

# FlowStims v1.0 documentation

---

- Basic operation
- Parameters
  - Setting parameter values
  - Multiple values
  - Dependencies
  - Total number of trials
  - Randomization of trials
  - Performance issues
  - List of parameters
- Log files
- Network communication

## Basic operation

FlowStims uses a .txt file to specify which stimuli are to be used, how to display them, and, optionally, to set up communication with other machines via UDP.

An example of such a file is `example-params.txt`. Any text following a # character is treated as a comment, and ignored by FlowStims.

Typo-checking is very limited, so be careful when entering values. Unforseen characters might cause the program to crash upon startup. If that happens, please double check every entered value. A good idea is to use `example-params.txt` as a starting point.

When run, FlowStims will prompt the user for a parameters file. Upon selection of a suitable file, the presentation begins. The program will terminate automatically once all trials are over. The user can also hit the escape key (Esc) to quit it early. All trial information is saved in a .log file (see below).

## Parameters

### Setting parameter values

Each parameter occupies a separate line in the file. The format for setting values is "parameter value", that is, the parameter name and its associated value are separated by a space.

### Multiple values

Some parameters are special in that they allow for the input of multiple values. E.g., you can choose two different temporal frequencies for your stimuli. This will result in interleaved trials with either temporal frequency. To do so, simply separate the additional values with a space.

## Dependencies

An arrow `->` was used in `example-params.txt` before all parameters allowing multiple values whose total number of values must be equal to that of some other parameter. This means it is dependent on the preceding parameter that does not begin with an arrow (the "parent" parameter). E.g., for each dot intensity (grayscale level) you choose, you must also specify intensities for the background and interstimulus screen. Failing to meet this constraint will cause the program to crash.

You can choose to write the parameter names without the arrows, or leave them as a reminder.

## Total number of trials

Make sure to keep track of the total number of trials given your parameter selection. E.g., if you have 4 stimulus variations and 8 directions of motion and you show each one a single time, that is a total of  $4 \times 8 = 32$  trials, or 1 trial block. The number of trial blocks (ie, how many times each stimulus variation in every direction will be shown) is controlled by `nTrialBlocks` and will essentially determine the total presentation time (multiply if by the total trial length to get the actual time in seconds).

## Randomization of trials

The order of the trials is automatically randomized. In order to ensure that the different stimulus variations are properly mixed, every stimulus variation (except for direction of motion) is shown before one repeats. This is particularly important if there is a significant difference in overall screen luminance between some of the stimuli, in which case there is no chance for several stimuli with similar luminance to be shown in sequence (therefore preventing undesired adaptation effects).

## Performance issues

In most cases, FlowStims can be used to generate stimuli on the fly during a recording. However, depending on your hardware and on the parameters chosen, some stimuli might take longer to run than expected (in which case you might end up with longer trial lengths), especially if you choose high quality graphics and/or very high spatial frequency stimuli. You can easily check the exact duration of your trials by inspecting trial log file that is generated after each run.

One option around this is to enable `fastRendering` and/or disable `antiAlias`. Another is to decrease the `frameRate` at which the stimuli are displayed. Yet another is to save the presentation as a movie (see `saveMovieFrames`) and simply play it during the experiment (that is, without running FlowStims). (The latter will probably require some software to play the movie while logging the times of each frame, which is not currently provided.)

## List of parameters

### Version info

The first line of the file must contain the word `FlowStims` followed by the version number. This is used to prevent parameter files from being used by a different version than the original one they were written for.

### Setup

These include parameters for setting up the display, resolution, quality of the graphics, etc.

`scrWidthPx` Screen width, in pixels. E.g., `800`

`scrWidthCm` Physical width of screen (cm). E.g., `24.5`

`scrDistCm` Distance of the animal from screen (cm).

`monitor` Selects the screen on which to display to program (default is 1 for your primary monitor).

`fullScreen` Set it to 1 for fullscreen mode (default). If set to `0`, disables fullscreen mode (for running it in a smaller window, usually for debugging purposes).

`fastRendering` Default is 1. Disable this (set to `0`) for improved graphics (can be slow in some machines).

`antiAlias` Turn this on (set to 1) for smoother graphics (can be slow if using high spat freqs). Default is 1.

`frameRate` Number of frames displayed every second (default is `60` fps).

`saveMovieFrames` Set it to 1 to save a screenshot (.png) of every frame, which are saved into a newly-created "movieframes" folder. These can be used for creating a movie (if you install Processing, it comes with a Movie Maker tool for doing just that). Note: the total file size of the frames can grow really fast, so be careful! For safety, `nTrialBlocks` is set to 1 when using this option (as usually this will be used to generate a demo movie, or a movie to be looped over when used in an experiment).

`saveTrialScrShots` Set it to 1 to save a single screenshot at the end of each trial (default is `0`). Note: there can be a slight delay when saving screenshots, so usually it is not a good idea to have this option turned on during an actual experiment.

`randomSeed` Pseudorandom number generator initial state. Choose any positive integer (e.g., 34) to make the presentation reproducible (the exact same randomization will take place every time you run the program using the same params file). To use a random initial state, use `-1` (default).

### All stimuli

These include parameters that are common to all stimulus variations used.

`trialLenSec` Trial length (stimulus presentation), in seconds. E.g., `1.5`

**preStimLenSec** Pre-stimulus interval (for best results with flow stimuli, this should usually be at least 1 sec)

**postStimLenSec** Post-stimulus interval. It is especially useful when using stimuli with different interstimulus screen intensities; this way the transition between stimulus and interstim screen before and after the trial will be symmetric. Note: Total time spent per trial is given by  $\text{preStimLenSec} + \text{trialLenSec} + \text{postStimLenSec}$ .

**nTrialBlocks** Number of blocks of trials (i.e., repetitions of each combination of parameters). See **Total number of trials** above for more details.

**nDirs** Number of principal directions of motion (evenly divided around 360 degrees). E.g., choosing 4 will result in the set [0, 90, 180, 270].

**dirDegShift** Amount by which to shift the directions that were specified by **nDirs**. E.g., setting it to 90 when **nDirs** is set to 2 will change [0, 180] into [90, 270] (an option that is not available when using **nDirs** alone!).

**tempFreq** Temporal frequency (cycles/s). Can choose multiple. E.g., 2 for a single value, or e.g. 2.5 4 for two values, etc.

**nFadeFrames** Sets the number of frames used for linear fade-in/out transitions at the beginning and end of trials (use 0 or 1 for no fade).

## Flow stimuli

**useFlows** Whether to display flows or not (1 or 0).

**nDots** Number of dots used in the flow elements, can choose multiple. E.g., 1 3 for showing interleaved 1- and 3-dot flows.

---

**dotFgVal** Pixel value (grayscale) for foreground (dots) [0-255], can choose multiple.

→**dotBgVal** Pixel value (grayscale) for background [0-255]. Must be given the same number of values as **dotFgVal**.

→**dotInterVal** Pixel value (grayscale) of interstimulus screen (use -1 for avg screen luminance of the stimulus). Must be given the same number of values as **dotFgVal**.

---

**dotDiamDeg** Diameter of single dots (in degrees); allows multiple values. Note: flows with **nDots** other than 1 might have dots with different diameter in order to preserve constant area (depending on whether **equalArea** is enabled, see below).

→**dotSpacing** Initial spacing between flow element centers (in multiples of **dotDiamDeg**); this determines how dense the flow will be. Of course, if the motion is not rigid, then this spacing ends up being an average. Must get a value for each **dotDiamDeg** used above.

→**dotSpatFreq** *Optional*: If known, the spatial frequencies corresponding to each dot size used in **dotDiamDeg** can be entered here. When provided, they will be used to compute more accurate velocities based on the temporal frequency. Must get a value for each **dotDiamDeg** used above.

---

**equalArea** Whether to adjust the diameter of patterns with `nDots > 1` so as to preserve the same total area as that of single dots. Defaults to yes (1).

**rigidTrans** Set it to 1 for a rigid translation of the flow elements, i.e., no "jitter" -- no motion components other than main direction of motion (a constant parallel flow field and no separation force during trials).

**sepWeight** Ratio between repulsion force between elements and the force from the flow field -- should be at least around 1.5 for best results, and might need some adjustment for different dot sizes. Default: 1.5

**dirStd** Std. dev. of the direction distribution of the underlying flow field. Default: 0.09

**posStd** Std. dev. of initial position of dots on screen, as a fraction of `dotSpacing`. Default: 0.1

**fixRandState** Set it to 1 to make identical all trials of the same flow variation (fixes the initial dot positions and flow field). Default is 0 (each single trial has a random initial configuration).

**maxForce** *Advanced:* Adjusts how responsive are the dots to the forces they are subject to. Usually this should be left unchanged, unless perhaps when using different frame rates or extreme dot sizes or spacings. Default: 0.04

**tileSize** *Advanced:* Tile size of the underlying vector field, in multiples of the `dotSpacing`. Larger tiles will result in more flow elements following the same direction locally. Default: 2.5

## Grating stimuli

**useGratings** Whether to display square-wave gratings or not (1 or 0).

**gratWidthDeg** Width of grating bars, in degrees of visual angle, i.e.  $1/(2 * \text{spat.freq.})$ .

**randGratPhase** Set it to 1 to enable random initial phase in each trial; default is 0 (fixed initial phase)

---

**gratFgVal** Pixel value (grayscale) of half the bars ("foreground") [0-255], can choose multiple

→ **gratBgVal** Pixel value (grayscale) of the other half of the bars ("background"), [0-255]. Must be given the same number of values as **gratFgVal**.

→ **gratInterVal** Pixel value (grayscale) of interstimulus screen, [0-255] (use -1 for avg screen luminance of the stimulus). Must be given the same number of values as **gratFgVal**.

## Log files

Every time FlowStims is successfully run, it will generate two log files:

DATE\_TIME\_params.log and DATE\_TIME\_trials.log, where DATE and TIME stand for the actual date and time at the time of execution. The first one is a record of all parameters used. This is helpful in cases where the original parameters file is lost, or modified. The second one is organized in space-separated columns and contains information about each trial and interstimulus interval (call these "periods") that were shown. The first line is a header row,

identifying each column:

`Frame` is the frame number where the current period started

`Time` is the time since the beginning of the presentation, in milliseconds

`Period` is a descriptor of the current period: prestimulus interval (PRESTIM), trial (TRIAL), or poststimulus interval (POSTSTIM)

`TrialNo` is a counter for the total number of individual trials so far (not used for interstim periods)

`Stimulus` contains info about the stimulus used in the current trial (not used for interstim periods), in the format `property=value`. The properties `stim`, `dir`, and `tfreq` are always present, and identify the stimulus class, direction of motion, and temporal frequency for that trial. Flow stimuli have the following specific properties: `nDots`, `diam`, `dotLv1`, `bgLv1`, and `interLv1` (see above for parameter descriptions); grating stimuli have the properties: `width`, `fgLv1`, `bgLv1`, `interLv1`, and `phase` (see above for parameter descriptions).

## Network communication

It is possible to set up FlowStims to send UDP datagrams whenever one of the events below is triggered (event name followed by description):

- `clientStart` Beginning of presentation (sent only once, at the very beginning of the movie)
- `clientEnd` End of presentation (including early exit with Esc key)
- `clientTrialStart` Start of each trial
- `clientTrialEnd` End of each trial
- `clientTimeStamp` Send a timestamp every X seconds

This can be done by including lines in the parameters file with the following format:

`<eventName> <host> <port> <msgType> <msg> <addNewLine> <encoding>`

The arguments: `host`, `port`, `msgType`, etc. must be configured by the user (do not change their order! -- the last two are optional):

- `host` Host address that will receive the packets (UDP datagrams)
- `port` Port that will be listening for the packets
- `msgType` Use 1 to send as string, 2 to send as integer (single byte), or 5 to send a string while appending the current date and time to the end of the msg
- `msg` Message to be sent
- `addNewLine` Either 0 or 1 -- whether to add a '\n' char at end of msg. Default is no (0)
- `encoding` Only applicable if sending a string. Default: UTF-8 (single byte chars)

Note: all network communication is automatically disabled if `saveMovieFrames` mode is on.

E.g.:

```
clientStart localhost 9090 1 1 #sends a 1 integer to localhost on port 9090 at the  
beginning of presentation
```

```
clientTrialStart 144.20.133.11 8999 2 0 #send a 0 integer to first host at beginning  
of every trial
```

```
clientEnd 144.20.133.57 8980 1 4 #send a '4' string to second host at beginning of  
every trial
```

For multiple hosts, repeat this in multiple lines changing host, port, etc. for each one (while using the same event name). Note: the option of multiple hosts is only available for the `clientStart` and `clientEnd` events.

E.g.:

```
clientStart 144.20.133.11 8999 2 0 #send a 0 integer to first host at the beginning  
of presentation
```

```
clientStart 144.20.133.57 8980 1 4 #send a '4' string to second host at the beginning  
of presentation
```