User's Manual for the code in Matlab

The code in `Matlab` and the User's Manual can be found at GitHub with the user's name `YalinR`.

1 Generating Random Variables from 8 UARS Distributions

In the code, α, β and γ specifying the location parameter $\boldsymbol{S}$ are indicated by `a,b,g` respectively. $\boldsymbol{S}$ is indicated by `S`. The concentration parameter κ is indicated by `k`.

The ranges for α, β, γ and κ are as follows: $\alpha \in [0, 2\pi], \beta \in [0, \pi], \gamma \in [0, 2\pi], \kappa \in (0, \infty)$.

(1) Function used: `randraw.m vmUARS.m`
(`randraw.m` is downloaded online: https://www.mathworks.com/matlabcentral /fileexchange/7309-randraw/content/randraw.m by Alex Bar-Guy Copyright (c) 2005, Alex Bar-Guy All rights reserved.)
`[O,S]=vmUARS(k,a,b,g,n)` generates n random variables from the Von Mises UARS distribution with location parameter `S (a, b, g)` and concentration parameter `k`. `S` is the calculated parameter from `a, b, g`. `O` are the generated `n` number of 3 by 3 matrix. The dimension of `O` is $3n$ by 3.

(2) Function used: `randraw.m`,
`densityMF.m`, `MFUARS.m`, `MFUARSv.m`, `MFUARSfinal.m`
(`randraw.m` is downloaded online,see above)
`[O,S]=MFUARSfinal(k,a,b,g,n)` generates `n` random variables from the Matrix Fisher UARS distribution with location parameter `S (a, b, g)` and concentration parameter `k`. `S` is the calculated parameter from `a, b, g`. `O` are the generated `n` number of 3 by 3 matrix. The dimension of `O` is $3n$ by 3.

(3) Function used: `wnUARS.m`
`[O,S]=wnUARS(k,a,b,g,n)` generates `n` random variables from the wrapped normal UARS distribution with location parameter `S (a, b, g)` and concentration parameter `k`. `S` is the calculated parameter from `a, b, g`. `O` are the generated `n` number of 3 by 3 matrix. The dimension of `O` is $3n$ by 3.

(4) Function used: `rwmb.m WTNUARS.m`
`[O,S]=WTNUARS(k,a,b,g,n)` generates `n` random variables from the wrapped

trivariate normal UARS distribution with location parameter `S (a, b, g)` and concentration parameter `k`. `S` is the calculated parameter from `a, b, g`. `O` are the generated `n` number of 3 by 3 matrix. The dimension of `O` is $3n$ by 3.

(5) Function used: `densitywmb.m rwmb.m BungeUARS.m`
`[O,S]=BungeUARS(k,a,b,g,n)` generates `n` random variables from the Bunge UARS distribution with location parameter `S (a, b, g)` and concentration parameter `k`. `S` is the calculated parameter from `a, b, g`. `O` are the generated `n` number of 3 by 3 matrix. The dimension of `O` is $3n$ by 3.

(6) Function used: `densitylorentz.m LorentzianUARS.m`
`[O,S]=LorentzianUARS(k,a,b,g,n)` generates `n` random variables from the Lorentzian UARS distribution with location parameter `S (a, b, g)` and concentration parameter `k`. `S` is the calculated parameter from `a, b, g`. `O` are the generated `n` number of 3 by 3 matrix. The dimension of `O` is $3n$ by 3.

(7) Function used: `densitycayley.m densitywmb.m rwmb.m CayleyUARS.m`
`[O,S]=CayleyUARS(k,a,b,g,n)` generates `n` random variables from the Cayley UARS distribution with location parameter `S (a, b, g)` and concentration parameter `k`. `S` is the calculated parameter from `a, b, g`. `O` are the generated `n` number of 3 by 3 matrix. The dimension of `O` is $3n$ by 3.

(8) Function used: `densityig.m densitywmb.m rwmb.m igUARS.m`
`[O,S]=igUARS(k,a,b,g,n)` generates `n` random variables from the isotropic Gaussian UARS distribution with location parameter `S (a, b, g)` and concentration parameter `k`. `S` is the calculated parameter from `a, b, g`. `O` are the generated `n` number of 3 by 3 matrix. The dimension of `O` is $3n$ by 3.

2 Bayesian Inference for Parameters in Matrix Fisher UARS Distribution

(1) Bayesian estimate and credible set for the location parameter `S` and the concentration parameter `k`, acceptance rates for the two parameters
Function used: `SUMO.m posteriordensityMFUARS.m mcmcburnin.m`
`[Shat,S95,Khat,K025,K975,rate1,rate2]`
`=mcmcburnin(k,a,b,g,delta,sigma,n,OS,m,burnin)` returns a Bayesian estimate `Shat`, a 95% credible set for `S`, a Bayesian estimate of `k`, a 95% credible set for `k` after one simulates `m` number of `S`s and `k`s from the posterior distribution and discards the first `burnin` number of `S`s and `k`s. rate1

and rate2 are the acceptance rates for `S` and `k`, respectively. In the function, `k` is the starting value for κ. `a, b, g` can decide the starting value for $\boldsymbol{S}$. `delta` is the parameter for r in Metropolis-Hastings within Gibbs algorithm. `sigma` is the parameter for the proposal of κ. `OS` is the data set used for the Bayesian inference. `n` is the number of 3 by 3 matrices included in the data set `OS`. The dimension of `OS` is $3n$ by 3. The credible set for $\boldsymbol{S}$ is viewed in terms of the set of 3 cones of constant angle `S95` centered at `Shat`. (I used the method of Humbert et al.(1996) to get `Shat`).

(2) Trace plots (to get the burn-in value)
Function used: `mcmc.m getvector.m plots.m`
`[h,kj,Shatm1,Shatm2,Shatm3,Shatm4,Shatm5,`
`Shatm6,Shatm7,Shatm8,Shatm9]`
`=getvector(k,a,b,g,delta,sigma,n,OS,m)` generates the results needed for the trace plots. The explanation for `k, a, b, g, delta, sigma, n, OS, m` is the same as that in (1). `plots.m` generates the trace plots. After the trace plots are generated, we can observe the minimum value of iteration when each plot is stable. Then the maximum value for all the 10 minimum values from the 10 trace plots is the burn-in value which can be used for (1).