##
Written by **Prateek Bansal** and **Vahid Keshavarzzadeh**
Last updated: July 2021
##

This folder includes MATLAB and R codes to replicate the results of the Monte Carlo study and Empirical study of the paper titled as "Designed Quadrature to Approximate Integrals in Maximum Simulated Likelihood Estimation". For empirical study, we provide all codes and output but do not provide the input data due to confidentiality concerns. However, it has been shared with the editor for the replicability check. We have tested MATLAB codes on MATLAB R2020b version and R codes on R-4.0.1 (OS: Window 10). No additional toolbox/package was required to run MATLAB codes.

This folder has four sub folders and files in each of these folders are described below:

1. Generate DQ: To generate designed quadrature (DQ) rules.

2. Simulation: MATLAB codes to conduct Monte Carlo study (uses "1. Generate DQ" output as input).

3. Simulation Table and Figure: Process simulation results to generate Tables 1 to 3 (Uses "2. Simulation" output as input). R code to generate Figure 1 (a way to visualize results of Tables 1 to 3) is also included.

4. Empirical study: This folder includes MATLAB codes and output files to replicate the empirical component of the paper. We provide instructions to generate the results presented in Tables 5 and 6 of the paper.

######################################
############ **1. Generate DQ** ###########
######################################

main.m: The user needs to specify the dimension, order of polynomial integration, and the number of nodes. DQ rules will be output in an excel file.

Changing from Gaussian to Uniform case: In the current setting the code generates Gaussian nodes but for the uniform case, the user should make the following changes:

- In the function quad_int_mul_u_sens.m, pol_mul_g should be replaced by pol_mul_jacobi for uniform case.
- In the generator.m, the fixed regularization parameters are provided within the generator.m file. For uniform case, uncomment the uniform part and comment the Gaussian part.

Remarks: Changing the number of nodes and regularizations affect the results. As a general comment, higher number of nodes are more likely to yield convergence at a given error tolerance. In other words, if the number of nodes does not work at a given polynomial order, the user can repeat the design with higher number of nodes. Regularization can also impact the solution. Using higher regularization makes the code slower but it produces more stable solution.

######################################
############### **2. Simulation** ##########
######################################

Tables 1, 2, and 3 correspond to random parameters (dimension of integral) 3, 5, and 10, respectively. All three Simulation studies follow the same naming conventions, and therefore, we describe specific files related to the Monte Carlo Study of Mixed Logit Model with 5 random parameters (i.e. Table 2). We write all codes with analytical gradients of the loglikelihood.

DIAG_5rand.m: Main code for diagonal variance-covariance matrix (includes data generation and estimation).

NONdiag_5rand.m: Main code for full (non-diagonal) variance-covariance matrix(includes data generation and estimation).

#########################
Inputs for above codes:
#########################

flexll_grad.m: Function to compute likelihood and gradient of the likelihood function for Halton and MLHS.

flexll2008_grad.m: Function to compute likelihood and gradient of the likelihood function for Designed Quadrature.

makedraws.m: Function to generate standard normal draws from Halton and MLHS.

multiprod.m: Function performs multiple matrix products, with array expansion enabled.

sim_plan_dq.xlsx: Simulation plan for DQ

TO_d_5_accu_6_node_50.xlsx: DQ rules for accuracy (order) 6, and 50 nodes (consistent naming convention), generated using main.m in folder "1. Generate DQ".

#########################
Outputs for above codes:
#########################

DIAG_5rand_output.mat: Output of the Monte Carlo study for diagonal variance-covariance matrix.

NONdiag_5rand_output.mat: Output of the Monte Carlo study for full (non-diagonal) variance-covariance matrix.

##
########## **3. Simulation Table and Figure** ###########
##

"2. Simulation" will provide 6 output files which will be input for this code. Since run times of simulation study presented in the folder "2. Simulation" for 3, 5, and 10 random parameters are 3 hours, 15 hours, 80 hours, respectively for each diagonal and full covariance scenario, we directly provide all six output files of the folder "2. Simulation" in this folder. We have then included code to generate each table which process the simulation output to summarise logliklihood, absolute percentage bias (APB), estimation time, loglikelihood function counts, and t-value (column headings of Tables 1 to 3). In this process, we prepare files to create the Figure 1 and use them as input for R code.

Table_1.m: This code geenrates 8x20 matrix (exactly replicate Table 1). User should look at variable "table_1" in workspace (NAN implies blank cells). This code also results into three_d.xlsx and three_full.xlsx files to create Figure 1.

Table_2.m: This code geenrates 10x25 matrix (exactly replicate Table 2). User should look at variable "table_2" in workspace (NAN implies blank cells). This code also results into five_d.xlsx and five_full.xlsx files to create Figure 1.

Table_3.m: This code geenrates 6x25 matrix (exactly replicate Table 3). User should look at variable "table_3" in workspace (NAN implies blank cells). This code also results into ten_d.xlsx and ten_full.xlsx files to create Figure 1.

plot_DQ.r: This R code takes three_d.xlsx, three_full.xlsx, five_d.xlsx, five_full.xlsx, five_d.xlsx, five_full.xlsx as input and provide plot.png as output.

plot.png: This file is the output of the R code which is slightly processed to produce Figure 1 in the manuscript.

######################################
########## **4. Empirical study** ###########
######################################

This folder includes MATLAB codes and output files to replicate the empirical component of the paper. We provide instructions to generate the results presented in Tables 5 and 6 of the paper:

automation.xlsx: It is the input data (not shared with the replication package). More details about the experiment design can be found in Liu et al. (2019): Liu, Y., P. Bansal, R. Daziano, and S. Samaranayake (2019). A framework to integrate mode choice in the design of mobility-on-demand systems. Transportation Research Part C: Emerging Technologies 105, 648-665.

automation.m: It is the main file which runs the estimation.

automation_output.mat: This is the output file which is generated by running automation.m. Running autonomous.m takes around 30 hours. So, we provided the output file for replication.

post_estimation_analysis_emp.m: It takes automation_output.mat as input. After running this code, the user can look for workspace variables table_5 and table_6 corresponding to tables 5 and 6 in the manuscript. NAN in table_5 implies blank cell.

Remark: All other excel files in this folder are related to designed quadrature rules and matlab files are functions which are used in automation.m. MATLAB functions are detailed in the description of “2. Simulation” folder.