

# Wfmm User's Guide

This documents the basics of how to use the code provided for implementing the Bayesian wavelet-based functional mixed models methodology introduced in Morris and Carroll (2006). The code implements the Markov Chain Monte Carlo (MCMC) procedure described in Section 5 of the paper and outputs posterior samples for many model quantities.

[Morris, JS and Carroll, RJ \(2006\)](#). Wavelet-based functional mixed models, *Journal of the Royal Statistical Society, Series B*, 68(2): 179-199.

## Sample program call (from DOS window):

```
wfmm input.mat output.mat > log_file.log
```

**Input:** The program requires an input file (“input.mat” in the call) that is a matlab file that can specify the following objects:

1. **Y:**  $N$ -by- $T$  matrix, each row containing one of the observed functions on an equally-spaced grid of length  $T$ . This is the only variable that is required. Defaults will be taken for everything else if they are omitted.
2. **model:** Matlab structure indicating details of model. The following elements can be specified:
  - a. **X:**  $N$ -by- $p$  matrix containing the desired covariates for the  $p$  fixed effect functions in the model. Default:  $N \times 1$  column of 1's.
  - b. **Hstar:** integer indicating the number of levels of random effect functions to compute GCP-by-blocks, default: 0. This should only be used for very large  $m$ , ( $>70$ ).
  - c. **Z{h},  $h=1, \dots, H$ :**  $N$ -by- $m_h$  matrices containing covariates for each set of random effect functions. If omitted, it is assumed that it is a fixed effects model.
  - d. **C:**  $N$ -by- $c$  matrix, the columns indicate the functions that share a common residual error  $S$ ; by default this is an  $N \times 1$  matrix of all 1's.
3. **wavespecs:** Matlab structure describing wavelet settings to be used. The following elements can be specified:
  - a. **wavelet:** text string indicating the wavelet basis to use. Abbreviations for wavelet bases are given below. Default: 'db4'.
  - b. **nlevels:** integer indicating the number of levels of decomposition. If omitted, an optimal number of levels is computed.
  - c. **boundary:** boundary correction method assumed, default: 'periodic'.
  - d. **extendedMode:** whether (1) or not (0) to keep extra boundary wavelet coefficients (so  $K > T$ ) (1 by default)
  - e. **P:** fraction of “energy” retained during wavelet compression, default: 1 (no compression)

- f. **t**: number of functions that must be greater than threshold P in order to be retained during compression, default: 0 , included if any functions are greater than P.
4. **MCMCspecs**: Matlab structure describing details of MCMC. The following elements can be specified:
- a. **B**= number of MCMC samples to obtain, default: 1000.
  - b. **burnin**= burn-in length, default: 1000.
  - c. **thin**= thinning parameter; e.g. if 10, then keep every 10 samples in MCMC, default: 5.
  - d. **propvarTheta**= multiple of var(MLE) to use in proposal variance for variance components in step 2 of the MCMC, default: 1.5
  - e. **nj\_nosmooth**= number of lowest frequency wavelet levels for which we want a vague prior (no smoothing), default: 2.
- The following parameters are simply for numerical stability:
- f. **minp**= minimum value for any  $\pi_{ij}$ , default:  $10^{-14}$
  - g. **minT**= minimum value for  $T_{ij}$ , default: 1
  - h. **bigT**= value to use for  $T_{ij}$  when vague prior desired (no smoothing), default: 1000
  - i. **maxO** = maximum odds ratio, by default:  $10^{20}$  (prevents overflow)
  - j. **minVC** = minimum value of variance component, default:  $10^{-6}$  (prevents instability of variance components wandering near zero)
  - k. **VC0\_thresh** = minimum size for important variance component, default: 0.01.
  - l. **delta\_theta** = multiple for prior on theta: “number of datasets of information” in prior (see discussion in Morris, et al. (2003) JASA, 98:591-597), default:  $10^{-4}$ .
  - m. **thetaMOM\_maxiter** = maximum number of iterations in finding MOM starting values for variance components, default: 100
  - n. **thetaMOM\_convcrit**= convergence criteria for iterative procedure for finding MOM starting values for variance components, default:  $10^{-3}$
  - o. **time\_update**=number of iterations between updates to the log file during MCMC loop, default: 100.
  - p. **missing\_data**=flag indicating whether to process normal data Y or imputed data Vstar (0=normal, 1=imputed), default: 0.

## Output:

The output of the program is a Matlab data file (“output.mat” in the sample call), containing the following Matlab objects, as well as an input\_Init.mat containing results of the initialization phase of the computation. Error messages and status are written to standard output, which can be redirected to a log file. Processing status can be monitored by periodically typing the log file

The following variables are stored in the input\_Init.mat file:

- **Y, model, wavespecs, MCMCspecs**: copies of input variables

- **D**: matrix containing wavelet coefficients for observed data
  - **PI**: matrix containing the  $\pi_{ij}$  estimated by the Empirical Bayes procedure described in Section 4.4 of Morris and Carroll (2006), based on theta\_MLE.
  - **PI\_MOM**: matrix containing the  $\pi_{ij}$  estimated by the Empirical Bayes procedure based on theta\_MOM.
  - **Tau**: matrix containing the  $T_{ij}$  by the Empirical Bayes procedure described in Section 4.4 of Morris and Carroll (2006), based on theta\_MLE.
  - **Tau\_MOM**: matrix containing the  $T_{ij}$  by the Empirical Bayes procedure, based on theta\_MOM.
  - **theta\_MOM**: matrix containing method of moments starting values for the wavelet-space variance components  $q_{jk}$  and  $s_{jk}$  in model (3).
  - **theta\_MLE**: matrix containing profile maximum likelihood starting values for the wavelet-space variance components.
  - **se\_Theta**: estimate of the variance of theta\_MLE, to use in automatic proposal variances in Metropolis-Hastings procedure described in step (b) of Section 5 in Morris and Carroll (2006)
  - **betans**: matrix containing non-shrunken estimate of wavelet coefficients for fixed effects conditioning on starting values of variance components, given by equation (5) in Morris and Carroll (2006).
  - **Vbetans**: Matrix containing variance of these wavelet-spaced estimates, given by equation (6) in Morris and Carroll (2006).
  - **alpha**: Matrix containing starting values for shrinkages for wavelet coefficients for fixed effect functions, which are their posterior probabilities of being “nonzero”. Condition on theta\_MLE for variance components.
  - **alpha\_MOM**: same as alpha, only based on theta\_MOM.
  - **prior\_Theta\_a, prior\_Theta\_b**: matrices containing the prior hyperparameters for the inverse gamma distributions on the wavelet-space variance components
  - **Wv**: structure containing, for each wavelet coefficient, the following statistics, using starting values of the variance components for  $\Sigma_{jk}$ 
    - $XvX=X'(\Sigma_{jk})^{-1}X$
    - $XvZ=X'(\Sigma_{jk})^{-1}Z$
    - $XvD=X'(\Sigma_{jk})^{-1}D$
    - $ZvZ=Z'(\Sigma_{jk})^{-1}Z$
    - $ZvD=Z'(\Sigma_{jk})^{-1}D$
    - $dvd=\text{diag}(D'(\Sigma_{jk})^{-1}D)$
    - $L1=\det(\Sigma_{jk})$
    - $L2=(d_{jk}-X B_{jk})'(\Sigma_{jk})^{-1}(d_{jk}-X B_{jk})$
- where  $\Sigma_{jk}$  is the marginal variance of  $d_{jk}$ .

The following variables are stored in output.mat

**Y, model, wavespecs, MCMCspecs**: as input

**D**: matrix containing wavelet coefficients for observed data

- **PI:** matrix containing the  $\pi_j$  estimated by the Empirical Bayes procedure described in Section 4.4 of the paper Morris and Carroll (2006).
- **Tau:** matrix containing the  $T_{ij}$  by the Empirical Bayes procedure described in Section 4.4 of Morris and Carroll (2006).
- **betans:** matrix containing non-shrunken estimate of wavelet coefficients for fixed effects conditioning on starting values of variance components, given by equation (5) in Morris and Carroll (2006).
- **betahat:** betans\*alpha; shrinkage starting values for betas.
- **ghat:**  $p$ -by- $T$  matrix containing the posterior mean for each fixed effect function.
- **ghatns:** Inverse discrete wavelet transform of betans.
- **Q05\_ghat:**  $p$ -by- $T$  matrix containing the pointwise .05 quantile for each fixed effect function, which is the lower bound for the 90% posterior credible interval.
- **Q95\_ghat:**  $p$ -by- $T$  matrix containing the pointwise .95 quantile for each fixed effect function, which is the upper bound for the 90% posterior credible interval.
- **theta:** Mean of MCMC theta samples.

MCMC samples are output in binary double precision format, one file for each variable with filename *Input\_variablename.dat*:

- *Input\_wbeta.dat*: file containing MCMC posterior samples for wavelet coefficients for fixed effects. Each row contains all  $B_{ijk}^*$ ,  $i=1, \dots, p, j=1, \dots, J, k=1, \dots, K_j$ , for one iteration of MCMC.
- *Input\_beta.dat*: file containing MCMC posterior samples for data-space fixed effect functions. Each row contains  $B_{ij}$ ,  $i=1, \dots, p, j=1, \dots, T$ , for one iteration of MCMC.
- *Input\_theta.dat*: file containing MCMC posterior samples for variance components in wavelet space. Each row contains  $\{q_{hjk}, h=1, \dots, H, s_{jk}\}$ , in that order, for  $j=1, \dots, J, k=1, \dots, K_j$
- *Input\_newtheta.dat*: file containing Metropolis-Hastings acceptance probabilities for the set of variance components for each wavelet coefficient, indexed by scale  $j$  and location  $k$ .

### Comments:

- This code has been compiled and used on Microsoft Windows systems. In the near future, we will also test it on Linux, as well.
- The current interface assumes you create the input files and want to post-process the output files in Matlab.
- The current version of the code assumes:
  1. You want to estimate the shrinkage hyperparameters using the empirical Bayes method.
  2. You want vague proper priors for the variance components, centered at the starting values with information equivalent to delta\_Theta observations.
  3. The random effect functions are independent and identically distributed, so  $P=R=I$

- This code yields MCMC samples for the quantities in the wavelet-space model, (3) in Morris and Carroll (2006), plus MCMC samples for the fixed effect functions  $B$  in the data space model (2). You will notice a number of output files with \*.dat extensions. These are binary scratch files storing the MCMC samples so that they do not overflow memory.
- MCMC samples of  $Q_h$  and  $S_i$  matrices can be obtained by applying the 2-D IDWT to the corresponding diagonal wavelet-space matrices. They are generated only for  $T < 1000$ , since their large size will cause memory issues in large data sets.
- MCMC samples of the random effect functions and their wavelet coefficients are not calculated by default, again to save memory and computing resources. We have matlab scripts to compute these that we are in the process of converting over to C, and will post once they are complete.
- Our scripts for post-processing the posterior samples from the MCMC to do various types of Bayesian inference are in Matlab. We are in the process of converting some of them over to C, at which time they will be posted along with their documentation.
- For extremely large data sets, one may need to compute the memory requirements of fitting their data and compare it with the capabilities of their machine. See formulas below for estimating RAM and disk usage.
- **Wavelet Bases:** The current version accepts both columns of abbreviations for the wavelet bases specified below. Here are the available wavelets, and the corresponding notations for wfmm and Matlab:

WFMM wavelets	Matlab equivalent
"haar",	"db1" or "haar"
"d4",	"db2"
"d6",	"db3"
"d8",	"db4"
"d10",	"db5"
"d12",	"d6"
"d14",	"db7"
"d16",	"db8"
"d18",	"db9"
"d20",	"db10"
"s4",	"sym2"
"s6",	"sym3"
"s8",	"sym4"
"s10",	"sym5"
"s12",	"sym6"
"s14",	"sym7"
"s16",	"sym8"
"c6",	"coif1"
"c12",	"coif2"
"c18",	"coif3"
"c24",	"coif4"
"c30"	"coif5"

Estimating disk and RAM usage:

N = Number of Functions  
 p = Number of fixed effect functions  
 T = Number of observations/function  
 B = Number of MCMC samples  
 K = Number of wavelet coefficients  
 K' = Number of non-thresholded wavelet coefficients  
 m = Number of random effect functions    H = Number of levels of random effect functions  
 c = Number of strata for residual error functions

**Disk Usage  $\approx 8 [BK'(p+H+c+1) + pT(B+4) + 3NT + 2NK' + K'(p^2+m^2+pm+m+3+7p+7(H+c)) ]$**

**RAM Usage  $\approx 8[2pT+(H+c+p+2)K'+T + (0.05B+4)pT +6T+ (4(H+c)+2p)K']$**

Parallel processing

The processing has also been divided into three executables for initialization (wfmm1), MCMC loop (wfmm2), and postprocessing (wfmm3). This allows multiple MCMC chains to be run simultaneously using a grid computing resource like Condor, and have their results combined in the postprocessing step. Their command line arguments are:

*wfmm1 input.mat > log\_file.log*

This takes the same *input.mat* as input and outputs a *input\_Init.mat* file as described above.

*wfmm2 input\_Init.mat output*

Takes the *input\_Init.mat* file as input and outputs MCMC samples as *output\_variablename.dat* binary files

*wfmm3 input\_Init.mat output output\_summary.mat number\_of\_files*

The following is an example of a parallel processing bat file for Condor using these three executables. It relies only on a command to submit jobs to the grid (condor\_submit), and a command to wait until all of the submitted jobs have run (condor\_wait):

```

wfmm1 %1.mat > %1_init.log
condor_submit -a Dataset=%1 -a ThreadNumber=%2 wfmm_condor.sub
condor_wait %1.log
wfmm3 %1_Init.mat %1_results %1_summary.mat %2 > %1_summary.log
  
```

%1 (first argument of the bat file) is filename of the input mat file, %2 (second argument of the bat file) is number of parallel jobs requested for the MCMC computation.

The condor\_submit command also requires a submit file that describes the jobs. The filename and number of jobs parameters are passed to the condor submit file as parameters Dataset and ThreadNumber. An example file is shown below.

```
# A basic submit file

# On Windows the universe is vanilla
universe = vanilla

# Set the executable name here
executable = WFMM2.exe

# Set command line arguments here
arguments = $(Dataset)_Init.mat $(Dataset)_results_$(Process).mat

# Set requirements here (memory, OS, etc.)
requirements = (OpSys == "WINNT40" || OpSys == "WINNT50" || OpSys ==
"WINNT51") && (memory > 1000)

# List the input files here
transfer_input_files = $(Dataset)_Init.mat, Z:\bin\icudt241.dll,
Z:\bin\icuin24.dll, Z:\bin\icuio24.dll, Z:\bin\icuc24.dll,
Z:\bin\libmat.dll, Z:\bin\libmx.dll, Z:\bin\libut.dll, Z:\bin\libz.dll,
Z:\bin\msvc71.dll, Z:\bin\msvcr71.dll, Z:\bin\libguide40.dll

# Leave this alone
transfer_files = ALWAYS

# You can rename these files, but be sure they're defined
# These may be useful for debugging purposes
output = $(Dataset)_$(Process).txt
error = $(Dataset).err
log = $(Dataset).log

# Set the number of copies to submit here
queue $(ThreadNumber)
```

These files should be adaptable to any grid computing system.