Fast-dm: A Free Program for Efficient Diffusion Model Analysis

Andreas Voss[1] and Jochen Voss[2]

Abstract

In the present paper a flexible and fast computer program, called *fast-dm,* for diffusion model data analysis is introduced. *Fast-dm* is free software and can be obtained from the authors of this paper. The program allows estimating all parameters of Ratcliff's (1978) diffusion model from the empiric response time distributions of any binary classification task. *Fast-dm* is easy to use: it reads input data from simple text files, while program settings are specified by commands in a control file. The program allows to estimate even complex, hierarchical models and to fix individual parameters to given values. Detailed directions for use of *fast-dm* are presented, as well as results from three short simulation studies exemplifying the utility of *fast-dm*.

Keywords: Diffusion Model, *fast-dm*, PDE Method, Reaction Time Modelling, Decision Making

1 Introduction

Stochastic diffusion models (Ratcliff, 1978) provide a framework to analyse data from fast binary decisions. Compared to earlier models, there are several advantages of this kind of data analysis: Firstly, both response times and error-rates are entered simultaneously in one analysis. This is a great improvement over traditional techniques that are only based on either response times or error rates and thus might lead to misleading results (*e.g.*, about task difficulty) in case of a speed-accuracy trade-off. For example, it might be that the participant's responses slow down not because a task is more difficult but because they set themselves a more conservative response criterion. This is a case where an analysis of mean response times would come to wrong conclusions while the effect (adaptation of response strategy) could still be mapped by a diffusion model analysis. Secondly, the diffusion model is well suited to reflect the structure of the given information. Results are not one-dimensional (*i.e.*, good performance *vs.* bad performance) but consist of several parameters which make it possible to draw detailed conclusions about how a task is performed. In other words, the diffusion model helps to explain *why* responses are fast or slow and *why* few or many errors are made. Often, psychological theories allow developing hypotheses from about one specific parameter of the diffusion model (Voss, Rothermund, & Voss, 2004). Therefore, the diffusion model is a powerful tool to test psychological theories. These structural results are possible because of a third advantage of the diffusion model: The degree of utilisation of input data is excellent, because not only RT means are used but the complete RT *distribution* is analysed.

On the other hand there are some problems connected with diffusion-models data analysis: Firstly, the computation is expensive. This problem has been recently addressed by Voss and Voss (2006). The authors introduced a new method (the "PDE-method") that allows fast computations even at high accuracy. The basic idea of the PDE method will be sketched below. A second problem of this kind of modelling is the lack of standard software for diffusion-model data analysis. So far, diffusion model analysis was restricted to researchers capable of developing their own solutions. Only recently a MATLAB based program to estimate the Parameters of Ratcliff's diffusion model has been introduced by Vandekerckhove and Tuerlinckx (2006). Another solution has been provided Wagenmakers, van der Maas, &

1 Institut für Psychologie, Universität Freiburg, D-79085 Freiburg, Germany

2 Mathematics Institute, University of Warwick, Coventry, CV4 7AL, United Kingdom

Grasman, (in press): Their EZ-diffusion procedure allows to calculate rough estimates for some model's parameter from RT-Means, standard deviations, and the percentage of errors. However, the EZ-model does not cover all parameters, and the utilization of data is not as good as in complete estimation procedures.

In this paper we present a user-friendly, flexible, and efficient program, *fast-dm*, which addresses both of the mentioned problems. *Fast-dm* uses the PDE method (Voss & Voss, 2006) to compute the predicted cumulative distribution function (CDF) and it uses the Kolmogorov-Smirnov statistic (Voss et al., 2004) for the optimization of parameters. The program, including the source code, is available from the *fast-dm* web-site *http://www.psychologie.uni-freiburg.de/Members/voss/fast-dm* . *Fast-dm* should be easily portable to most computer systems. It was tested on Linux and Microsoft Windows systems.

In the following sections, we give a short description Ratcliff's (1978) diffusion model and of the PDE method (Voss & Voss, 2006) used in our implementation. Subsequently, details of the *fast-dm* algorithm and directions of usage are presented, as well as results from simulation studies.

2 Description of Ratcliff's Diffusion Model

The diffusion model is a model well suited to analyse data from fast binary decisions. It is based on the assumption that information available to the participant is represented as a one-dimensional quantity. A 'counter' on this dimension changes as a function of time and the decisional process continues until the counter reaches either one of two decision thresholds. In the model the dynamic behaviour of the counter is described by a diffusion process. This process is driven by a systematic component and a random component: The systematic component causes a constant change over time, while the random component adds Gaussian noise to the process.

The diffusion process is terminated as soon as it leaves the interval delimited by the decision thresholds and the exit point from the interval represents the binary decision. In other words, a process termination at the lower threshold means that decision 'A' is reached while a termination at the upper threshold stands for the alternate decision 'B'. With a given set of parameters, the diffusion model allows to predict RT distributions for both decisions. For a situation in which one decision might be considered `correct' and the other one `wrong', the RT distributions for correct responses and errors, respectively, as well as the ratio of both distributions are thus predicted.

There are several parameters in the diffusion model (*e.g.*, Ratcliff & Rouder, 1998). In a diffusion model data analysis these parameters are estimated so that the fit between predicted and empiric RT-distributions is optimal. The result of this parameter estimation procedure can then be interpreted in terms of cognitive processes (Voss et al., 2006). We conclude this section by giving an account of these model parameters. The simple diffusion model consists of four parameters (threshold separation, starting point, drift rate, and response-time constant), which will be described first.

The first parameter is the *width of the interval* between decision thresholds (a). If a is small, RTs are fast and the random influence on the decision process is rather large, that is, many errors occur. On the other hand, large values of a result in fewer error responses and long response times. The second parameter is the strength (and direction) of the systematic influence on the diffusion process, called the *drift rate* (v). Large (absolute) values of the drift cause short process duration, both for correct and erroneous responses, and also small percentages of errors, whereas a drift near zero leads to longer process durations and to more similar probabilities of the two process outcomes. The third model parameter regards the *starting value* of the process (z). The closer the process starts to a threshold, the higher the probability that this threshold is reached due to the random fluctuations. Therefore, the RT-distribution of one threshold gets larger the nearer the process starts to it and corresponding exit times decrease. The opposite is true, obviously, for the opposite threshold: Here, the process terminates less frequently and exit times increase with increasing distance of starting point and threshold. The fourth parameter covers the duration of all extra-decisional parts of the response time. This so-call *response-time constant* (t_0) models the time used by processes of stimulus encoding and, in large parts, the execution of a response.

The full model takes into account that the processing of a task varies to a certain degree across trials of an experiment: *inter-trial variability* is modelled for the starting value (z), the drift rate (v), and the response-time constant (t_0). The starting value in each trial is no longer assumed to equal z, but instead follows a uniform distribution around z with the width s_z. Likewise, for the response-time constant a uniform distribution with width s_{t0} around t_0 is assumed. The drift rate is modelled as a normal distribution with mean v and standard deviation s_v. A more exhaustive description of the interpretation of the parameters in terms of cognitive processes might be found elsewhere (*e.g.*, Ratcliff & Tuerlinckx, 2002; Voss, et al., 2004).

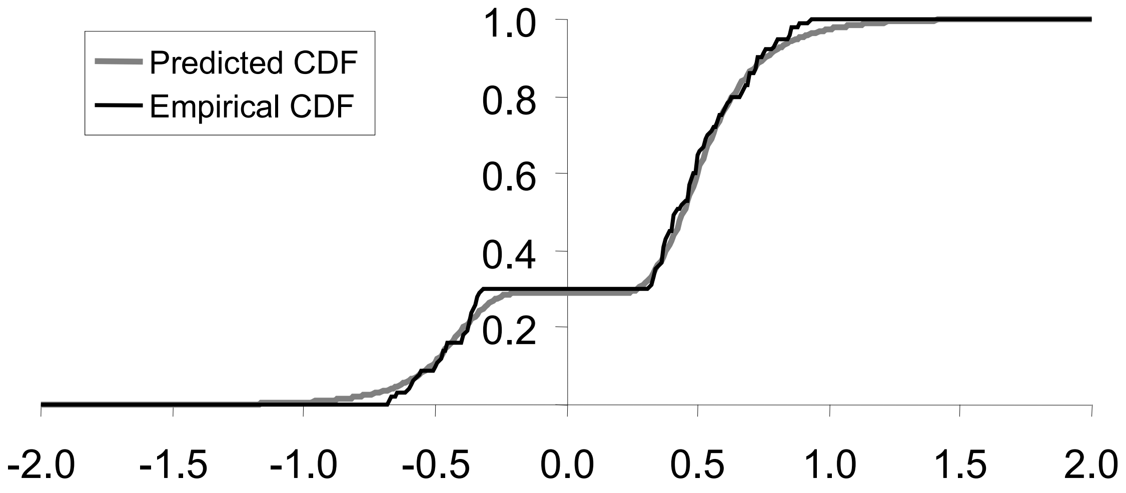

Figure 1. Example for a predicted and an empirical cumulative distribution function (CDF). *Fast-dm* merges the CDFs of correct responses and errors together by assigning negative signs to all error responses (or, more specific, to all responses at the lower threshold of the model). Therefore, the value of the CDF at the zero point of the abscissa shows the ratio of error responses. The given graph visualises a data set with about 30% error responses. The left side of the graph has not the typical shape of an RT distribution because here a mirror image of the original distribution is shown.

3 The PDE Method for Diffusion Model Analysis

As pointed out above, diffusion model analysis is expensive. To calculate one predicted RT-distribution from a set of parameters many operations have to be performed and, while searching the parameter space for an optimal set of parameters, many of these predicted RT-distributions have to be calculated. This can easily result in computation times of several hours or even days for a complete analysis. Therefore choice of an efficient algorithm to compute the required cumulative distribution functions (CDFs) is of great importance. Recently Voss and Voss (2006) suggested the PDE method to calculate the CDF for the RT-distribution in the diffusion model and showed that processing times were faster by several orders of magnitude, compared to the traditional closed form solution. In the following, we give a very short introduction to the PDE algorithm.

Mathematically, the CDF for a given set of parameters is found by solving a partial differential equation (PDE) associated with the diffusion process, namely the Kolmogorov backward equation. There is a well-known closed form solution for this problem that allows calculating the CDF as a function of the models parameters (Feller, 1971; cf. also Ratcliff, 1978; Voss et al. 2004). This formula, however, contains an infinite sum that—at least for certain parameter constellations—has to be evaluated many times before it converges satisfactorily. An alternative approach is to solve the PDE numerically (see Voss & Voss, 2006, for details) instead of resorting to the "closed" solution. This method, referred to as the PDE method here, avoids the problem of the infinite sum and—since numerical solutions to PDEs are mathematically well understood—proven standard methods can be employed.

The solution of the Kolmogorov backward equation is the probability that a Brownian Motion with constant drift v and start at z hits the boundary $a > z$ before time t and before the first visit at 0, considered as a function of the time t and the parameter z. For the numerical solution both the t and the z ranges are discretised and the solution is approximated by a function defined on the resulting grid. The algorithm starts at time 0 with a vector containing the initial condition for the different z-values on the grid points and then, in each step, uses the approximation for a fixed time to calculate the approximated values at a (slightly) later time. Details of this procedure can be found elsewhere (*e.g.*, Morton & Mayers, 1994; Press, Teukolsky, & Vetterling, 1992, chapter 19). Smaller step sizes, in both dimensions, increase the accuracy of the solution but, at the same time, also increase the computational cost.

Besides avoiding the infinite sum of the closed-form solution, the PDE method has the additional advantage that CDFs are computed for the whole range of different starting points simultaneously. Consequently, the best starting point can be picked out directly, thereby reducing the search space by one dimension and increasing the speed of parameter estimation.

Table 1. Commands of the control file for a *fast-dm* experiment.

Command	Example	Description
`precision` *value*	precision 2.0	Defines the precision of calculation (optional command; default value is 3.0, minimum is 1.0).
`set` *parameter value*	set st0 0	A parameter is not estimated but fixed to the given value (optional command; default is no fixation).
`depends` *parameter condition*	depends v stimulus	For different experimental conditions, different values for the parameter are estimated (optional command; by default there are no dependencies).
`format` *RESPONSE TIME*	format * stim RESPONSE TIME	Defines the format each line of the data file(s) (required command).
load "*filename*"	load "*.dat"	The `load` command defines the names of the data files. Wildcards (*) can be used as templates (required command).
save "*filename*"	save "*.out"	The `save` command defines the names of individual output-files. Wildcards may also be used in the same way as in the `load` command (optional command)
log "*filename*"	log "protocol"	The `log` command works similar like the `save` command; however, here all results are saved within a common file (optional command)

4 Software Internals: The *fast-dm* Algorithm

The calculation of the predicted RT distributions in *fast-dm* uses the PDE method described above. The parameter estimation procedure of *fast-dm* is based on the Kolmogorov-Smirnov approach (Voss, et al., 2004). That is, the maximal vertical distance of the predicted and the empirical cumulative response-time distribution (*i.e.*, the KS test statistic) is minimised. To get a single optimisation criterion, the response time distributions from both thresholds are merged into one distribution by equipping all reaction times from the lower threshold with negative signs while using positive signs for the times from the upper threshold (*Figure 1*). The CDF of the resulting combined distribution conserves the original shapes of the individual distributions (as the shapes of the function to the left and to the right of time zero), the ratio of errors is represented as the value of the combined CDF at time zero.

For many psychological applications some parameters will depend on experimental conditions (*e.g.*, different drift rates are assumed for different stimuli) while the remaining parameters are assumed to be constant over conditions. In this case the data set is split according to the experimental conditions and CDFs are calculated and compared for each subset separately, resulting in a different *KS*-value for each condition. When different experimental conditions are used, *Fast-dm* transforms the individual KS-values into probabilities values (Conover, 1999; Press, et al., 1992) and uses the product of the *p*-values across conditions as the optimisation criterion.

To estimate the model parameters a multidimensional search is conducted, using a version of the simplex-downhill method (Nelder & Mead, 1965) to optimise the fit (*i.e.*, to maximise *p*). For the parameters a, v and t_0 the output of the EZ-diffusion model (Wagenmakers, et al., in press) is used to obtain initial values for the simplex search. For the three inter-trial variability parameters, an arbitrary value (0.20 in our current implementation) is used as the initial value. Since the simplex method has sometimes difficulties in high dimensional spaces, the simplex algorithm is consecutively run three times with increasingly strict stop criteria, starting each run with the best solution of the previous run. Our tests indicate that this procedure improves results notably.

In a diffusion model analysis there is always a speed-accuracy trade-off, that is, the more accurate the calculation, the slower the algorithm runs. There are several points in the program where decisions have to be made about the accuracy settings (*e.g.*, the increment for the calculation of the integrals, and for the discretisation parameters of the PDE method). The effects of these settings are not independent of each other: Increasing accuracy at a point in the algorithm that works already well might increase processing time without affecting accuracy because much larger inaccuracies might be caused by another part of the software. While developing *fast-dm,* a systematic search (re-using the simplex-downhill method) was performed to optimise the speed while yielding an *a priori* defined level of accuracy. This has been done for different accuracy levels. As a result, the accuracy of *fast-dm* can be controlled by a single precision parameter which internally maps to the optimised parameter values.

5 Using *fast-dm*

In this paper, we describe *fast-dm* version 23. The program, including the source-code[3], can be obtained from the *fast-dm* web-site *http://www.psychologie.uni-freiburg.de/Members/voss/fast-dm* . On this site future versions of *fast-dm,* and all instructions necessary to install and use these, will be published.

5.1 Installation

To run *fast-dm* on Linux or Unix systems download the compressed source-code file (*fast-dm-23.zip* or *fast-dm-23.tar.gz*, respectively; the contents of theses achieves are identical), save it on your computer and unpack the archive. In the resulting folder 'fast-dm-23' type the following commands:

```
./configure
make
```

The compiler will automatically build the program *fast-dm* for the parameter estimation.[4] If you want to run *fast-dm* on a Microsoft Windows system, you can either download the precompiled binaries (*fast-dm-23_bin.zip*) or, if you have a C-Compiler, you can also use the source code (see the README file in the source-code archive for some notes on this topic).

5.2 Experiment Control Files

An experiment control file is used to pass commands for a specific experiment to *fast-dm*. The control file specifies which data files are to be read and what kind of model should be fitted to the data. By convention control files use the file name extension ".ctl" and by default *fast-dm* tries to read the description for the analysed experiment from the file "experiment.ctl". On systems supporting command line arguments different experiment control files can be chosen by calling *fast-dm* with the name of a control file as the first argument.

Control files are evaluated line by line. Empty lines and lines starting with "#" are ignored. All lines (including the last command line) must be completed by a new-line character (*i.e.*, the 'return' or 'enter' key has to be used). For all other lines the first word is interpreted as a command, the following words are used as arguments to this command. *Table 1* summarises all possible commands. Most commands are optional and some commands may be used more than once: For example, if different stimuli depend on experimental conditions, the command `depends` has to be repeated for each affected parameter. In the control file, commands must always follow the order as presented in Table 1.

The command `precision` *value* defines the accuracy of the calculation of the predicted CDFs. The entered value has to be a real number of 1.0 or above, with accuracy and computation time increasing with higher numbers (values smaller than 2 provide no useful results due to large inaccuracies and values greater than 5 will lead to an unreasonable long processing duration). If the command is not used, the default value of 3.0 is employed.

To set parameters to a fixed value the command `set` *parameter value* has to be applied. Valid entries for *parameter* are a, z, v, t0, sz, sv, and st0. For *value* the desired value of the parameter is entered. For example, to set the variability of the response time constant to zero the command `set st0 0` is used. For the starting point the `set` command works somewhat differently; z is not set to an absolute value but to a fraction of *a*.

3 *Fast-dm* is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. *Fast-dm* is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. When you download the source-code, a copy of the GNU General Public License should accompany this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

4 Additionally, the programs *plot-cdf* and *make-samples* are built. *Plot-cdf* is a small tool to get the function values of the predicted CDF in millisecond-steps for a given set of parameters. It was used to generate *Figure 1.* Make-samples is a program to synthesize data sets for a given set of parameters. This program was used tot generate the samples for experiments 1, 2, and 3. These tools are not described in this paper, but more information can be obtained from the authors or by inspecting the provided source code.

Table 2. Examples of two Control Files

Content of "*experiment.ctl*"	Description

`format REPONSE TIME`
`load "exp.dat"`
`save "results.dat"`
	The file *exp.dat* is loaded. This text file must contain two values in each line: First, the response, and second the reaction time. All parameters (a, z, v, t0, sz, sv, st0) are estimated from the data, and results are saved to file *results.dat.* For the precision, the default value 3.0 is used.

`precision 2.0`
`set z 0.5`
`set st0 0`
`depends a task`
`depends v stim task`
`format task * stim RESPONSE TIME`
`load "*.dat"`
`save "*.out"`
`log "protocol.log"`
	Fast-dm loads all files from with the affix ".dat" from the active directory. If a file is found, parameters are estimated and saved to according files with the affix ".out", and, additionally, to the common output file called *protocol.log.* The starting point is not estimated but fixed to *a / 2* and the variability of t_0 (*i.e.*, st0) is not estimated but fixed to 0. For each of several different tasks, separate values for *a* are estimated. Likewise, different drift rates are estimated for each stimulus type within each task. For the sake of speed, the precision is reduced to 2.

Consequently, the command `set z 0.5` fixes *z* to be equal to 0.5 *a*, that is, the process starts at the midpoint between both thresholds. The `set` command is optional; if no `set` command is provided, all parameters are estimated.

The command `depends` *parameter condition* can be used to allow parameters to take different values in different conditions of an experiment. For the *parameter* argument any parameter name can be inserted (see above for valid parameter names). For *condition* a new name has to be defined here. The name can be chosen freely but it must not contain any blanks. For example, the command `depends v stimulus` means that for each kind of stimulus, which is defined below, another drift rate is estimated. It is also possible that a parameter depends on more than one condition. For example, in a design with two kinds of stimuli (*e.g.*, target *vs.* distractor) and two levels of difficulties (*e.g.*, easy, *vs.* hard) the command `depends v stimulus difficulty` causes four different drift rates to be estimated, one for easy targets, difficult targets, easy distractors and for difficult distractors, respectively. The `depends` command is optional. By default, no parameter depends on experimental conditions.

The first required command is `format`. It specifies the composition of the data files (see below). Specifically, it defines the order of information that is read from each line of a data file. All condition names that are used by a `depends` command have to be specified again in the format definition. Additionally, the template *RESPONSE* has to be used to indicate whether the process outcome of a line of the data file belongs to the lower threshold (value 0) or to the upper threshold (value 1), respectively. The template *TIME* is used for the response time . Asterisks (`*`) indicate that a column of the data file is ignored for the diffusion model analysis (*e.g.*, the trial number). For example, the command `format * * stimulus * difficulty * TIME RESPONSE` indicates that the third column of the data file contains the information about the stimulus type, the fifth column contains difficulty information of a trial, the seventh column contains the reaction time in seconds and the eights column contains the actual response.

The command `load` *"filename"* is also required. The `load` command defines the names of one or more data files. File names have to be enclosed in double quotes. An asterisk may be included in the filename as a wildcard. In this case, all matching files within the current directory are load sequentially, and for each data file, a new set of parameters is estimated. For example, `load "*.dat"` is used to analyse all data files with the extension `".dat"`. Each data file has to have the same format. If the file name does not include an asterisk, fast-dm loads only one data file.

There are two different ways to save the results of *fast-dm*. Firstly, it is possible to save results in *individual save files*, that is, for each data file one save file can be generated, containing the estimated parameter values, the fit-index, and the elapsed calculation time. This is done with the `save` *"filename"* command. Like in the `load` command, the use of an asterisk as wildcard is possible; however, `load` and `save` commands must match:

```
# block trial difficulty target RESPONSE TIME
1 1 easy 0 0 1.475
1 2 easy 1 1 0.815
1 3 hard 0 1 1.007
1 4 easy 0 0 0.906
1 5 hard 1 1 1.131
1 6 hard 1 0 0.525
1 7 hard 0 0 0.835
1 8 easy 1 1 0.722
1 9 easy 0 1 0.916
1 10 hard 0 1 1.112
1 11 hard 0 0 0.673
1 12 easy 1 1 0.775
```

Figure 2. Sample for the first lines of a data file. The first line is ignored by *fast-dm* because it starts with'#'. Afterwards, each line contains the information of one trial of an experiment. For the diffusion-model analysis the first two columns (*i.e.*, 'block' and 'trial') can be ignored. As can be seen in the example, information about experimental conditions of a trial can either be words or numbers.

whenever the `load` command contains an asterisk, so must `save` and vice versa. The `save` command is optional. It can be replaced by the `log` command (see below).

Since individual log files are not very convenient for further processing in most statistical analysis programs, *fast-dm* offers the possibility to save all results within one summary log file. This is done by the command `log` *"filename"*. The log file name must not contain any wildcards. The `log` command is optional. However, *fast-dm* cannot be used without output; therefore, either `save` or `log` has to be used.

Whenever there is a conflict between commands (*e.g.*, if `depends` uses an identifier that is not part of the `load` command), or between commands and data files (*e.g.*, if there are less columns in the data file than defined in the `format` statement) an error message is returned, and *fast-dm* is aborted. *Table 2* gives examples for a very simple and for a more complex control file.

5.3 Data Files

Data files contain the participants' responses and measured reaction times. Optionally they can also contain information about experimental conditions. Typically each data file describes one session of one participant in an experiment. By convention data files use the file name extension ".dat".

Data files are evaluated line by line. Empty lines and lines starting with "#" are ignored. All other lines must contain the entries given by the "format" statement in the control file, in the order defined there and separated by white space.

The values in the "RESPONSE" column must be either 0 (for the lower threshold) or 1 (for the upper threshold), respectively. In the experiment these values will either correspond to the actual response (e.g., left key vs. right key) or to the success value (i.e., correct vs. error). The entries in the "TIME" column must be floating point numbers, giving the reaction time in seconds. All other entries correspond to experimental conditions and can be either numbers or words. *Figure 2* shows an example of the first lines of a data file. The command `format * * difficulty target RESPONSE TIME` could be used in the control file to read this data.

5.4 Save Files

Save files are used by *fast-dm* to store the results of the parameter estimation process. For each data file read a separate save file is written. By convention data files use the file name extension ".out". Save files are plain text files containing the estimated (or fixed) values for all parameters, the probability value of the KS statistic, and the duration of the estimation procedure in seconds. If the `depends` command is applied, parameters are indexed; indices are separated by an underline (_) from the parameter name. As indices the number (or word, respectively) defining the experimental condition in the data file is used. If a parameter depends on more than one condition, indices are given in the order of condition specifiers in the `depends` command.

5.5 Summary Log Files

In summary log files information from different data files can be accumulated within a single output file. The first line of the summary log file contains the names of the variables. Following this, the results for each

```
experiment experiment.ctl (1 data sets):
  precision: 2
  format of "data.dat": * * difficulty target RESPONSE TIME
  optimised parameters: a, z, v_difficulty_target, t0, sz, sv, st0
dataset data.dat:
  a, z, v_easy_0, t0, sz, sv, st0 (9+55 samples)
  a, z, v_easy_1, t0, sz, sv, st0 (46+18 samples)
  a, z, v_hard_0, t0, sz, sv, st0 (18+46 samples)
  a, z, v_hard_1, t0, sz, sv, st0 (37+27 samples)
  ... p = 0.246247
  ... p = 0.47401
  ... p = 0.484875
  -> z = 0.503522
  -> a = 1.239615
  -> v_easy_0 = -1.241465
  -> t0 = 0.256393
  -> sz = 0.109391
  -> sv = 0.281429
  -> st0 = 0.099568
  -> v_easy_1 = 1.201333
  -> v_hard_0 = -0.272049
  -> v_hard_1 = 0.745809
1 dataset processed, total CPU time used: 33.365000s
```

Figure 3. Sample of the screen output of *fast-dm*. The output contains information about the experiment (lines 1 to 4) and about the specific data set being processed (starting from line 5). Because there are four stimulus types, the data set is divided in four sub-samples. For example, in sub-sample 1 (containing stimuli of the type "easy_0") there were "55" response coded with 0 and 9 responses coded with "1". The three *p*-values are the results from the three consecutive runs of the simplex algorithm. Below, the estimates for all parameters are shown.

data file are saved in a new line. The summary log file is convenient to be read by statistical software for further interference statistical analysis of the estimates. To get this kind of output the `log` command has to be used.

5.6 Screen Output

While active, *fast-dm* gives direct output in a text terminal on the screen (see *Figure 3,* for an example). In the first four lines information about the experiment is given. First, the name of the control file is presented. Then, the chosen precision value, and the name (or the template for the name) and format of the data files are shown, as defined by the `load` command. The forth line contains a list of all parameters that are optimised. If any parameter depends on an experimental condition, it is indexed with the name of the condition variable.

Starting from the fifth line, information for a specific data set (*i.e.*, one data file) is presented. First, the name of the data file is provided. Then, all experimental conditions are listed: That is, if one or more parameter depends on a condition variable, the data set is divided in sub samples; if no dependencies are defined, only one sample is listed. In any case, all parameters belonging to one specific condition, are presented in one line. Parameters are indexed as described in the section *log files*. For each condition, the number of both responses (0 *vs.* 1) is specified.

Below, the best fit values (*p*-values) of three consecutive runs of the SIMPLEX algorithm are shown, as soon as a processing loop is finished. Note that a run of the simplex might take some minutes, especially for high precision values, large samples and many conditions. After the third run of the simplex, the estimates for all parameters are shown.

6 Application Examples

6.1 Experiment 1: Simple Models

For Experiment 1, three data samples with 20, 200 and 2000 reactions, respectively, have been simulated from the same sets of parameter values. The data sets were saved to the files "sample1.dat" to "sample3.dat". In the control file the following commands were used:

```
format RESPONSE TIME
```

Table 3. Original and Recovered Parameter Values from the 3 Data Sets of Experiment 1.

		Recovered Values		
Parameter	Original Values	Data Set 1 (N=20).	Data Set 2 (N=200)	Data Set 3 (N=2000).
a	1.00	1.06	1.04	0.97
z	0.50	0.62	0.59	0.48
v	1.00	0.77	0.96	0.98
t_0	0.30	0.30	0.30	0.32
s_z	0.00	0.13	0.11	0.31
s_v	0.00	0.94	0.39	0.23
s_{t0}	0.00	0.00	0.05	0.05
p		.98	1.00	.86
time		30.95	70.81	105.27

Notes. *p*: probability of the *KS*-statistic; *time*: Duration of the parameter estimation procedure in seconds.

Table 4. Original and Recovered Parameter Values from the two Data Sets of Experiment 2.

	Data Set 1 (N=200)		Data Set 2 (N=200)	
Parameter	Original Values	Recovered Values	Original Values	Recovered Values
a	1.00	1.01	1.00	1.05
z	0.50	-	0.80	-
v	1.00	1.11	0.00	2.23
t_0	0.30	0.29	0.30	0.23
s_z	0.00	-	0.00	0.00
s_v	0.00	-	0.00	0.75
s_{t0}	0.00	-	0.00	0.03
p		.90		.01
time		0.27		70.36

Notes. In the parameter estimation procedure, *z* was fixed to 0.5 *a* for both data sets. For data set 1, the variability parameters were fixed to 0. *p*: probability of the *KS*-statistic; *time*: Duration of the parameter estimation procedure in seconds.

```
load "sample*.dat"
save "sample*.out"
```

The original values used for the simulation as well as the results of *fast-dm* are shown in *Table 3*. As can be seen, the program can recover most parameter values well. Even from only 20 reaction times it is possible to get a rough estimate of the true parameter values. It is much more difficult to reestimate the variability parameters. The reason for this difficulty is that they have a rather small influence on the shape of the CDF; nonetheless including these parameters might in some cases increase model fit considerably (Ratcliff & Tuerlinckx, 2002).

6.2 Experiment 2: Fixing Parameters

Experiment 2 gives two examples of how to fix certain parameters (*Table 4*). To analyse data set 1, the following control file was used.

```
set z 0.5
set sz 0
set sv 0
set st0 0
format RESPONSE TIME
load "sample1.dat"
save "sample1.out"
```

Table 5. Original and Recovered Parameter Values from the 4 Data Sets of Experiment 2 (200 responses per data set).

	Original				Recovered			
	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Conditions								
c1	1	1	2	2	1	1	2	2
c2	1	2	1	2	1	2	1	2
Parameter								
a	1.00				1.01			
z	0.70		0.30		0.69		0.33	
v	2.00	0.50	2.00	0.50	2.16	0.41	1.76	0.37
t_0	0.30				0.30			
p					0.99			
t					11.68			

Notes. p: probability of the *KS*-statistic; *time*: Duration of the parameter estimation procedure in seconds.

As can be seen from Table 4, parameters are recovered well, and the computation of this simple model is very fast. For data set 2 fixations were chosen that prohibited recovering the original values: Data was simulated for an "asymmetric model", that is, a model with z = 0.8 *a*. Nonetheless, for the recovery, *z* was set to 0.5 *a*. Thus it should be tested whether wrong fixations lead to a statistical misfit. The following commands were used:

```
set z 0.5
format RESPONSE TIME
load "exp2_2.dat"
save "exp2_2.out"
```

The results show that estimates for the drift and for the variability of the drift are poor. This is due to a trade-off effect: because the high amount of responses at the upper threshold cannot be mapped on *z*, it is mapped on *v*. Nonetheless, the predicted CDF does not fit the data well: The significant *p*-value of the model indicates a model misfit, as was expected.

6.3 Experiment 3: Hierarchical Models

For Experiment 3 a more complex data set was simulated: Four experimental conditions defined by a 2 x 2 design were assumed were *z* and *v* depend on conditions while all other parameters were assumed to be constant over conditions (*Table 5*). For each of the four cells of the design, 200 responses were simulated, resulting in a total sample of 800 responses. The following commands were used in the control file:

```
depends z c1
depends v c1 c2
set sz 0
set sv 0
set st0 0
format c1 c2 RESPONSE TIME
load "exp3.dat"
save "exp3.out"
```

The input consists of one data file (called "exp3.dat") with 4 data columns: First, there are two variables coding the conditions as in *Table 5,* followed be the response (0 vs. 1) and the response time. The `depends` commands cause a hierarchical model, in which all parameters have the same value across conditions besides *z* (depending on *c1*) and *v* (depending on both *c1* and *c2*). Obviously, other models would be possible as well. For example, the simulated data could also be recovered if the drift would only depend on the condition variable *c2*. Variability parameters are set to zero. Again, *fast-dm* could recover most parameter values well (*Table 5*).

7 Summary

Fast-dm is a new and flexible tool that allows for a fast and precise diffusion-model data analysis. The program uses the PDE method recently introduced by Voss & Voss (2006) and for the parameter estimation the

Kolmogorov-Smirnov statistic is applied (Voss, et al., 2004). *Fast-dm* estimates all parameters of Ratcliff's (1978) diffusion model ($a, z, v, t_{0,}, s_z, s_v,$ and s_{to}). Also, hierarchical models can be fitted, that is, some parameters may be valid for a whole data set, while other parameters are allowed to vary between conditions. It is also possible to fix parameters to given values, thereby making more simple models possible (*e.g.*, by setting the variability parameters to 0 or by fixing *z* to *a/2*).

Fast-dm is free software and the source code is available at *http://www.psychologie.uni-freiburg.de/Members/voss/fast-dm* . For Microsoft Windows, we provide also executable binaries of *fast-dm*. The program can be used, redistributed and modified under the terms of the GNU General Public License (see footnote 3 for details);

8 References

Conover, W. J. (1999). *Practical nonparametric statistics (3rd Ed.)*. New York: Wiley.

Feller, W. (1971). *An introduction to probability theory and its applications, (2nd. Ed.). Vol. II.* New York: Wiley.

Morton, K. & Mayers, D. (1994). *Numerical Solution of Partial Differential Equations.* Cambridge University Press.

Nelder, J.A., & Mead, R. (1965). A simplex method for function minimization. *Computer Journal, 7*, 308-313.

Press, W.H., Teukolsky, S.A., & Vetterling, W.T. (1992). *Numerical Recipes in C (2nd Ed.).* Cambridge University Press.

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review, 85*(2), 59-108.

Ratcliff, R., & Rouder, J. N. (1998). Modelling response times for two-choice decisions. *Psychological Science, 9*(5), 347-356.

Ratcliff, R., & Tuerlinckx, F. (2002). Estimating parameters of the diffusion model: Approaching to dealing with contaminant reaction and parameter variability. *Psychonomic Bulletin and Review, 9*(3), 438-481.

Vandekerckhove, J., & Tuerlinckx, F. (2006). *Fitting the Ratcliff diffusion model to experimental data.* Manuscript submitted for publication.

Voss, A., Rothermund, K., & Voss, J. (2004). Interpreting the parameters of the diffusion model: An empirical validation. *Memory and Cognition, 32,* 1206–1220.

Voss, A., & Voss, J. (2006). *A Fast Numerical Algorithm for the Estimation of Diffusion-Model Parameters.* Manuscript submitted for publication.

Wagenmakers, E.-J., van der Maas, H.L.J., Grasman, R.P.P.P. (in press). An ez-diffusion model for response time and accuracy. *Psychonomic Bulletin & Review.*