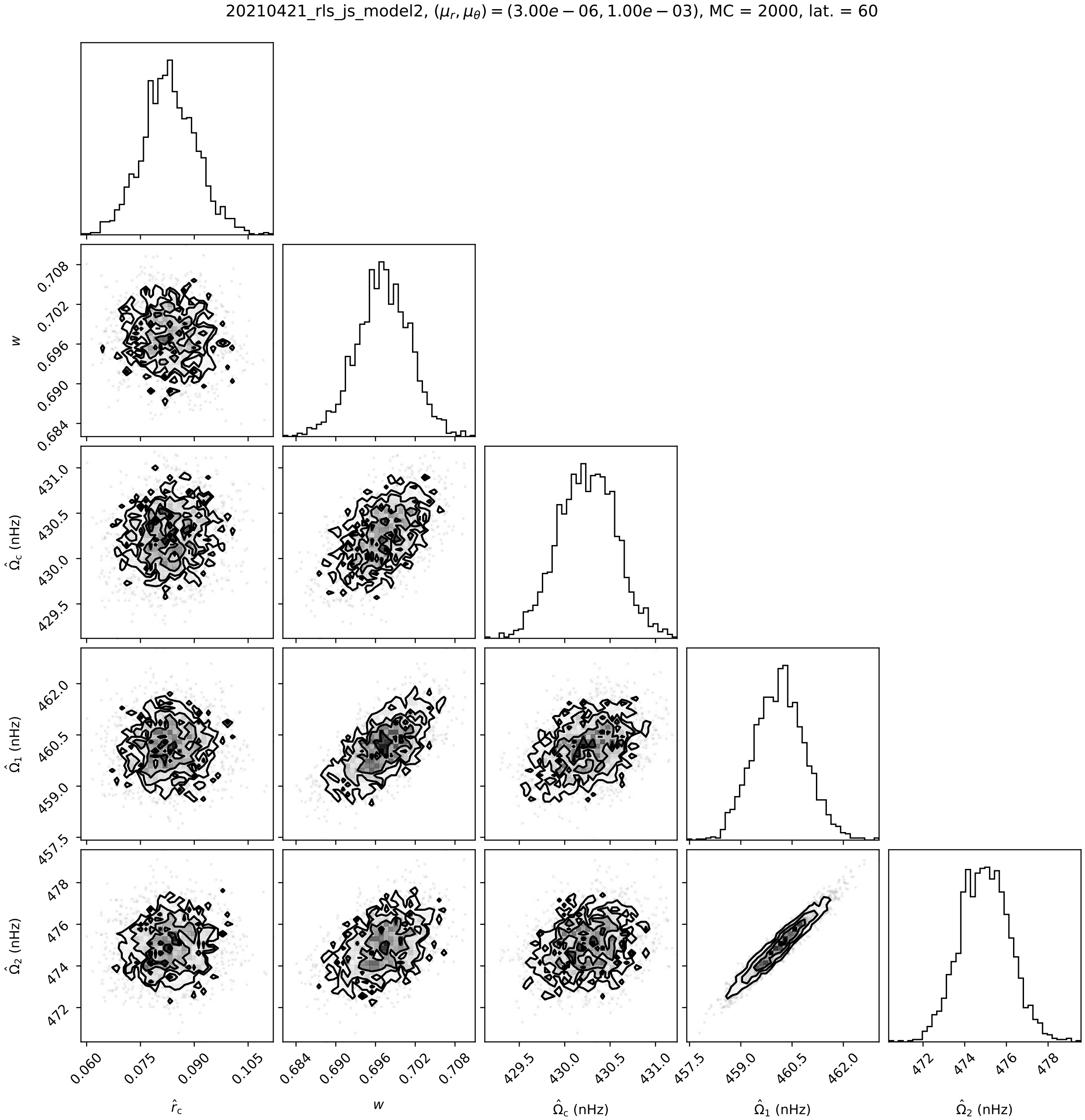


Figure B11: As Fig. B7, but at latitude 60°.

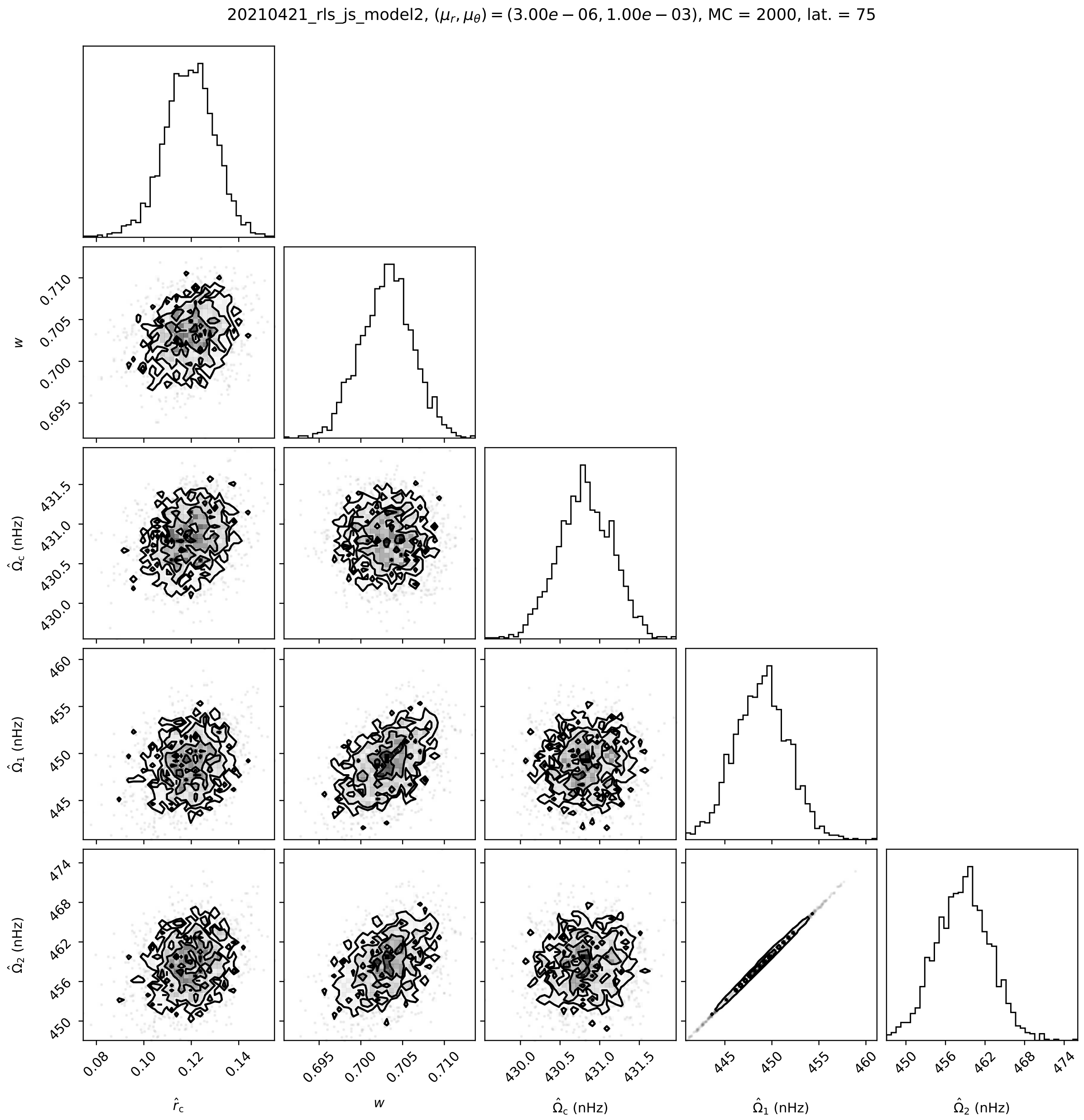


Figure B12: As Fig. B7, but at latitude 75°.

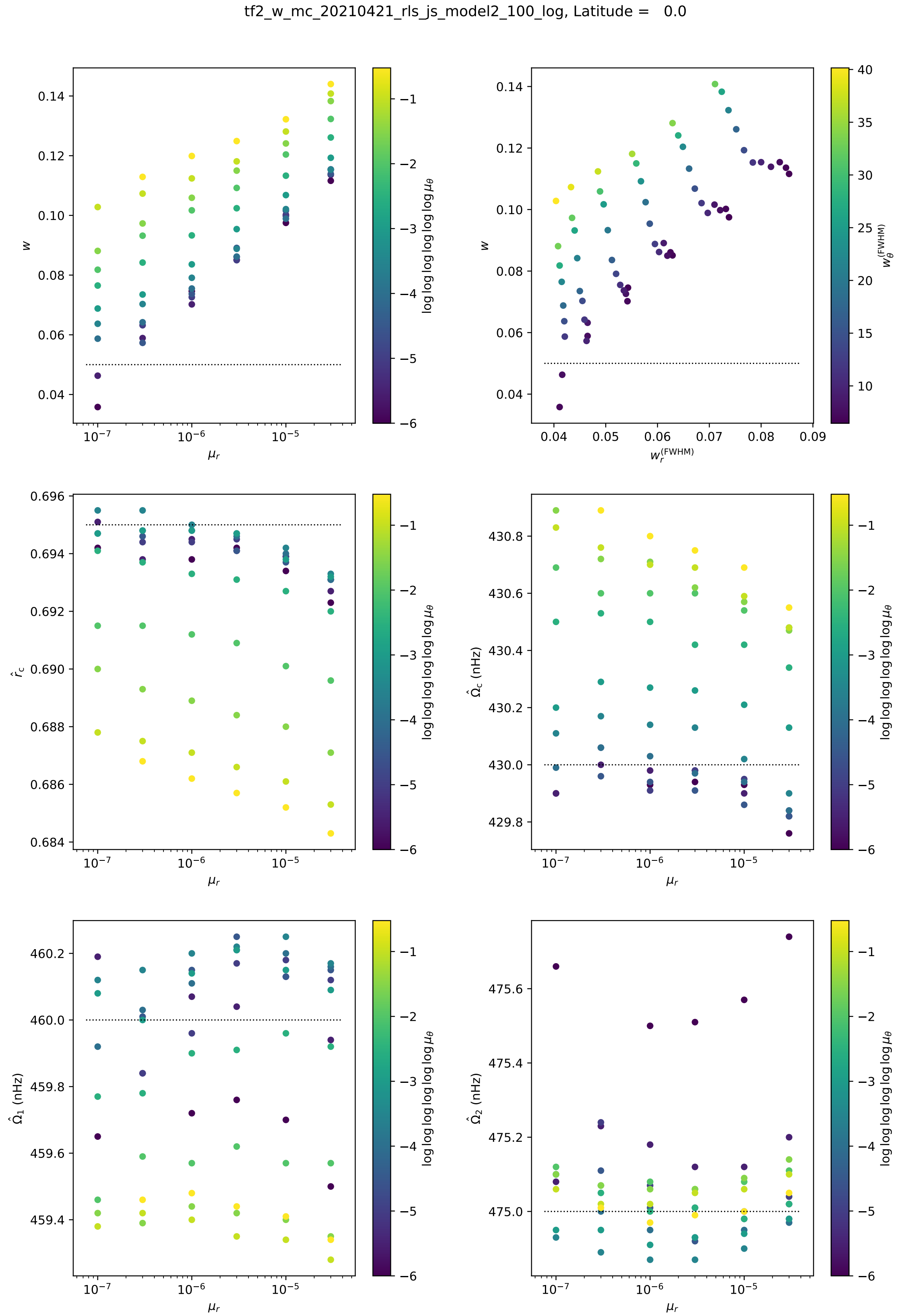

tf2_w_mc_20210421_rls_js_model2_100_log, Latitude = 0.0

Figure B13: Average results at the equator of fits using 100 MC samples, as functions of the trade-off parameter μ_r and colour-coded with the trade-off parameter μ_θ, for RLS inversions of JS a coefficients. See caption to Fig. 6 for details.

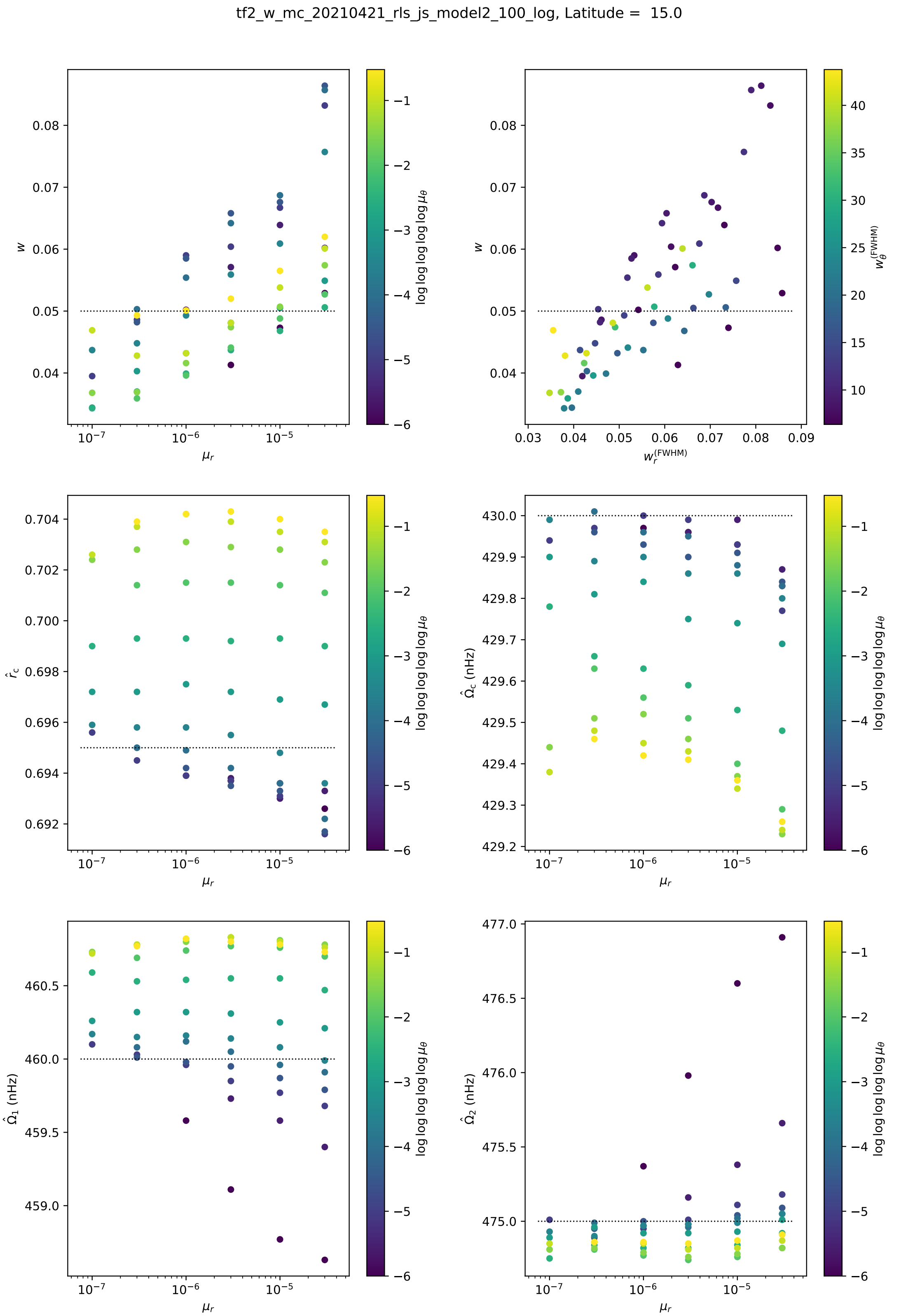


Figure B14: As Fig. B13, but at latitude 15°.

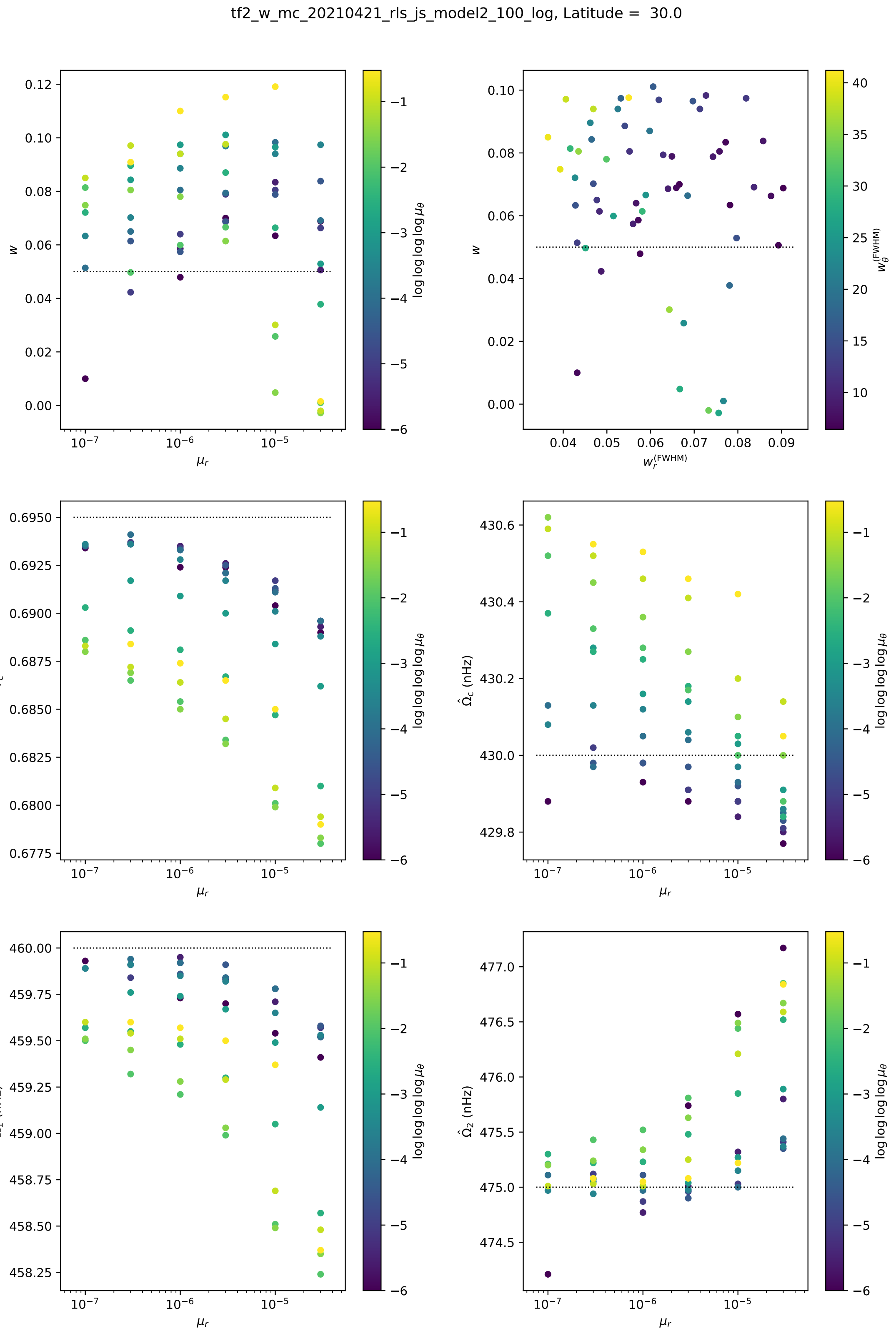


Figure B15: As Fig. B13, but at latitude 30°.

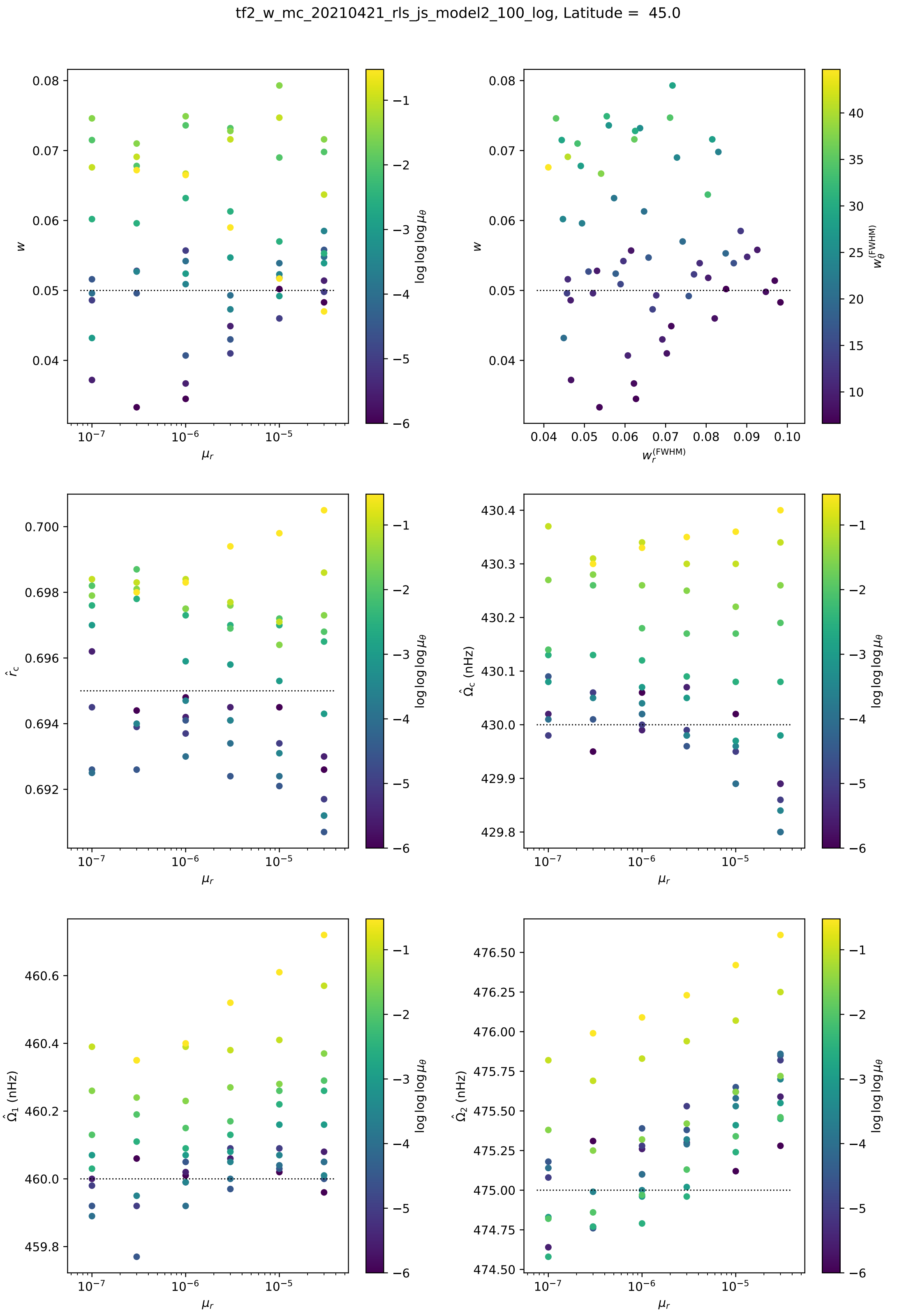


Figure B16: As Fig. B13, but at latitude 45°.

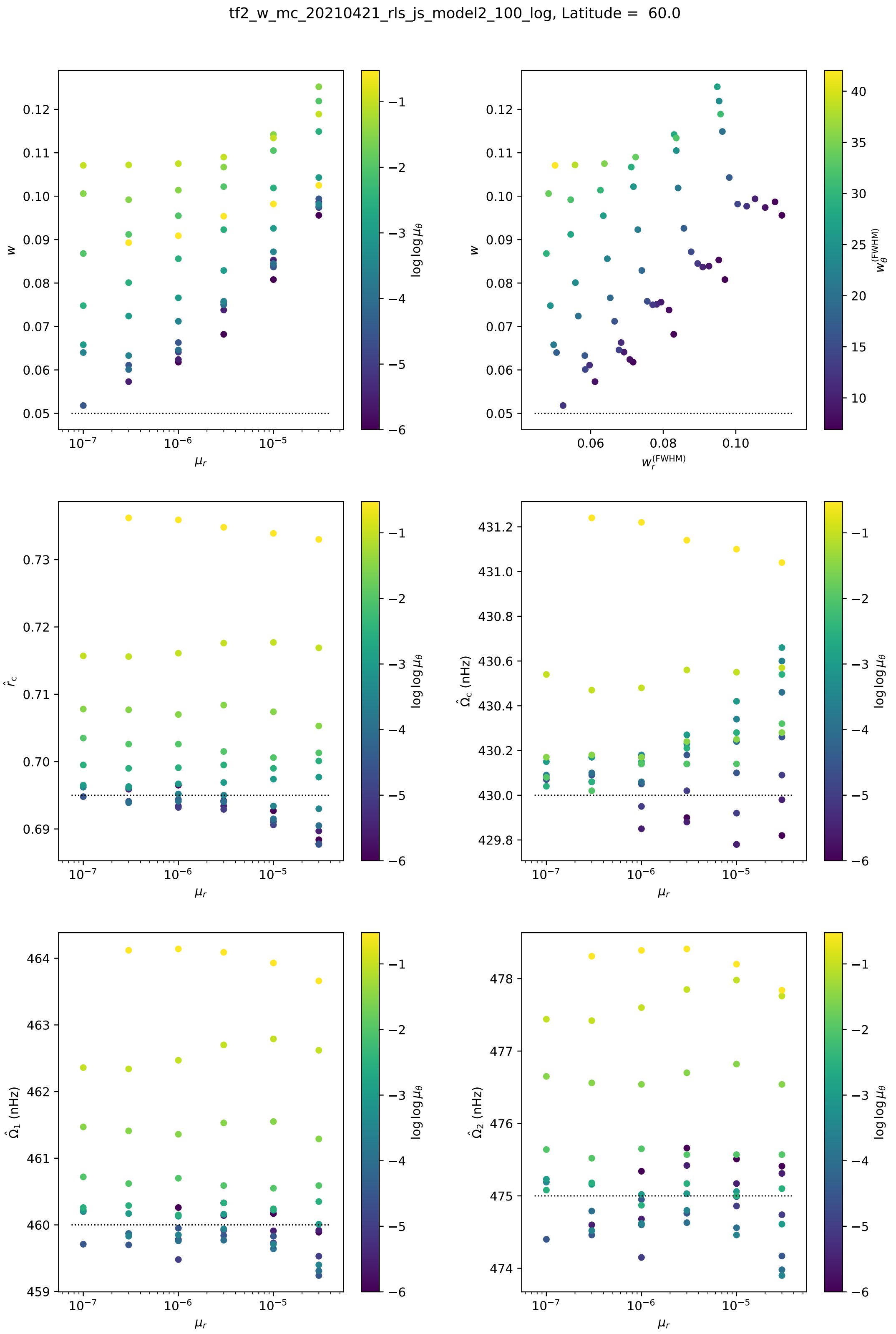


Figure B17: As Fig. B13, but at latitude 60°.

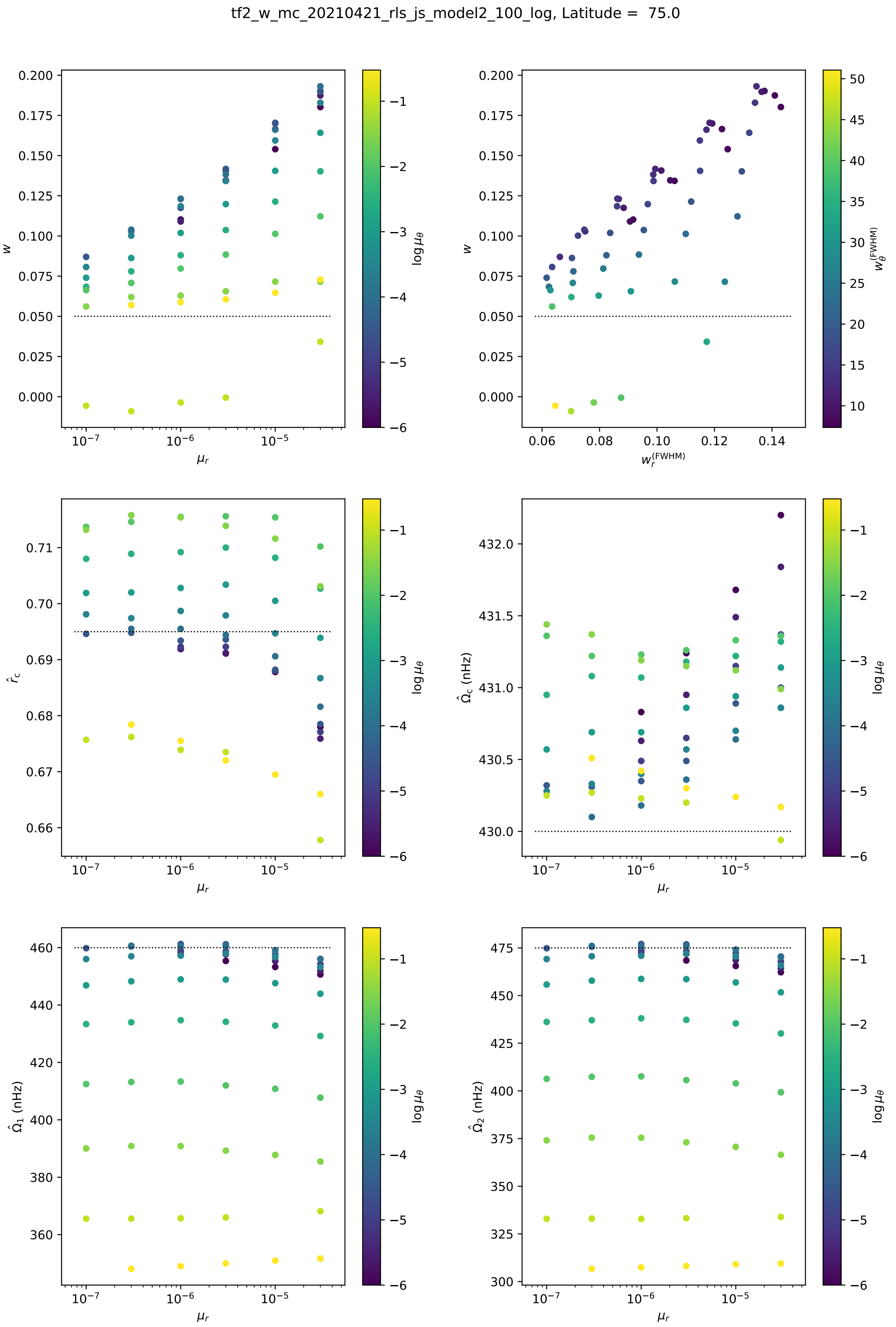


Figure B18: As Fig. B13, but at latitude 75°.

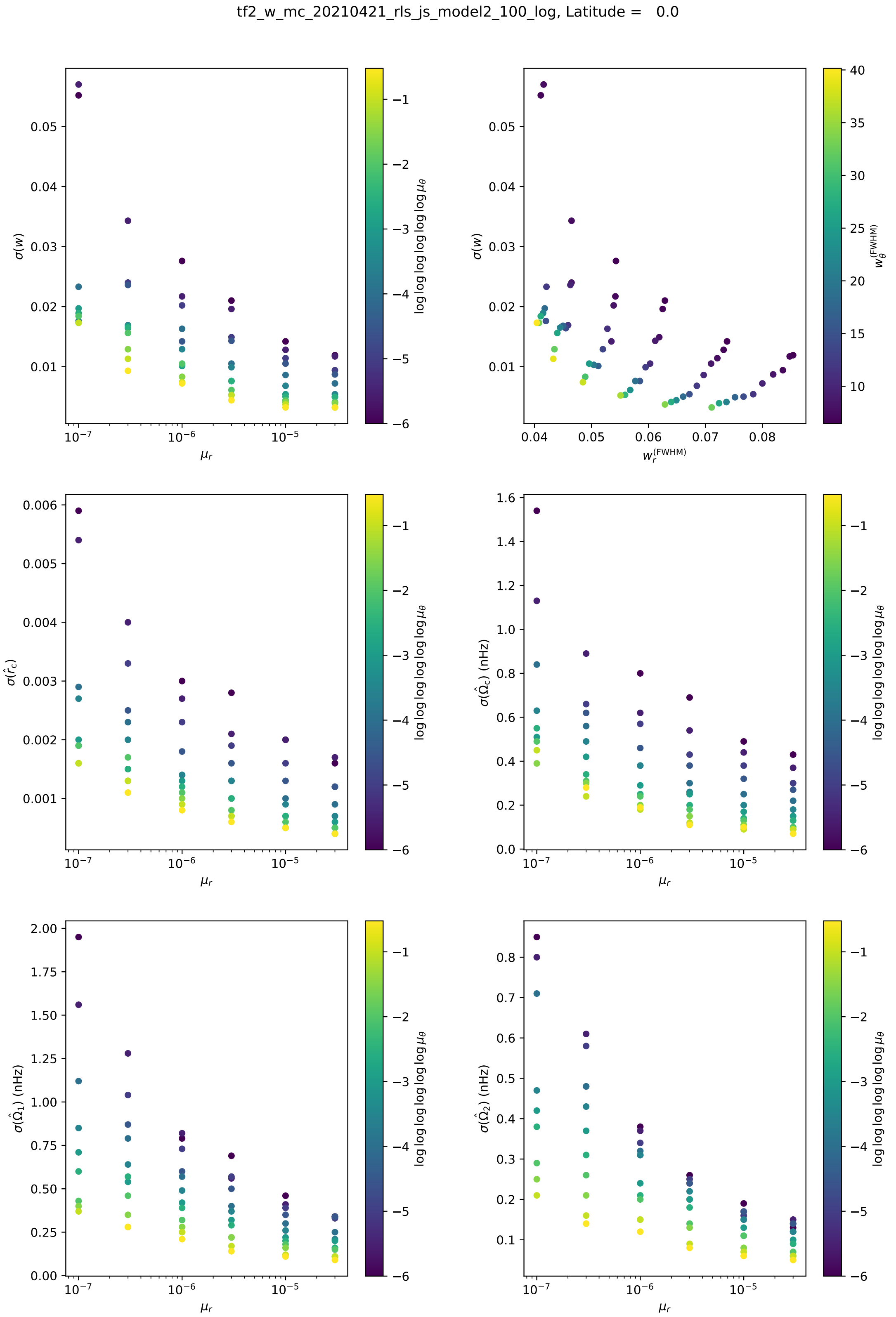


Figure B19: Standard deviations for the fits at the equator using 100 MC samples, shown in Fig. B13; see caption to Fig. 6 for details.

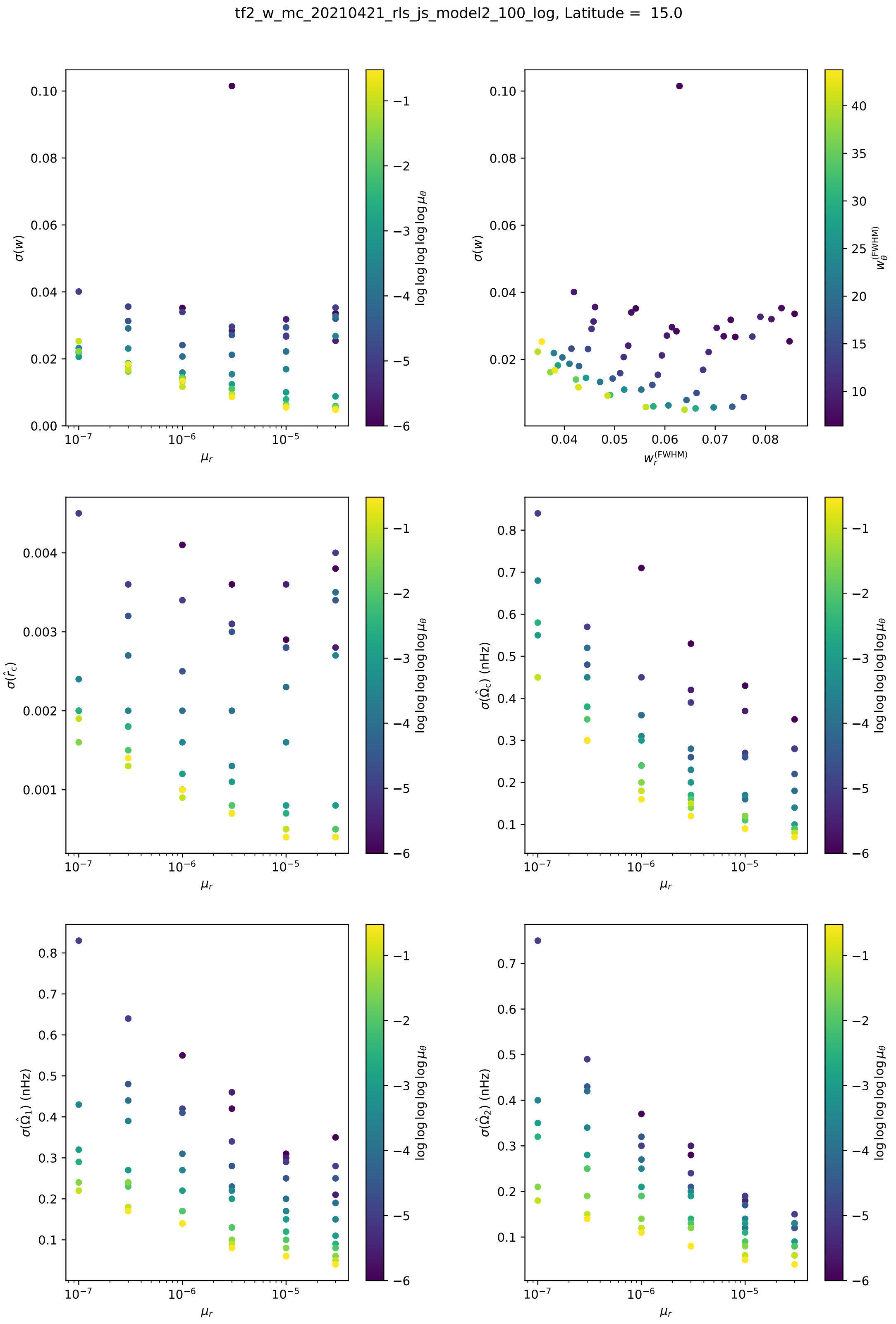


Figure B20: As Fig. B19, but at latitude 15°.

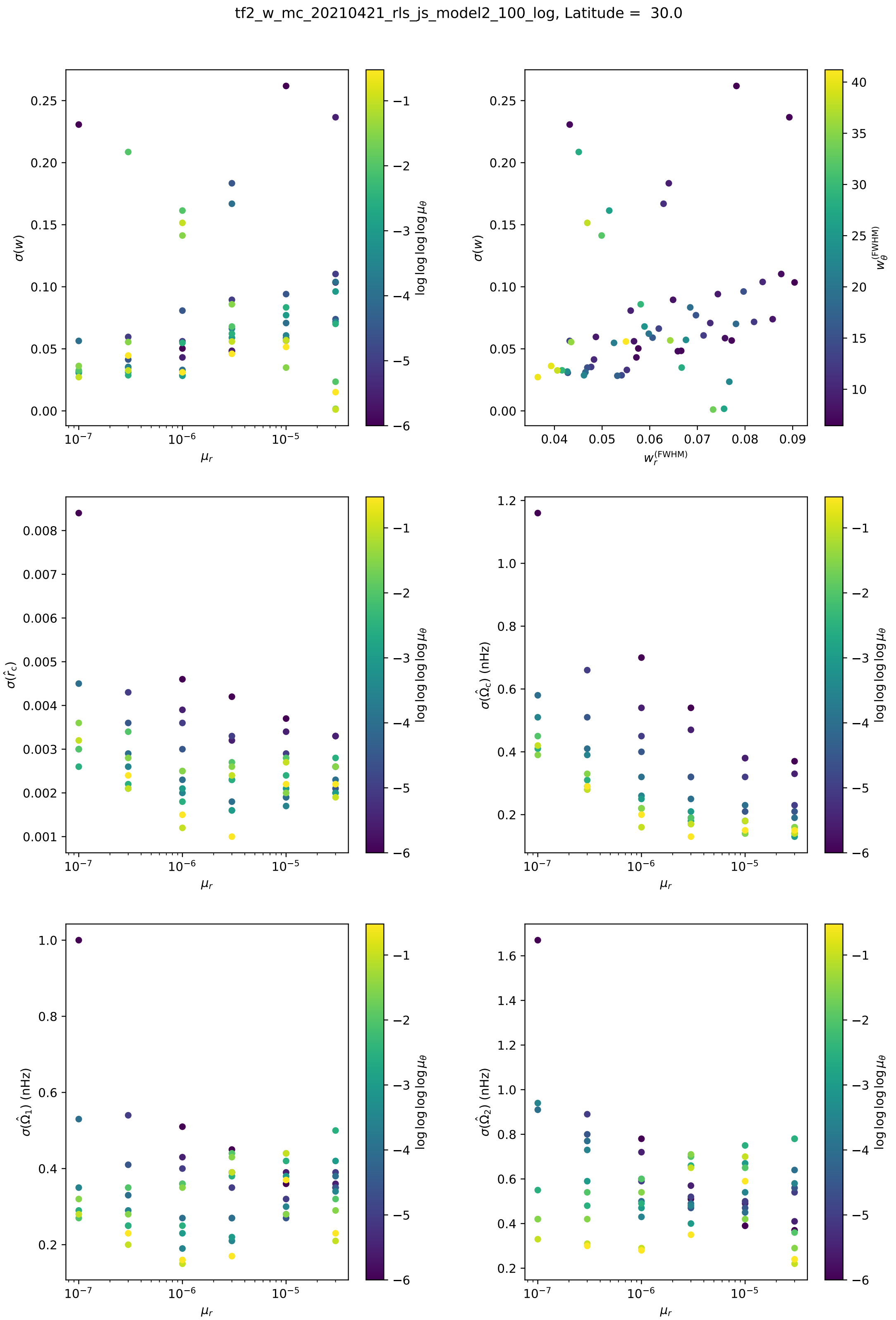


Figure B21: As Fig. B19, but at latitude 30°.

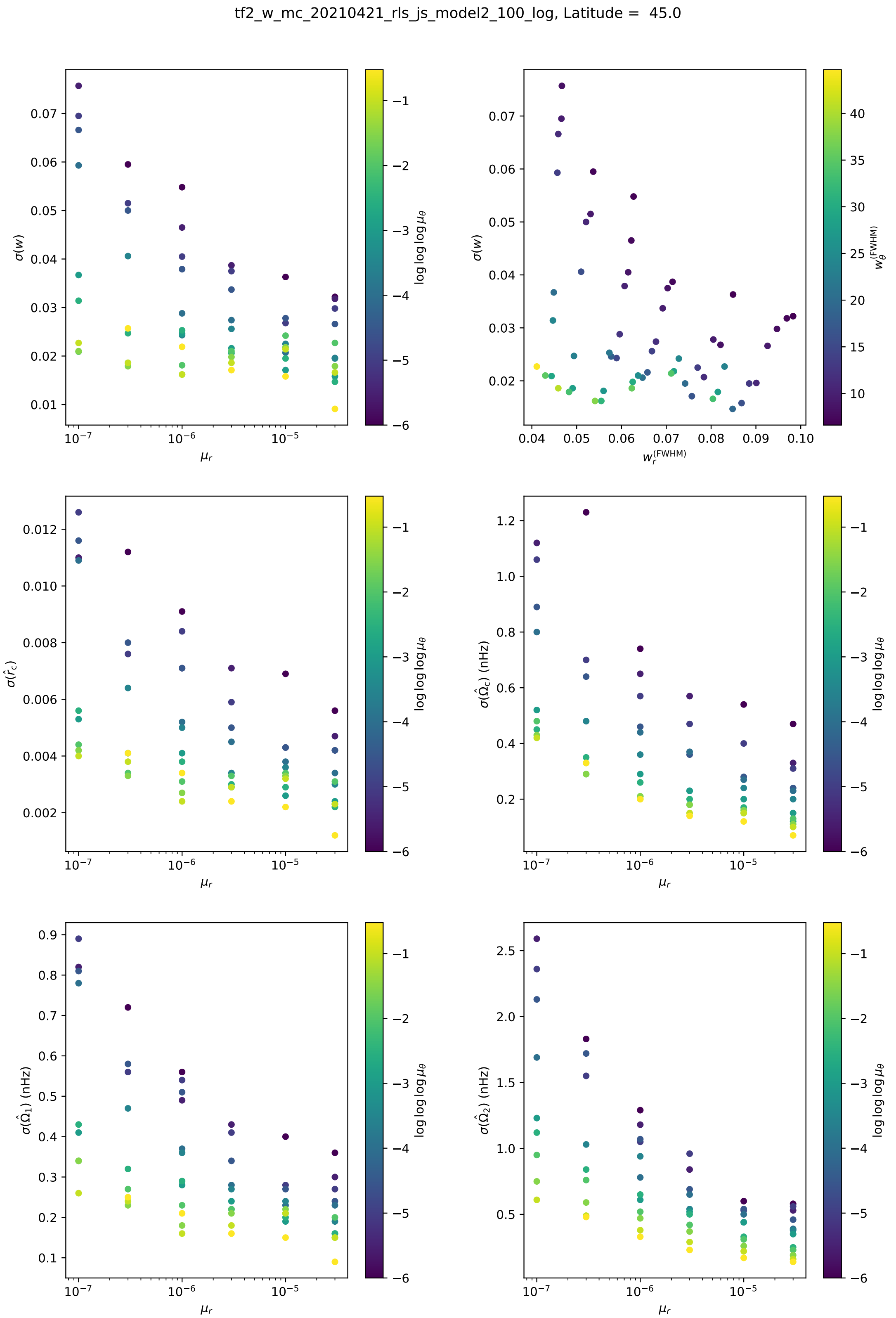


Figure B22: As Fig. B19, but at latitude 45°.

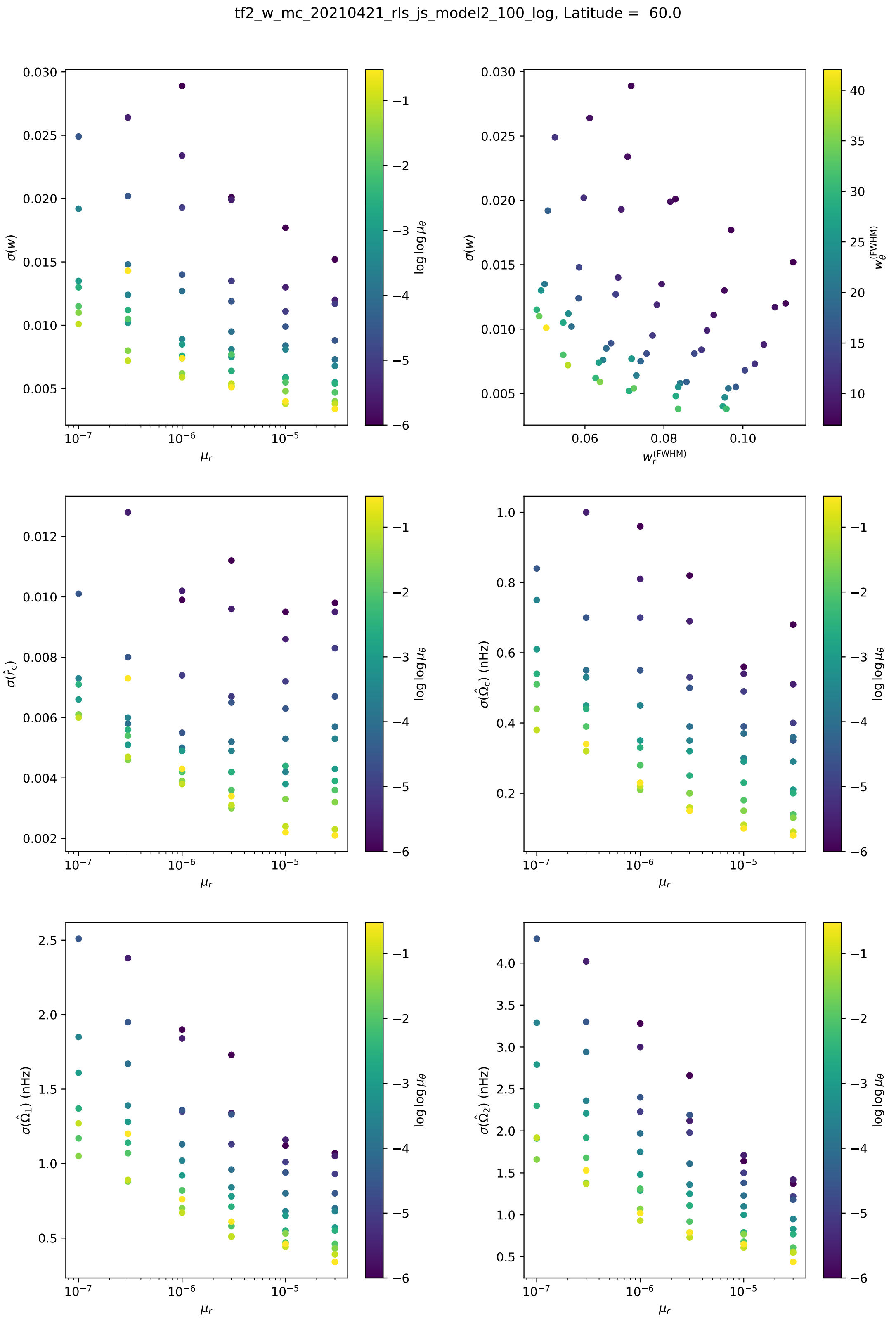


Figure B23: As Fig. B19, but at latitude 60°.

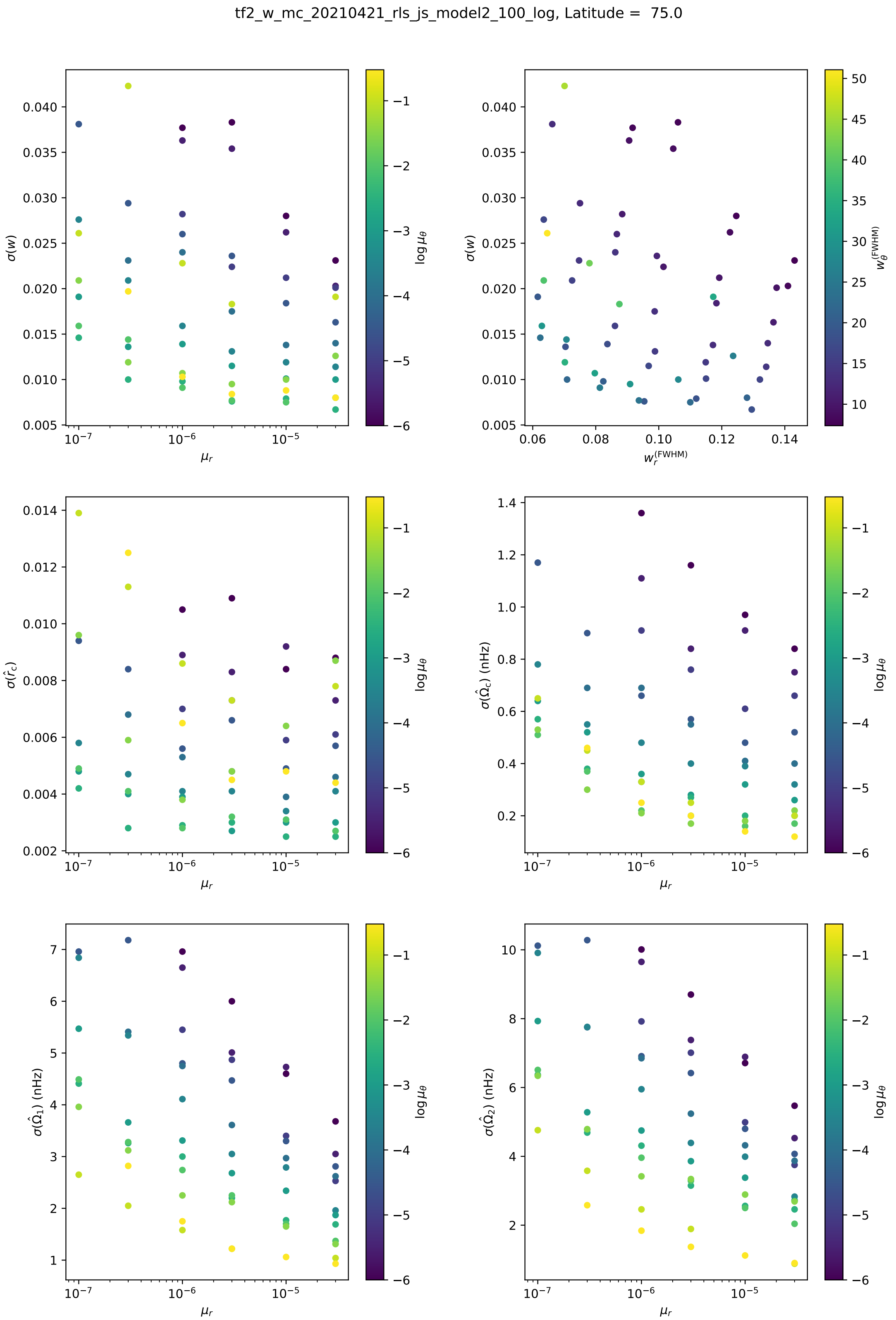


Figure B24: As Fig. B19, but at latitude 75°.