PFMCal: Photonic force microscopy calibration extended for its application in high-frequency micro-rheology

A. Butykai[a], F. M. Mor[b], R. Gaál[b], P. Domínguez-García[c,*], L. Forró[b], S. Jeney[b]

[a] *Department of Physics, Budapest University of Technology and Economics and MTA-BME Lendület Magneto-optical Spectroscopy Research Group, 1111 Budapest, Hungary*
[b] *Laboratory of Physics of Complex Matter, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland*
[c] *Dep. de Física Interdisciplinar, Universidad Nacional de Educación a Distancia (UNED), Madrid 28040, Spain*

Abstract

The present document is an update of the previously published MatLab code for the calibration of optical tweezers in the high-resolution detection of the Brownian motion of non-spherical probes [1]. In this instance, an alternative version of the original code, based on the same physical theory [2], is outlined, but this is focused on the automation of the calibration of measurements using spherical probes. The new added code is quite useful for high-frequency micro-rheology studies, where the probe radius is known but the viscosity of the surrounding fluid maybe not. This extended calibration methodology is automatic, without the need of a user's interface. A code for calibration by means of thermal noise analysis [3] is also included; this is a method that can be applied when using viscoelastic fluids if the spring constant of the trap is previously estimated [4]. The new code can be executed in MatLab and using GNU Octave.

Keywords: Calibration of optical tweezers, Brownian motion, Power spectral density (PSD), Mean square displacement (MSD), Velocity autocorrelation function (VAF)

*Corresponding author.
E-mail address: pdominguez@fisfun.uned.es

NEW VERSION PROGRAM SUMMARY

Program Title: PFMCal

Licensing provisions: GPLv3

Programming language: MatLab 2016a (MathWorks Inc.) and GNU Octave 4.0

Operating system: Linux and Windows.

Supplementary material: A new document `README.pdf` *includes basic running instructions for the new code.*

Journal reference of previous version: Computer Physics Communications, 196 (2015) 599

Does the new version supersede the previous version?: No. It adds alternative but compatible code while providing similar calibration factors.

Nature of problem (approx. 50-250 words): The original code uses a MatLab-provided user's interface, which is not available in GNU Octave, and cannot be used outside of a proprietary software as MatLab. Besides, the process of calibration when using spherical probes needs of an automatic method when calibrating long amounts of different data focused to micro-rheology.

Solution method (approx. 50-250 words): The new code can be executed in the latest version of MatLab and using GNU Octave, a free and open-source alternative to MatLab. This code generates an automatic calibration process which requires only to write the input data in the main script. Additionally, we include a calibration method based on thermal noise statistics, which can be used with viscoelastic fluids if the trap stiffness is previously estimated.

Reasons for the new version: This version extends the functionality of `PFMCal` *for the particular case of spherical probes and unknown fluid viscosities. The extended code is automatic, works in different operating systems and it is compatible with GNU Octave.*

Summary of revisions: The original MatLab program in the previous version, which is executed by `PFMCal.m`*, is not changed. Here, we have added two addi-*

Fluid	η	k	β_{MSD}	β_{VAF}	β_{PSD}	β_{σ^2}	β_E
Water	1.02	6.5	11.9	10.7	12.7	11.75 ± 0.03	11.9 ± 0.2
`PFMCal`	[0.98]	6.4	$\beta = 12.2$				
Acetone	0.33	6.1	10.0	9.7	9.6	9.81 ± 0.01	9.9 ± 0.2
`PFMCal`	[0.33]	5.8	$\beta = 10.1$				

Table 1: Comparison between calibration factors and trap stiffnesses, β (μm/V) and k (μN/m), for interferometry-based optical tweezers, calculated for spherical probes with radius $a = 0.94$ μm in water or acetone at 21°C by using automatic methods here exposed and by means of the GUI-based original code `PFMCal`. Density of the bead $\rho_p = 1.57$ g/cm^3, water density 0.997 g/cm^3 and acetone density 0.791 g/cm^3. Values between square brackets are fixed inputs. The extended calibration methodology provides the expected zero-shear viscosities η (mPa.s) values. The obtained values for β and k using the methods summarized in this table are compatible between them.

*tional main archives named **PFMCal_auto.m** and **PFMCal_histo.m**, which implement automatic calculations of the calibration process and calibration through Boltzmann statistics, respectively. The process of calibration using this code for spherical beads is described in the **README.pdf** file provided in the new code submission. Here, we obtain different calibration factors, β (given in μm/V), according to [2], related to two statistical quantities: the mean-squared displacement (MSD), β_{MSD}, and the velocity autocorrelation function (VAF), β_{VAF}. Using that methodology, the trap stiffness, k, and the zero-shear viscosity of the fluid, η, can be calculated if the value of the particle's radius, a, is previously known. For comparison, we include in the extended code the method of calibration using the corner frequency of the power-spectral density (PSD) [5], providing a calibration factor β_{PSD}.*

Besides, if we are able to obtain an estimation of k, along with the known value of the particle's radius, we can use thermal noise statistics to obtain calibration factors, β, according to the quadratic form of the optical potential, β_E, and related to the Gaussian distribution of the bead's positions, β_{σ^2}. This method has been applied to the calibration of optical tweezers when using non-Newtonian viscoelastic polymeric liquids [4].

*An example of the results using this calibration process is summarized in Table 1. Using the data provided in the new code submission, for water and acetone fluids, we calculate all the calibration factors by using the original **PFMCal.m** and by the new non-GUI code **PFMCal_auto.m** and **PFMCal_histo.m**. Regarding the new code, **PFMCal_auto.m** returns η, k, β_{MSD}, β_{VAF} and β_{PSD}, while **PFMCal_histo.m** provides β_{σ^2} and β_E. Table 1 shows how we obtain the expected viscosity of the two fluids at this temperature and how the different methods provide* *good agreement between trap stiffnesses and calibration factors.*

Additional comments including Restrictions and Unusual features (approx. 50-250 words): The original code, `PFMCal.m`*, runs under MatLab using the Statistics Toolbox. The extended code,* `PFMCal_auto.m` *and* `PFMCal_histo.m`*, can be executed without modification using MatLab or GNU Octave. The code has been tested in Linux and Windows operating systems.*

1. Acknowledgements

A.B. was supported by the Hungarian Research Funds, OTKA K108918. F.M.M. acknowledges support from the National Centre of Competence in Research (NCCR Nano-Bio, Project 1.4) and the Swiss National Science Foundation (SNSF, Project numbers 206021_121396 and 200021_143703). F.M.M. was supported by the National Competence Center in Biomedical Imaging (NCCBI). P.D.G acknowledges M.E.C.D. for financial aid by project FIS2013-47350-C5-5-R.

[1] A. Butykai, F. Mor, R. Gaál, P. Domínguez-García, L. Forró, J. S., Calibration of optical tweezers with non-spherical probes via high-resolution detection of brownian motion, Comput. Phys. Commun. 196 (2015) 599 – 610.

[2] M. Grimm, T. Franosch, S. Jeney, High-resolution detection of brownian motion for quantitative optical tweezers experiments, Phys. Rev. E 86 (2012) 021912.

[3] E.-L. Florin, A. Pralle, E. H. K. Stelzer, J. K. H. Hörber, Photonic force microscope calibration by thermal noise analysis, Appl. Phys. A 66 (1998) S75–S78.

[4] P. Domínguez-García, L. Forró, S. Jeney, Interplay between optical, viscous, and elastic forces on an optically trapped brownian particle immersed in a viscoelastic fluid, Appl. Phys. Lett. 109 (14) (2016) 143702.

[5] K. Berg-Sørensen, H. Flyvbjerg, Power spectrum analysis for optical tweezers, Rev. Sci. Instrum. 75 (3) (2004) 594–612.