

tweezercalib 2.0: Faster version of MatLab package for precise calibration of optical tweezers [☆]

Poul Martin Hansen ^a, Iva Marija Tolić-Nørrelykke ^{a,1}, Henrik Flyvbjerg ^b,
Kirstine Berg-Sørensen ^{a,*}

^a The Niels Bohr Institute, Blegdamsvej 17, DK-2100 Copenhagen Ø, Denmark

^b Biosystems Department and Danish Polymer Centre, Risø National Laboratory, DK-4000 Roskilde, Denmark

Received 28 October 2005; received in revised form 28 October 2005; accepted 23 November 2005

Available online 18 January 2006

Abstract

We present a vectorized version of the MatLab (MathWorks Inc.) package *tweezercalib* for calibration of optical tweezers with precision. The calibration is based on the power spectrum of the Brownian motion of a dielectric bead trapped in the tweezers. Precision is achieved by accounting for a number of factors that affect this power spectrum, as described in vs. 1 of the package [I.M. Tolić-Nørrelykke, K. Berg-Sørensen, H. Flyvbjerg, Matlab program for precision calibration of optical tweezers, Comput. Phys. Comm. 159 (2004) 225–240]. The graphical user interface allows the user to include or leave out each of these factors. Several “health tests” are applied to the experimental data during calibration, and test results are displayed graphically. Thus, the user can easily see whether the data comply with the theory used for their interpretation. Final calibration results are given with statistical errors and covariance matrix.

New version program summary

Title of program: tweezercalib

Catalogue identifier: ADTV_v2_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/ADTV_v2_0

Program obtainable from: CPC Program Library, Queen’s University of Belfast, N. Ireland

Reference in CPC to previous version: I.M. Tolić-Nørrelykke, K. Berg-Sørensen, H. Flyvbjerg, Comput. Phys. Comm. 159 (2004) 225

Catalogue identifier of previous version: ADTV

Does the new version supersede the original program: Yes

Computer for which the program is designed and others on which it has been tested: General computer running MatLab (Mathworks Inc.)

Operating systems under which the program has been tested: Windows2000, Windows-XP, Linux

Programming language used: MatLab (Mathworks Inc.), standard license

Memory required to execute with typical data: Of order four times the size of the data file

High speed storage required: none

No. of lines in distributed program, including test data, etc.: 135 989

No. of bytes in distributed program, including test data, etc.: 1 527 611

Distribution format: tar. gz

[☆] This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

* Corresponding author. Present address: Physics Department, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark. Tel.: +45 45 25 31 01, fax: +45 45 93 16 69.

E-mail address: kirstine.berg-sorensen@fysik.dtu.dk (K. Berg-Sørensen).

¹ Present address: Max Planck Institute of Molecular Cell Biology and Genetics, Dresden. Permanent address: Rugjer Bošković Institute, Bijenička 54, HR-10000 Zagreb, Croatia.

Nature of physical problem: Calibrate optical tweezers with precision by fitting theory to experimental power spectrum of position of bead doing Brownian motion in incompressible fluid, possibly near microscope cover slip, while trapped in optical tweezers. Thereby determine spring constant of optical trap and conversion factor for arbitrary-units-to-nanometers for detection system.

Method of solution: Elimination of cross-talk between quadrant photo-diode's output channels for positions (optional). Check that distribution of recorded positions agrees with Boltzmann distribution of bead in harmonic trap. Data compression and noise reduction by blocking method applied to power spectrum. Full accounting for hydrodynamic effects: Frequency-dependent drag force and interaction with nearby cover slip (optional). Full accounting for electronic filters (optional), for "virtual filtering" caused by detection system (optional). Full accounting for aliasing caused by finite sampling rate (optional). Standard non-linear least-squares fitting. Statistical support for fit is given, with several plots facilitating inspection of consistency and quality of data and fit.

Summary of revisions: A faster fitting routine, adapted from [J. Nocedal, Y.x. Yuan, Combining trust region and line search techniques, Technical Report OTC 98/04, Optimization Technology Center, 1998; W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes. The Art of Scientific Computing, Cambridge University Press, Cambridge, 1986], is applied. It uses fewer function evaluations, and the remaining function evaluations have been vectorized. Calls to routines in Toolboxes not included with a standard MatLab license have been replaced by calls to routines that are included in the present package. Fitting parameters are rescaled to ensure that they are all of roughly the same size (of order 1) while being fitted. Generally, the program package has been updated to comply with MatLab, vs. 7.0, and optimized for speed.

Restrictions on the complexity of the problem: Data should be positions of bead doing Brownian motion while held by optical tweezers. For high precision in final results, data should be time series measured over a long time, with sufficiently high experimental sampling rate: The sampling rate should be well above the characteristic frequency of the trap, the so-called *corner frequency*. Thus, the sampling frequency should typically be larger than 10 kHz. The Fast Fourier Transform used works optimally when the time series contain 2^n data points, and long measurement time is obtained with $n > 12$ –15. Finally, the optics should be set to ensure a harmonic trapping potential in the range of positions visited by the bead. The fitting procedure checks for harmonic potential.

Typical running time: Seconds

Unusual features of the program: None

References: The theoretical underpinnings for the procedure are found in [K. Berg-Sørensen, H. Flyvbjerg, Power spectrum analysis for optical tweezers, Rev. Sci. Ins. 75 (2004) 594–612].

© 2005 Elsevier B.V. All rights reserved.

PACS: 87.80.-y; 06.20.Dk; 07.60.-j; 05.40.Jc

Keywords: Optical tweezers; Calibration; Power spectrum analysis; Precision calibration of optical tweezers

1. Revisions

In this new version of the program, we avoid the use of MatLab Toolboxes not included with a standard MatLab license. The present package can thus be applied with the MatLab installations of more users.

We have constructed a new fitting routine, `fit_nonl.m` and a new linesearch routine, `linesearch.m`, based on recommendations in [1] and [2], respectively. These routines replace the MatLab toolbox function `LSQNONLIN`. The routine `fit_nonl.m` is based on Algorithms 2.6 and 3.1 in [1]. The routine `linesearch.m` is constructed in analogy with `LNSRCH` in [2], it is used to perform the backtracking part of `fit_nonl.m`. These two routines use fewer function evaluations than `LSQNONLIN` which makes the fitting procedure significantly faster.

A small error in the calculation of the backing of the fits has been corrected. This correction increases the backings of the two fits in the testcase from 37% to 42% and from 61% to 65%, i.e. by insignificant amounts.

Fitting parameters have been rescaled such that they all take values of order 1. This makes fitting more robust. Final results are, of course, stated in physical units, and rescaling is done automatically, with no intervention from the user.

Evaluation of the function fitted has been optimized for MatLab through vectorization. Index-matching, previously performed through calls to the time-consuming MatLab standard function `FIND`, has been rephrased using a new function `quickfind.m`. In general, throughout the present package calls

of array elements have been rephrased in a manner optimizing speed. When blocking, values of indices are kept and reused through the MatLab feature `PERSISTENT`.

The graphical user interface has been revised to comply with MatLab 7.0. A new button `Close all` that closes all figures apart from the input window [3, Fig. 1] has been added to the lower right corner of the input window. Also, a help-file, `ahelp.m`, has been added to the program package.

Figures showing histograms of positions now also display an inset lin-log plot that will reveal non-Gaussian tails in the histogram. The Gaussian plotted on top of the histogram of positions is not a fit, but simply the Gaussian with same norm, mean, and variance as the data that were binned to form the histogram. It would have been better to show a Gaussian resulting from a maximum likelihood fit to the histogram. But the positions that are binned to form the histogram, are typically sampled much faster than $t_{\text{trap}} = 1/(2\pi f_c)$, the characteristic time for decorrelation of the bead in the trap (see [4]). As a result, they are highly correlated by the dynamics of the bead. Consequently, we do not know the correct probability (likelihood) to optimize. Thus the Gaussian plotted on top of the histogram is merely a useful benchmark for this first, simple test of the quality of the data by visual inspection. It will not yield the statistical support for the hypothesis we are testing here, that the trapping force is Hookean in the range of positions visited by the bead. But this plot will indicate if the hypothesis is obviously false. If not, later fits *will* yield such a support, but for

Table 1
Comparison of CPU time used by version 1 and 2.0 of our MatLab code. A data set of 2^{17} points was used. The computer was a Pentium 4, 2.6 GHz, and the operating system was WindowsXP

Part of program	CPU time consumed by vs. 1	CPU time consumed by vs. 2.0
View data	50 s	5.5 s
Fit spectra	1110 s	3.9 s
Distribution of $P^{(ex)}/P^{(fit)}$	110 s	0.9 s

the whole dynamical theory for the bead's motion, including a Hookean trapping force [4].

Apart from this change, data processing and program use is unchanged since vs. 1 of the program, and is described in detail in [3].

2. CPU times

Typical CPU times consumed by vs. 1 of the program package (tweezercalib) and of the present, updated version (tweezercalib 2.0) are given in Table 1.

3. Possible further extensions

The present version 2.0 of the program package is so fast that most users can analyze their data in seconds, right after recording them. Thus the program package is now a tool that can be used to optimize experiments and experimental setups

while working on them. If data-acquisition is done with, e.g., LabView (National Instruments), the LabView program may be modified by the user to call directly the MatLab package presented here. LabView instruments that call tweezercalib 2.0 have been constructed and applied with good results in the optical tweezers laboratory at the Niels Bohr Institute, University of Copenhagen [5].

Acknowledgements

We are grateful to all users of vs. 1 of the package who have given us feed-back. Our in-house users in the optical tweezers laboratory at the Niels Bohr Institute have been a continuous source of stimulation. This project is supported by the Danish Natural Science Research Council, the Carlsberg Foundation and the Lundbeck Foundation.

References

- [1] J. Nocedal, Y.x. Yuan, Combining trust region and line search techniques, Technical Report OTC 98/04, Optimization Technology Center, 1998.
- [2] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes. The Art of Scientific Computing, Cambridge University Press, Cambridge, 1986.
- [3] I.M. Tolić-Nørrelykke, K. Berg-Sørensen, H. Flyvbjerg, Matlab program for precision calibration of optical tweezers, Comput. Phys. Comm. 159 (2004) 225–240.
- [4] K. Berg-Sørensen, H. Flyvbjerg, Power spectrum analysis for optical tweezers, Rev. Sci. Ins. 75 (2004) 594–612.
- [5] D. Trpceviski, Private communication.