

QUEST+ for MATLAB Cheat Sheet

written by Anthony Bruno

January 2022

Implementing QUEST+ in MATLAB requires the use of **three** main functions to run an experiment:

(*This is an example of running QUEST with a **cumulative normal** probability function. The psychometric function parameters may differ depending on the probability function used to generate the QUEST struct)

1. `qpInitialize(stimulus parameters, psychometric function parameters, probability function)`

- a. This function is called when creating a struct for one QUEST procedure. An infinite amount of QUEST structs can be created and used in a single program, perhaps if a subject completes multiple experimental conditions
- b. **stimulus parameters** refer to the possible stimulus values you want to allow QUEST to show during a trial block
 - i. QUEST can handle multiple stimulus dimensions (i.e., intensity *and* pitch) but only one dimension is needed (i.e., temporal interval)
- c. **psychometric function parameters** refer to the range of values for mean, slope, and lapse that the experimenter thinks will account for subjects' perceptual performance
 - i. Mean/PSE – the 50% point on the psychometric function
 - ii. Slope/threshold/SD – the distance from the 50% point to 80% point on the psychometric function
 - iii. Lapse – the rate of error due to factors outside perceptual performance (helps make the psychometric function more accurate)
- d. **probability function** refers to whether the psychometric function will be fit according to a cumulative normal, Weibull, or logit distribution
- e. How does this look in MATLAB?
 - i. `questStruct = qpInitialize('stimParamsDomainList', {stimulus list(s)}, 'psiParamsDomainList', {mean list, slope list, lapse list}, 'qpPF', @qpPFNormal);`

2. `qpQuery(quest struct)`

- a. This function is called when you want to obtain the value of the test stimulus to be presented on the current trial
- b. **quest struct** refers to the QUEST struct you wish to query to obtain the test value for the current trial. Below, I am using the hypothetical questStruct I created to query for the next stimulus value
- c. How does this look in MATLAB?
 - i. `stimulusOnThisTrial = qpQuery(questStruct);`

3. `qpUpdate(quest struct, stimulus, outcome)`

- a. This function is called after a trial ends and a response is recorded; the stimulus value and the outcome on the current trial (NOT correct/incorrect) is passed to the QUEST struct so that a good stimulus value can be suggested on the next trial.
 - b. **outcome** refers to the participant's response with respect to the task
 - i. For example, for a yes/no task the outcome should be linked to yes/no NOT whether the response is correct
 - ii. if yes, outcome = 2
 - iii. if no, outcome = 1
 - c. How does this look in MATLAB?
 - i. **questStruct = qpUpdate(questStruct, stimulusOnThisTrial, outcome);**
-

What is QUEST+ doing while a participant is completing a block of trials?

In short:

- A 3 dimensional matrix is created from the psychometric function parameters given to QUEST when a struct is initialized
 - The rows of the matrix represent different combinations of psychometric function parameters
 - As trial data is recorded and the struct is updated, QUEST assigns probabilities to the different combinations of psychometric function parameters
 - Once a block of trials is completed, the combination of psychometric function parameters with the highest probability is the 'winner' and represents the best psychometric function fit for that subject
 - Since QUEST is trying to find the best fit while presenting stimuli that are good for any subject, it is important to consider the relationship between the stimulus parameters you define and the psychometric function parameters such as the mean
-

Once you've collected data using the QUEST procedure, a few more functions are needed to obtain a subject's psychometric function fit:

1. **qpListMaxArg(posterior)**

- a. This function is called to obtain the row number of psychometric function parameters that was assigned the highest probability and therefore best fit for the current subject
- b. **posterior** refers to the list in the QUEST struct that stores the probabilities for combinations of psychometric function parameters
- c. How does this look in MATLAB?
 - i. **psiParamsIndex = qpListMaxArg(questStruct.posterior);**

2. **psiParamsDomain(psychometric parameters index)**

- a. This function is called to obtain the values for mean, slope, and lapse that belong to the 'best fit' row in the psychometric function parameter matrix

- b. **psychometric function parameters index** refers to the value obtained from `qpListMaxArg()` that points to the row with the combination of psychometric function parameters that best fits the trial data
 - c. How does this look in MATLAB?
 - i. `psiParamsQuest = questStruct.psiParamsDomain(psiParamsIndex,:);`
3. **qpFit(trial data, probability function, psychometric parameters index, number of outcomes, lower bounds, upper bounds)**
- a. This function is called to find the maximum likelihood fit while using the psychometric function parameters with the highest probability as the starting point
 - i. A lot of the arguments in the `qpFit` function call are variables that exist in the QUEST struct (i.e., trial data, probability function, number of outcomes)
 - b. **number of outcomes** refer to the number of possible responses a subject can make (default is 2 which is why it doesn't need to be adjusted in `qpInitialize` for a yes/no experiment)
 - c. **lower and upper bounds** refer to the minima and maxima of the psychometric function parameters defined when the QUEST struct is initialized
 - d. How does this look in MATLAB?
 - i. `psiParamsFit = qpFit(questStruct.trialData, questStruct.qpPF, psiParamsQuest, questStruct.nOutcomes, 'lowerBounds', [], 'upperBounds', []);`

(Refer to `qpQuestPlusNormCdfDemo.m` for a good demonstration of these functions as well as code for plotting the psychometric function given the trial data)