Journal of Non-Newtonian Fluid Mechanics 261 (2018) 153–163

Contents lists available at ScienceDirect

Journal of Non-Newtonian Fluid Mechanics

journal homepage: www.elsevier.com/locate/jnnfm

A simple and efficient numerical procedure to compute the inverse Langevin function with high accuracy

José María Benítez, Francisco Javier Montáns*

Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros, 3, Madrid, 28040, Spain

ARTICLE INFO

Keywords:
Inverse langevin function
Spline approximation
Polymers
Chain models

ABSTRACT

The inverse Langevin function is a fundamental part of the statistical chain models used to describe the behavior of polymeric-like materials, appearing also in other fields such as magnetism, molecular dynamics and even biomechanics. In the last four years, several approximants of the inverse Langevin function have been proposed. In most of them, optimization techniques are used to reduce the relative error of previously published approximants to reach orders of magnitude of $O(10^{-3}\% - 10^{-2}\%)$. In this paper a new simple and efficient numerical approach to calculate the inverse Langevin function is proposed. Its main feature is the reduction of the relative errors in all the domain $x = [0, 1)$ to near machine precision, maintaining function evaluation CPU times similar to those of the most efficient approximants. The method consists in the discretization of the Langevin function, the calculation of the inverse of these discretization points and their interpolation by cubic splines. In order to reproduce the asymptotic behavior of the inverse Langevin function, a rational function is considered only in the asymptotic zone keeping C^1 continuity with the cubic splines. We include customizable Matlab codes to create the spline coefficients, to evaluate the function, and to compare accuracy and efficiency with other published proposals.

1. Introduction

The inverse Langevin function is frequently used in fields like polymer science [1–3], rubber hyperelasticity [4,5] and biomechanics [6]. In these contexts, a single molecular chain can be modeled as a freely-joined chain (FJC), which is composed of n linked rigid segments of equal length l, randomly coiled, so that $L = nl$ is the contour length of the chain. Its configuration is defined by the end-to-end distance or tie points distance, r, that is the length between the ends of the chain. When $r = 0$, both points are coincident and the entropy is maximum, whereas if the chain is cross-linked, the entropy decreases since the number of possible configurations that the chain can take also decreases [7,8]. The entropy of a molecule network, S, composed by N chains per reference volume, can be defined in terms of the available configurations, which are mathematically given by a probability distribution function (PDF) $p(r)$

$$S = c + N k_B \ln(p(r)) \tag{1}$$

where c is a constant and k_B is the Boltzmann constant.

When a single chain, $N = 1$, develops finite strain, Gaussian PDFs are no longer valid since they do not account for the limit of the chain extensibility when $r \to L = nl$. Thus, a PDF based on the inverse Langevin function is used so that a relationship between the force applied in a single chain and its extensibility can be obtained through

$$f = -T\frac{\partial S}{\partial r} = \frac{k_B}{l} T \mathcal{L}^{-1}\left(\frac{r}{nl}\right) \tag{2}$$

where T is the absolute temperature and $\mathcal{L}^{-1}$ is the inverse Langevin function [8]. The Langevin function is defined as

$$x = \mathcal{L}(y) = \coth(y) - \frac{1}{y} \tag{3}$$

There is no explicit form of its inverse, $\mathcal{L}^{-1}(x)$, and several approximants have been proposed. The first one was proposed by Kuhn and Grün [9] and consists in a Taylor expansion series around $x = 0$. Although, as expected, approximants based on Taylor expansion series show a good accuracy around the fixed point $x = 0$, they have divergence problems in the vicinity of $x = 1$, when the chain starts to be fully stretched, so $r \to L = nl$. Recent studies have proven that with 115 series terms, a good accuracy is found in the interval [0, 0.95] and that above 500 terms the convergence radius remains unchanged [10,11]. It is obvious that the computational evaluation of such number of terms is exceedingly expensive.

To account for the singularity of the inverse Lanvegin function at $x = 1$, several Padé approximants have been proposed. These approximants are rational functions denoted by $[m/n]$, where m and n are the degree of the polynomials corresponding to the numerator and the

* Corresponding author.
E-mail addresses: josemaria.benitez@upm.es (J.M. Benítez), fco.montans@upm.es (F.J. Montáns).

https://doi.org/10.1016/j.jnnfm.2018.08.011
Received 19 June 2018; Received in revised form 23 August 2018; Accepted 31 August 2018
Available online 5 September 2018

Table 1
Maximum relative error of the main approximants, their complexity as suitable for analytical treatment (polynomials with integer-valued powers) and comparative mean CPU time in their evaluation. These approximants (except Cohen's, which have the largest error) are included in the Matlab script given in the Appendix.

Approximant	ε_r (%)	Complexity	CPU time (10^{-8} s)
Cohen [13]	4.94	5	5.77
Kröger [2]	$2.75 \cdot 10^{-1}$	9	8.25
Petrosyan [19]	$1.79 \cdot 10^{-1}$	-	13.3
Nguesson et al. [17]	$4.65 \cdot 10^{-2}$	-	42.17
Jedynak [11]	$7.69 \cdot 10^{-2}$	11	9.94
Marchi and Arruda [18]	$4.37 \cdot 10^{-3}$	-	43.63

denominator, respectively. They account for the singularity at $x = 1$ in terms of $1/(1-x^i)^j$, being i and j natural numbers excluding the zero. Examples of this kind of approximant are the proposals of Warner [12], Cohen's rounded Padé proposal [13] and Puso [14]. These approximants are usually known as single-point Padé approximants, since the polynomial coefficients are determined with the information of the function $\mathcal{L}^{-1}(y)$ and its derivatives that is given by Taylor expansion series around $x = 0$. When these coefficients are calculated by evaluating the function in some points within the domain [0, 1) the approximants are called multi-point Padé approximants. Examples of multi-point Padé approximants have been proposed by Jedynak [15] and Darabi and Itskov [16]. Kröger [2] improved some of the aforementioned single- and multi-point Padé approximants considering the exact asymptotic, symmetry and integral behavior of the inverse Langevin function.

In the last years, a new trend of improving some of the existing approximants by optimization methods has emerged. Such is the case of the error-corrected approximants developed by Nguessong et al. [17]. They improved the accuracy of some of the above mentioned multi-point Padé approximants calculating the error functions corresponding to the approximant and subtracting both. To do so, Neguessong et al. [17] developed an optimization of the error function parameters by a least squares minimization. Following this idea, Marchi and Arruda [18] proposed not only the optimization of these parameters but also the polynomial coefficients of the approximant to minimize its relative error. Recently, Petrosyan [19] deduced a function accounting for the asymptotic behavior of the inverse Langevin function and minimized its absolute error with a sine and a quadratic function.

Also recently, Jedynak [11] improved the approximants given by Kröger [2] and proposed a new one applying the minimax approximation theory. The approximation theory is a method to determine the degree of the polynomial or rational approximation that minimizes the error of a certain function. Jedynak minimized the relative error of the inverse Langevin function increasing the complexity of the Kröger interpolant to 11, that is to say with a rational function of [9/2] degree. The polynomial coefficients were determined solving a system of nonlinear equations by means of Remez's algorithm.

An approximant not based on Taylor series or Padé functions was proposed by Bergström. It is a piecewise function resulting from dividing the domain in two subdomains to account for the asymptotic behavior of $\mathcal{L}^{-1}$, Ref. [20]. Despite having good accuracy, this approximant has some drawbacks that make its application to physical problems difficult, see Refs. [15] and [18].

The fact that most of the above-mentioned papers have been published in the last four years, from 2014 to 2017, shows that the calculation of the inverse Langevin function is of much current interest and that a satisfactory, computationally efficient and accurate solution has not been reached yet [2,11,15,16,18,19]. These works have been mainly focussed on improving the accuracy of previous researches. To summarize and to compare the accuracy of the main approximants proposed, their maximum relative error, ε_r, is shown in Table 1. According to these data, a lot of effort has been invested to reduce the maximum relative error of the rounded Cohen approximant in only one order of magnitude, $O(10^{-1}\%)$ in the case of Kröger or Petrosyan. The order of magnitude of the error has not improved substantially with the error minimizing techniques. With this optimization methods the order of magnitude of the relative error is $O(10^{-2} - 10^{-3}\%)$ [11,17,18].

The accuracy of the inverse Langevin function is not a minor issue since it has a relevant influence in the results obtained in computational simulations. Indeed, the inverse Langevin function is used in many models implemented in finite element codes [1,21–24], including commercial finite element programs such as ABAQUS or ADINA. During simulations, the inverse Langevin function is typically evaluated millions of times, so iterative methods are avoided and explicit approximants are preferred. Recently, Ammar has shown the important influence of the approximants accuracy in the results obtained in simple finite element simulations in the framework of the dilute polymer kinetic theory [1]. When complex calculations are performed, the approximant accuracy can be a critical factor to obtain reliable results. Therefore, an approximation of the inverse Langevin function with a simple and computationally efficient implementation in a finite element code, and with a suitable accuracy, is needed to ensure adequate efficient computational predictions.

In this paper we present a simple computationally-oriented technique to calculate the inverse Langevin function that reduces the maximum relative error close to the one that a computer can obtain. In this proposal, the inverse Langevin function is obtained by means of a cubic spline representation of the function. However, to reproduce the asymptote at $x = 1$, the discretization close to the asymptotic zone is replaced by a [1/2] rational function that fulfills $\mathcal{C}^1$ continuity conditions with cubic splines, so constitutive tangents keep continuity. Thus, the approximation to the inverse Langevin function consists in a series of piecewise polynomials and a rational function in the vicinity of the asymptote. The derivatives and integrals are also immediate (explicit) and continuous. We note that although this method requires the calculation of the spline coefficients, these coefficients are obtained just once. In fact, the coefficients can be previously stored and embedded in the code at program compilation time. The price to pay for high accuracy and efficiency is an increased storage. However, the storage needs are negligible for current digital devices. For example, if the Langevin function discretization is performed in 10,001 points (a number of points that gives high accuracy as shown below), 10,000 cubic splines must be calculated. Considering that every polynomial has four coefficients, 40,000 coefficients have to be stored in the computer memory. Taking into account that a double precision number is stored in 8 bytes, 320kB are necessary to store all the spline coefficients, which is at least four orders of magnitude less than the typical RAM memory (e.g. 6GB) of an economic laptop computer or a graphic card (GPU), and also several orders of magnitude less than the typical memory needed to solve an industry problem. Near machine precision is obtained with 100,000 spline pieces which takes just about 3MB of memory. Remarkably, function evaluation times are almost independent of the desired, and obtained, accuracy.

Matlab codes are given in the appendices and commented in the text below. Although the method is explained using Matlab, its implementation in any computer language is straightforward.

2. Determination of the inverse Langevin function by cubic splines interpolation

2.1. Splines

Piecewise cubic splines are cubic interpolating polynomials which have minimum curvature and preserve continuity of derivatives between contiguous segments up to second order. Some background on splines may be found, for example in Reference [25]. Some applications in the context of hyperelasticity in polymers and biological tissues may be found, for example, in [26–30].