Behavior Research Methods
https://doi.org/10.3758/s13428-022-01948-8

jsQuestPlus: A JavaScript implementation of the QUEST+ method for estimating psychometric function parameters in online experiments

Daiichiro Kuroki[1] · Thomas Pronk[2]

Accepted: 1 August 2022

Abstract

The two Bayesian adaptive psychometric methods named QUEST (Watson & Pelli, 1983) and QUEST+ (Watson, 2017) are widely used to estimate psychometric parameters, especially the threshold, in laboratory-based psychophysical experiments. Considering the increase of online psychophysical experiments in recent years, there is a growing need to have the QUEST and QUEST+ methods available online as well. We developed JavaScript libraries for both, with this article introducing one of them: jsQuestPlus. We offer integrations with online experimental tools such as jsPsych (de Leeuw, 2015), PsychoPy/JS (Peirce et al., 2019), and lab.js (Henninger et al., 2021). We measured the computation time required by jsQuestPlus under four conditions. Our simulations on 37 browser–computer combinations showed that the mean initialization time was 461.08 ms, 95% CI [328.29, 593.87], the mean computation time required to determine the stimulus parameters for the next trial was less than 1 ms, and the mean update time was 79.39 ms, 95% CI [46.22, 112.55] even in extremely demanding conditions. Additionally, psychometric parameters were estimated as accurately as the original QUEST+ method did. We conclude that jsQuestPlus is fast and accurate enough to conduct online psychophysical experiments despite the complexity of the matrix calculations. The latest version of jsQuestPlus can be downloaded freely from https://github.com/kurokida/jsQuestPlus under the MIT license.

Keywords Online experiments · Psychophysical threshold · Psychometric functions · Adaptive psychometric methods

Introduction

Adaptive psychometric procedures enable experimenters to estimate thresholds efficiently by determining the stimulus parameters based on the stimuli and the observer's responses in the preceding trials (for reviews, see Leek, 2001; Treutwein, 1995). This efficiency is desirable for experiments in general, but even more so when faced with time constraints, such as in experiments on children and/or in clinical settings. Adaptive procedures have been refined over the decades and show enduring popularity, but as yet mainly in laboratory-based psychophysical research (e.g., Keefe et al., 2021; Kim & Chong, 2021; Levinson et al., 2021; Song et al., 2021; Yu & Postle, 2021).

On the other hand, there has been an increase in online psychophysical experiments (e.g., Bunce et al., 2021; Kawabe, 2021; Santacroce et al., 2021; Sasaki & Yamada, 2019). Santacroce et al. (2021) were not planning to do their experiments online but decided to conduct part of the experiments online due to COVID-19. In addition to efficient recruitment of a large number and wide variety of participants (Reips, 2021), online experiments can proceed regardless of any lockdown measures. Considering the rise in online psychophysical experiments, there is a growing need for online experiment tools to match the functionality offered in the lab. Specifically, we focus on online experiments as web applications that can be deployed via the Internet to web browsers on laptops, desktops, smartphones, and tablets. We focus on the former, because these types of online experiments are extremely widely supported and

✉ Daiichiro Kuroki
kurokid@kyudai.jp

[1] Department of Psychology, School of Letters, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan

[2] Behavioural Science Lab, Faculty of Social and Behavioural Sciences, University of Amsterdam, Amsterdam, Netherlands

Published online: 07 September 2022

based on durable open standards, such as HTML, CSS, and JavaScript (Pronk et al., 2020).

Given the popularity of adaptive procedures, these are excellent candidates to offer online. For one of the older and most well-known adaptive methods, the up-down staircase (Levitt, 1971), some excellent online implementations are already available (e.g., Hadrien & Jaquiery, 2016; Hirst, 2020). A more modern method is based on a Bayesian framework introduced by Watson and Pelli (1983), which combines the experimenter's prior knowledge of psychometric parameters with actual data obtained through a series of trials in the current experiment. The original staircase method based on this Bayesian framework, named QUEST (Watson & Pelli, 1983), assumes one stimulus dimension, two response options (e.g., Yes/No or Correct/Incorrect), and can estimate one psychometric parameter, usually a threshold. In the QUEST method, a unimodal probability density function (PDF), such as a Gaussian function, is assumed as a prior PDF as shown in the MATLAB-based program (Pelli, 1996). The PDF is updated every trial to best fit the stimulus intensities and responses in the preceding trials. The stimulus intensity for the next trial and the final estimate of the threshold are determined by using the mode of the current PDF (Watson & Pelli, 1983), the quantile (Pelli, 1987 as cited in Pelli, 1996), or the mean (King-Smith et al., 1994). The reference implementation of QUEST was written in BASIC, with a MATLAB version included in Psychtoolbox (Brainard, 1997; Pelli, 1997), and a Python version (Straw, 2008) included in PsychoPy (Peirce, 2007; Peirce et al., 2019).

Watson (2017) extended the QUEST method to allow multiple stimulus parameters, multiple psychometric parameters, and more than two responses options. The QUEST+ method calculates the expected entropies of the PDF of the psychometric parameters, selecting stimulus parameters that minimize the expected entropy for the next trial (Kontsevich & Tyler, 1999; Watson, 2017). Finally, the parameters with the highest PDF are regarded as the estimates. For the QUEST+ method, Watson's (2017) reference implementation is in Mathematica, followed by versions in MATLAB (Brainard, 2017; Jones, 2018) and Python (Höchenberger, 2019).

The QUEST and QUEST+ implementations above have been written in programming languages (BASIC, Mathematica, MATLAB, and Python) that are not compatible with web browsers. Hence, using any of these implementations in the context of an online study would require server-side infrastructure and client–server communication, thereby introducing significant complexity and possible latency. Alternatively, QUEST and QUEST+ implementations in JavaScript could run in a web browser, thereby having the potential to be simpler and easier to integrate into existing online task software.

Hence, we developed JavaScript implementations of both QUEST and QUEST+, named jsQUEST (Kuroki & Pronk, 2021a) and jsQuestPlus (Kuroki & Pronk, 2021b). Since the QUEST+ method is an extension of the QUEST method, one might think that there is no need to develop JavaScript libraries for both. While that assertion is correct, we had to be prudent because of two reasons. Firstly, experimenters may prefer the traditional QUEST method for a simple experiment in which there are a single stimulus parameter, a single psychometric parameter (e.g., threshold), and two response options because they can more concisely code their experiment compared to the QUEST+ method. Secondly, it is possible that the QUEST method is persistently used in the laboratory and hence is required when directly replicating studies from such a lab online. For these reasons, we have developed both jsQUEST and jsQuestPlus, but the remainder of this paper will focus primarily on the QUEST+ method (jsQuestPlus).

In the next section, we briefly introduce the core functionality of jsQuestPlus via an example from the paper introducing QUEST+ (Watson, 2017). In the associated GitHub repositories, we offer brief tutorials for integrating jsQUEST and jsQuestPlus into three major experiment libraries: jsPsych (de Leeuw, 2015), lab.js (Henninger et al., 2021), and PsychoJS (Peirce et al., 2019).

Functions of jsQuestPlus

The QUEST+ method consists of three parts: initialization, determining the stimulus parameters for the next trial, and updating the data (based on the response to a given stimulus). We will explain the details according to the second example, labeled as *Estimation of contrast threshold, slope, and lapse {1, 3, 2}*, in Watson (2017). In this example, there is a single stimulus parameter (contrast) and three psychometric parameters (threshold, slope, and lapse). A Weibull function is assumed to be the psychometric function, and the task is two-alternative forced choice. Using the QUEST+ method, the three psychometric parameters can be estimated.

To initialize the QUEST+ data, the psychometric functions corresponding to each response must be specified. For example, the function representing probabilities of incorrect responses (response = 0) can be written as follows.

```
function func_resp0(stim, threshold, slope, guess, lapse) {
  const tmp = slope * (stim - threshold) / 20
  return lapse - (guess + lapse - 1) * Math.exp(-Math.pow(10, tmp))
}
```

This describes the Weibull function, which is also available in jsQuestPlus as *jsquest.weibull*. The function representing probabilities of correct responses (response = 1) can be written as follows:

```
function func_resp1(stim, threshold, slope, guess, lapse) {
  return 1 - func_resp0(stim, threshold, slope, guess, lapse)
}
```

The *func_resp0* and *func_resp1* are complementary, in other words, the probabilities they return add up to 1. Next, we need to specify the range of possible values for the stimulus and psychometric parameters. These parameters must be specified as an array, also when they are single values, for which *jsQuestPlus.linspace* and *jsQuestPlus.array* can be used:

```
// [-40, -39, -38, ..., -1, 0]
const contrast_samples = jsQuestPlus.linspace (-40, 0)
const threshold_samples = jsQuestPlus.linspace (-40, 0)
// [2, 3, 4, 5]
const slope_samples = jsQuestPlus.linspace (2, 5)
// [0, 0.01, 0.02, 0.03, 0.04]
const lapse_samples =jsQuestPlus.array (0, 0.01, 0.04)
// The parameter of guess is assumed as a single value.
const guess = [0.5]
```

Note that a larger number of samples will affect the execution time of the QUEST+ method. This will be discussed in more detail in the simulation section. After specifying the psychometric functions and possible parameters, initialize the QUEST+ object as follows:

```
const jsqp = new jsQuestPlus({
  psych_func: [func_resp0, func_resp1],
  stim_samples: [contrast_samples],
  psych_samples: [threshold_samples, slope_samples,
                  guess, lapse_samples]
})
```

Here, *jsqp* is an abbreviation of jsQuestPlus, but any valid JavaScript variable name could be used instead. The jsQuestPlus constructor should receive one argument, which is an object with three properties: *psych_func*, *stim_samples*, and *psych_samples*. Note that the elements in the *psych_samples* array (i.e., threshold, slope, guess, and lapse) must be written in the order specified in the psychometric function declaration. Although priors will be treated as a uniform probability over all psychometric parameter combinations by default, these can be specified individually. See the associated GitHub repositories for details.

After completing the initialization, the stimulus parameters that are predicted to yield the most informative results at the next trial can be obtained as follows:

```
stim = jsqp.getStimParams()
```

The *getStimParams* function returns the stimulus parameter(s) that minimize(s) the expected entropies of the PDF of the psychometric parameters. The QUEST+ method recommends presenting the stimulus with the returned parameters and obtaining the response. In the example task, the response is 0 or 1. This response should match the index of the corresponding psychometric function in the array passed to the jsQuestPlus constructor. If a correct response (response = 1) is obtained, update the PDF and the expected entropies as follows:

```
jsqp.update(stim, 1)
```

The presentation of stimuli, obtaining the responses, and updating of the data are repeated a predetermined number of times. Finally, the psychometric parameter estimates with the highest posterior probability can be obtained as follows:

```
const estimates = jsqp.getEstimates()
```

The *estimates* array includes the estimates of each psychometric parameter, that is, the threshold, slope, and lapse in this example.

Simulation

The computational complexity of the QUEST+ method has been found to increase in proportion to the number of stimulus parameters, the number of psychometric parameters, the number of response options, and the number of samples in each parameter (Watson, 2017). Watson conducted a simulation that measured the computation time for various numbers of samples and reported that it does not exceed one second per trial even under extremely high computational complexity. In laboratory-based experiments, the QUEST+ method is considered practical because a single computer with high computing power is often used. Web-based experiments could require longer computation times because (a) participants' computers have more variation in processing power than in a lab, and (b) in contrast to MATLAB or Mathematica, JavaScript is not optimized for matrix computations. Hence, it is important to ensure our library is tested for performance: does it deliver psychometric parameters fast enough? Hence, we examined the performance of jsQuestPlus as a function of computational complexity across a range of commodity devices and browsers.

Methods

For our performance test, we selected four conditions from Watson's (2017) simulations, corresponding to Examples 1, 2, 4, and 5 in the paper, respectively. The number of samples was calculated by multiplying the number of possible values in each parameter. For example, in Example 2 of Watson (2017), the number of samples is 41 (-40, -39, -38, …, 0) for the stimulus parameter, is 41 (-40, -39, -38, …, 0) for threshold, is 4 (2, 3, 4, 5) for slope, is 5 (0, .01, .02, .03, .04) for lapse, and is 2 for response; the total number of samples is 67240 by multiplying all the sample numbers. See also the complexity and timing section of Watson (2017). Our four conditions are summarized in Table 1.

The simulation program was written using jsPsych (de Leeuw, 2015) and jsQuestPlus. Following Watson's (2017) examples, the four conditions were repeated 32, 64, 32, and 64 times, respectively. We measured (a) the time it took to initialize jsQuestPlus, (b) the time it took to determine the stimulus parameters, specifically the time to call the *getStimParams* function, and (c) the time it took to call the update function. Each duration was measured using the *performance.now* function, which provides a timestamp with microsecond precision. Duration (a) was obtained once for each browser–computer combination, while durations (b) and (c) were averaged over trials since they were measured repeatedly for each browser–computer combination.

We asked the members of the authors' research groups to access the simulation program via the Internet. Participants could run the program as many times as they wished under the condition that such participations used different web browsers on the same device or used a different device. Both operating system (OS) and web browser information were obtained using platform.js (Dalton & Tan, 2020). In total, we obtained simulation data of 37 browser–computer combinations. OS and browser are summarized in Tables 2 and 3.

One might be interested in the result of computers with low performance. Although platform.js could not basically collect detailed information such as model number, we confirmed from the user agent information that SONY 801SO, SHARP SHV47, and HUAWEI RNE-L22 were included. These are smartphones—a device type of which recent studies suggest that they may be a suitable medium for administering cognitive tasks (Pronk et al., 2022). As for the RNE-L22, it was a low-spec model released several years ago (CPU: 2.36 GHz 4 core & 1.7 GHz 4 core; RAM: 4 GB). Moreover, detailed OS numbers are also recorded for iOS and Mac OS. We found a Mac using macOS High Sierra (10.13.6), which was released in 2017. The computation time on these computers can be used as a reference when conducting experiments using equipment with low computational power. All the user agent information is available at OSF (https://osf.io/tqesb/).

Table 1 Number of parameters, number of responses, and number of samples in the four simulated conditions

Condition	Number of stimulus parameters	Number of psychometric parameters	Number of responses	Total number of samples
1	1	1	2	3,362
2	1	3	2	67,240
3	2	3	2	660,660
4	3	4	2	911,250

Table 2 Number of operating systems tested

Operating system	Number
Android	4
iOS	6
Mac OS X	6
Windows	21

Table 3 Number of web browsers tested

Web browser	Number
Chrome	12
Chrome Mobile	4
Firefox	6
Firefox Mobile	1
Microsoft Edge	7
Safari	7

Results

The time required to initialize jsQuestPlus, determine the stimulus parameters, and update the data are summarized in Table 4. Values in brackets represent 95% confidence intervals assuming a *t* distribution ($df = 36$). While the time required for initialization and updating increased with the number of parameters, the time required to determine the stimulus parameters did not show such a trend, and was fast enough to be considered a negligible factor in an actual experiment. Although the time required for initialization was

Table 4 Computation times required to run jsQuestPlus in milliseconds. Confidence intervals (CIs) assume a *t* distribution (*df* = 36). The larger the condition number, the greater the computational load

Condition	Initialization		Determination of stimulus parameters		Update		Watson (2017)
	Mean	95% CI	Mean	95% CI	Mean	95% CI	
1	12.68	[8.49, 16.88]	0.07	[0.04, 0.09]	1.29	[0.96, 1.62]	4.4
2	47.38	[35.93, 58.82]	0.03	[0.02, 0.04]	12.25	[8.68, 15.81]	41
3	333.96	[239.35, 428.57]	0.05	[0.03, 0.07]	61.58	[39.04, 84.12]	200
4	461.08	[328.29, 593.87]	0.06	[0.03, 0.08]	79.39	[46.22, 112.55]	270

Watson (2017) reported the total time required to determine stimulus parameters and to update the data for the next trial

relatively long compared to the update times, it was less than one second. The time required for updating was, contrary to our concerns, much faster than that reported by Watson (2017). When using hardware with relatively low computational power, the computation time in the most demanding condition (condition 4) was 1713 ms (RNE-L22) and 353.3 ms (maOS High Sierra) for initialization, 0.12 ms (RNE-L22) and 0.08 ms (maOS High Sierra) for determination of stimulus parameters, and 198.3 ms (RNE-L22) and 38.9 ms (maOS High Sierra) for updating. Histograms for computation time in the most demanding condition is shown in Fig. 1. Histograms for all the conditions are available at OSF (https://osf.io/tqesb/).

A reviewer suggested presenting not only the time data but also the validation data of jsQuestPlus. The simulation program described above measured time as well as estimated psychometric parameters. Table 5 summarized the estimates and 95% confidence intervals (CIs) of the psychometric parameters for each condition. The 95% CIs include simulated values except for the slope and lapse parameters in the condition 2.

Discussion

This study presented a Bayesian adaptive psychometric method for online experiments named jsQuestPlus. It works in combination with existing online experimental tools such as jsPsych (de Leeuw, 2015), PsychoJS (Peirce et al., 2019), and lab.js (Henninger et al., 2021), and should work with other experimental tools like OpenSesame/OSWeb (Mathôt et al., 2012) and Gorilla (Anwyl-Irvine et al., 2020).

Our simulation showed that computation times were short enough for most online psychophysical experiments. With a large number of samples, initialization could be relatively slow. However, initialization is only required at the beginning of a series of trials, so it is unlikely that the initialization time will cause problems in conducting an experiment. The time for updating the data tends to become longer as the total number of samples increases and may take up to 100 ms on average, so the function should be called during a less time-critical phase of a task, such as after the end of the previous trial. The execution time of the function to determine the stimulus intensity (*getStimParams*) was less than 1 ms, so calling it immediately before the stimulus is presented should not be a problem. If the execution time would ever become a concern, execution of *getStimParams* could be relegated to a less time-critical phase of a task, similar to what we recommend for updating the data. A more flexible solution would be the incorporation of Web Workers (Mozilla, 2022), so that jsQuestPlus calculations are executed as a background process that is less likely to interfere with the task procedure.

The jsQuestPlus library could accurately estimate psychometric parameters except for the slope and lapse parameters in the condition 2. When inaccurate, the biases displayed by jsQuestPlus were similar to those observed in QUEST+ by Watson (2017). In other words, we explain the biases we observed in jsQuestPlus as being endemic to the QUEST+ method. Solving this problem is an interesting topic, but beyond the scope of our study. Regardless, the parameter that tends to be of the greatest interest to psychometric models, namely threshold, is being estimated very accurately.

In laboratory-based experiments, the QUEST method has been often used to modulate the contrast of a grating (e.g., Keefe et al., 2021; Kim & Chong, 2021; Yu & Postle, 2021). Such procedures require calibrating the monitor brightness. In online experiments, it is very difficult to calibrate the monitors used by participants. Moreover, the resolution of a standard

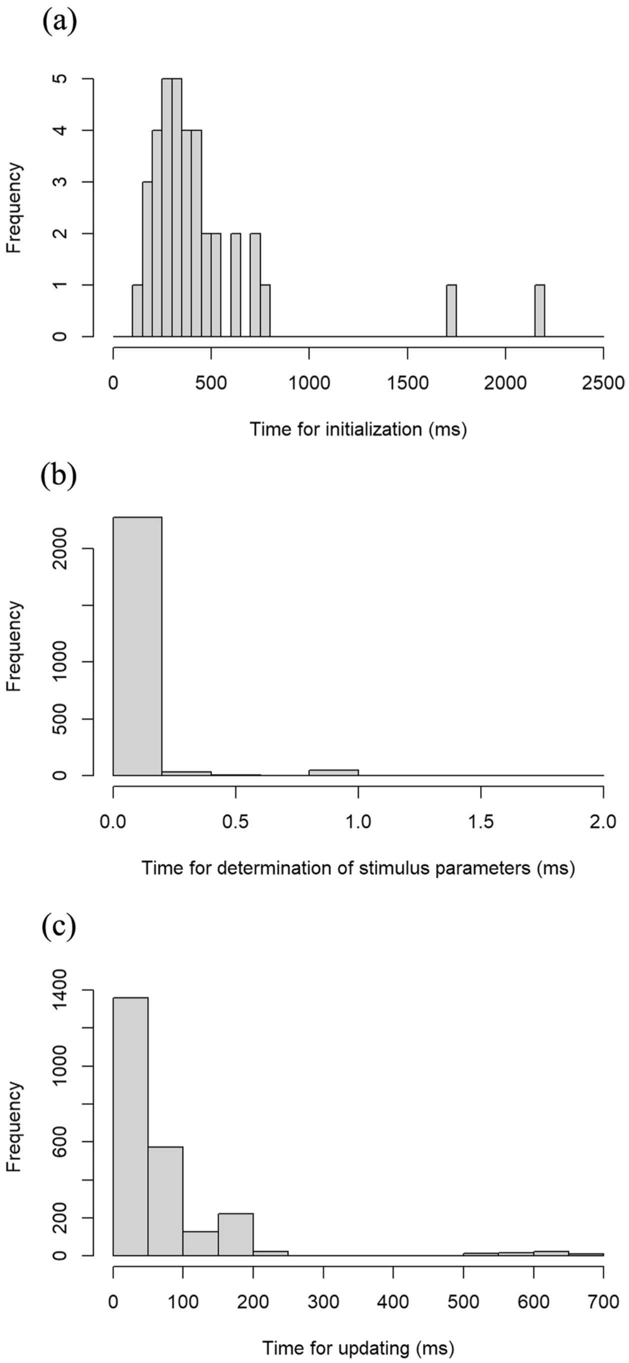

Fig. 1 Histograms for computation time in the most demanding condition (condition 4). **a** Time for initialization. **b** Time for determination of stimulus parameters. **c** Time for updating. The respective bin sizes are (**a**) 50 ms, (**b**) 0.2 ms, and (**c**) 50 ms

8-bit display (256 discrete levels of brightness) might be too small for some of the psychophysical experiments. A limited solution for these problems is to restrict the model of the devices and to allow participants to participate in the experiment if they can prepare their devices before taking part.

On the other hand, recent laboratory-based experiments have used the QUEST method to manipulate the number of random dots (Kurki, 2019), the motion direction of random dots (Song et al., 2021), and the size of an aperture (Luzardo & Yeshurun, 2021). These experiments should be suitable for online administration, especially now that technology for online random dot kinematograms (Rajananda et al., 2018), virtual chinrests (Li et al., 2020), and a jsPsych plugin for psychophysics (Kuroki, 2021) are available. Moreover, it is noteworthy that Myrodia et al. (2021) showed that there was no difference in perceptual thresholds of the perceived quality of computer-generated images between laboratory-based and online experiments using the QUEST+ method. Future research could investigate whether well-known results of laboratory-based experiments using the QUEST/QUEST+ method can be replicated online.

The accuracy of the QUEST+ method, both online and in the lab, can be affected by lapses in concentration. Jones (2019) reviewed several approaches for taking lapses into account and proposed to weigh participants' responses by the probability that a lapse occurred. For detecting whether a lapse occurred, they suggest using eye, head, or upper body movements, in combination with response latency or consistency. Such measures can also be acquired online since modern web browsers offer access to a wide range of sensors. For instance, online head- and eye-tracking can be performed via WebGazer (Papoutsaki et al., 2016) and mouse-tracking via MouseView (Anwyl-Irvine et al., 2021).

One useful feature that jsQuestPlus does not provide, but that QUEST+ based on Mathematica and MATLAB do, is fitting which is performed post hoc, and enables to estimate the psychometric parameters with a high degree of precision and range. As illustrated by Manning et al. (2018), more finely grained estimates can be closer to the true threshold, especially when analyzing data of children with attentional lapses. Although the fitting function is not included in jsQuestPlus, it can be performed afterwards using Mathematica or MATLAB. See the associated GitHub repositories for details.

While there are some limitations to conducting psychophysical experiments online, there are many advantages as well. For example, researchers can efficiently recruit a diverse group of participants and data collection at home can be conducted regardless of lock-down measures (as have been issued lately

Table 5 Simulated values and values estimated by jsQuestPlus. Confidence intervals (CIs) assume a *t* distribution (*df* = 36). For more information on the simulated conditions, see Watson (2017)

Condition	Psychometric parameter	Simulated value	Estimate	95% CI	Watson (2017)
1	Threshold	-20	-19.97	[-20.75, -19.19]	-20
2	Threshold	-20	-19.78	[-20.10, -19.47]	-20
	Slope	3	3.92	[3.47, 4.37]	5
	Lapse	0.02	0.0073	[0.0031, 0.0115]	0.04
3	Minimum threshold (t)	-35	-34.70	[-36.07, -33.33]	-32
	Coefficient (c0)	-50	-49.84	[-51.59, -48.08]	-56
	Coefficient (cf)	1.2	1.19	[1.14, 1.24]	1.4
4	Minimum threshold (t)	-40	-40.95	[-42.05, -39.85]	-35
	Coefficient (c0)	-50	-50.68	[-52.05, -49.31]	-50
	Coefficient (cf)	1.2	1.22	[1.17, 1.26]	1.2
	Coefficient (cw)	1.0	1.01	[0.97, 1.05]	1.0

in response to COVID-19). In addition, online experiments embrace open science values, because, in principle, anyone can replicate procedures without needing to purchase any software licenses or specialized hardware. We hope that the tools introduced here will further increase the variety of experiments that can be conducted online. Both the jsQUEST (https://github.com/kurokida/jsQUEST) and jsQuestPlus (https://github.com/kurokida/jsQuestPlus) libraries are available under the MIT license on the GitHub registry, where they can be downloaded, forked, discussed, and improved.

References

Anwyl-Irvine, A. L., Massonnié, J., Flitton, A., Kirkham, N., & Evershed, J. K. (2020). Gorilla in our midst: An online behavioral experiment builder. *Behavior Research Methods, 52*(1), 388–407. https://doi.org/10.3758/s13428-019-01237-x

Anwyl-Irvine, A. L., Armstrong, T., & Dalmaijer, E. S. (2021). MouseView.js: Reliable and valid attention tracking in web-based experiments using a cursor-directed aperture. *Behavior Research Methods*. https://doi.org/10.3758/s13428-021-01703-5

Brainard, D. H. (1997). The Psychophysics Toolbox. *Spatial Vision, 10*(4), 433–436. https://doi.org/10.1163/156856897X00357

Brainard, D. H. (2017). *mQUESTPlus: MATLAB implementation of Watson's Quest+*. https://github.com/BrainardLab/mQUESTPlus

Bunce, C., Gray, K. L. H., & Cook, R. (2021). The perception of interpersonal distance is distorted by the Müller-Lyer illusion. *Scientific Reports, 11*, Article 494. https://doi.org/10.1038/s41598-020-80073-y

Dalton, J.-D., & Tan, B. (2020). *platform.js: A platform detection library (Version 1.3.6)*. [Computer software] https://github.com/bestiejs/platform.js/

de Leeuw, J. R. (2015). jsPsych: A JavaScript library for creating behavioral experiments in a Web browser. *Behavior Research Methods, 47*(1), 1–12. https://doi.org/10.3758/s13428-014-0458-y

Hadrien, J., & Jaquiery, M. (2016). *StaircaseJS: Adaptive staircase procedure in JavaScript*. https://github.com/hadrienj/StaircaseJS

Henninger, F., Shevchenko, Y., Mertens, U. K., Kieslich, P. J., & Hilbig, B. E. (2021). lab.js: A free, open, online study builder. *Behavior Research Methods*. https://doi.org/10.3758/s13428-019-01283-5

Hirst, R. J. (2020). *Basic JND orientation discrimination demo*. https://gitlab.pavlovia.org/lpxrh6/staircase-demo

Höchenberger, R. (2019). *questplus: A QUEST+ implementation in Python*. https://github.com/hoechenberger/questplus

Jones, P. R. (2018). QuestPlus: A Matlab implementation of the QUEST+ adaptive psychometric method. *Journal of Open Research Software, 6*(1), Article 27. https://doi.org/10.5334/jors.195

Jones, P. R. (2019). Sit still and pay attention: Using the Wii Balance-Board to detect lapses in concentration in children during psychophysical testing. *Behavior Research Methods, 51*(1), 28–39. https://doi.org/10.3758/s13428-018-1045-4

Kawabe, T. (2021). Perceptual properties of the Poisson effect. *Frontiers in Psychology, 11*, Article 612368. https://doi.org/10.3389/fpsyg.2020.612368

Keefe, J. M., Pokta, E., & Störmer, V. S. (2021). Cross-modal orienting of exogenous attention results in visual-cortical facilitation, not suppression. *Scientific Reports, 11*, Article 10237. https://doi.org/10.1038/s41598-021-89654-x

Kim, C., & Chong, S. C. (2021). Partial awareness can be induced by independent cognitive access to different spatial frequencies. *Cognition, 212*, Article 104692. https://doi.org/10.1016/J.COGNITION.2021.104692

King-Smith, P. E., Grigsby, S. S., Vingrys, A. J., Benes, S. C., & Supowit, A. (1994). Efficient and unbiased modifications of the QUEST threshold method: Theory, simulations, experimental evaluation and practical implementation. *Vision Research, 34*(7), 885–912. https://doi.org/10.1016/0042-6989(94)90039-6

Kontsevich, L. L., & Tyler, C. W. (1999). Bayesian adaptive estimation of psychometric slope and threshold. *Vision Research, 39*(16), 2729–2737. https://doi.org/10.1016/S0042-6989(98)00285-5

Kurki, I. (2019). Stimulus information supporting bilateral symmetry perception. *Vision Research, 161*, 18–24. https://doi.org/10.1016/J.VISRES.2019.02.017

Kuroki, D. (2021). A new jsPsych plugin for psychophysics, providing accurate display duration and stimulus onset asynchrony. *Behavior Research Methods, 53*, 301–310. https://doi.org/10.3758/s13428-020-01445-w

Kuroki, D., & Pronk, T. (2021a). *jsQUEST: A Bayesian adaptive psychometric method for measuring thresholds in online experiments*. https://github.com/kurokida/jsQUEST

Kuroki, D., & Pronk, T. (2021b). *jsQuestPlus: A JavaScript library to use the QUEST+ method in online experiments*. https://github.com/kurokida/jsQuestPlus

Leek, M. R. (2001). Adaptive procedures in psychophysical research. *Perception & Psychophysics, 63*(8), 1279–1292. https://doi.org/10.3758/BF03194543

Levinson, M., Podvalny, E., Baete, S. H., & He, B. J. (2021). Cortical and subcortical signatures of conscious object recognition. *Nature Communications, 12*, Article 2930. https://doi.org/10.1038/s41467-021-23266-x

Levitt, H. (1971). Transformed up-down methods in psychoacoustics. *The Journal of the Acoustical Society of America, 49*(2B), 467–477. https://doi.org/10.1121/1.1912375

Li, Q., Joo, S. J., Yeatman, J. D., & Reinecke, K. (2020). Controlling for participants' viewing distance in large-scale, psychophysical online experiments using a virtual chinrest. *Scientific Reports, 10*, Article 904. https://doi.org/10.1038/s41598-019-57204-1

Luzardo, F., & Yeshurun, Y. (2021). Inter-individual variations in internal noise predict the effects of spatial attention. *Cognition, 217*, Article 104888. https://doi.org/10.1016/j.cognition.2021.104888

Manning, C., Jones, P. R., Dekker, T. M., & Pellicano, E. (2018). Psychophysics with children: Investigating the effects of attentional lapses on threshold estimates. *Attention, Perception, & Psychophysics, 80*(5), 1311–1324. https://doi.org/10.3758/s13414-018-1510-2

Mathôt, S., Schreij, D., & Theeuwes, J. (2012). OpenSesame: An open-source, graphical experiment builder for the social sciences. *Behavior Research Methods, 44*(2), 314–324. https://doi.org/10.3758/s13428-011-0168-7

Mozilla. (2022). *Using Web Workers*. https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API/Using_web_workers

Myrodia, V., Buisine, J., & Madelain, L. (2021). Comparison of threshold measurements in laboratory and online studies using a Quest+ algorithm. *Journal of Vision, 21*(9), Article 1959. https://doi.org/10.1167/jov.21.9.1959

Papoutsaki, A., Sangkloy, P., Laskey, J., Daskalova, N., Huang, J., & Hays, J. (2016). WebGazer: Scalable webcam eye tracking using user interactions. *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 3839–3845.

Peirce, J. W. (2007). PsychoPy-Psychophysics software in Python. *Journal of Neuroscience Methods, 162*(1–2), 8–13. https://doi.org/10.1016/j.jneumeth.2006.11.017

Peirce, J. W., Gray, J. R., Simpson, S., MacAskill, M., Höchenberger, R., Sogo, H., Kastman, E., & Lindeløv, J. K. (2019). PsychoPy2: Experiments in behavior made easy. *Behavior Research Methods, 51*(1), 195–203. https://doi.org/10.3758/s13428-018-01193-y

Pelli, D. G. (1996). *QuestDemo*. https://github.com/Psychtoolbox-3/Psychtoolbox-3/blob/master/Psychtoolbox/Quest/QuestDemo.m

Pelli, D. G. (1997). The VideoToolbox software for visual psychophysics: transforming numbers into movies. *Spatial Vision, 10*(4), 437–442. https://doi.org/10.1163/156856897X00366

Pronk, T., Wiers, R. W., Molenkamp, B., & Murre, J. (2020). Mental chronometry in the pocket? Timing accuracy of web applications on touchscreen and keyboard devices. *Behavior Research Methods, 52*(3), 1371–1382. https://doi.org/10.3758/s13428-019-01321-2

Pronk, T., Hirst, R. J., Wiers, R. W., & Murre, J. M. J. (2022). Can we measure individual differences in cognitive measures reliably via smartphones? A comparison of the flanker effect across device types and samples. *Behavior Research Methods*. https://doi.org/10.3758/S13428-022-01885-6

Rajananda, S., Lau, H., & Odegaard, B. (2018). A random-dot kinematogram for web-based vision research. *Journal of Open Research Software, 6*(1), Article 6. https://doi.org/10.5334/jors.194

Reips, U.-D. (2021). Web-based research in psychology. *Zeitschrift für Psychologie, 229*(4), 198–213. https://doi.org/10.1027/2151-2604/a000475

Santacroce, L. A., Carlos, B. J., Petro, N., & Tamber-Rosenau, B. J. (2021). Nontarget emotional stimuli must be highly conspicuous to modulate the attentional blink. *Attention, Perception, & Psychophysics*. https://doi.org/10.3758/s13414-021-02260-x

Sasaki, K., & Yamada, Y. (2019). Crowdsourcing visual perception experiments : a case of contrast threshold. *PeerJ, 7*, Article e8339. https://doi.org/10.7717/peerj.8339

Song, Y., Chen, N., & Fang, F. (2021). Effects of daily training amount on visual motion perceptual learning. *Journal of Vision, 21*(4), Article 6. https://doi.org/10.1167/jov.21.4.6

Straw, A. D. (2008). Vision Egg: An open-source library for real-time visual stimulus generation. *Frontiers. Neuroinformatics, 2*, Article 4. https://doi.org/10.3389/neuro.11.004.2008

Treutwein, B. (1995). Adaptive psychophysical procedures. *Vision Research, 35*(17), 2503–2522. https://doi.org/10.1016/0042-6989(95)00016-X

Watson, A. B. (2017). QUEST+: A general multidimensional Bayesian adaptive psychometric method. *Journal of Vision, 17*(3), 1–27. https://doi.org/10.1167/17.3.10

Watson, A. B., & Pelli, D. G. (1983). Quest: A Bayesian adaptive psychometric method. *Perception & Psychophysics, 33*(2), 113–120. https://doi.org/10.3758/BF03202828

Yu, Q., & Postle, B. R. (2021). The neural codes underlying internally generated representations in visual working memory. *Journal of Cognitive Neuroscience, 33*(6), 1142–1157. https://doi.org/10.1162/jocn_a_01702

Open practices statement The data and materials for all experiments are available at Open Science Framework (https://osf.io/tqesb/), and none of the experiments was preregistered.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.