Numerical computations of Marcus–hush–Chidsey electron transfer rate constants

Vincent Fourmond, Christophe Léger

Numerical computations of Marcus–Hush–Chidsey electron transfer rate constants

V. Fourmond*[a], C. Léger[a]

[a] *Aix-Marseille Université, CNRS, BIP UMR 7281, 31 Chemin J. AIGUIER, CS70071, F-13402 Marseille Cedex 20, (France)*

Abstract

Two theories are available to predict the values of the rate constants of electron transfer between a redox species and an electrode as a function of electrode potential: the Butler-Volmer theory, and the Marcus "Density of States" (MDoS) theory. While the Butler-Volmer is purely empirical, it is widely used, in part because of its ability to represent experimental data, but also because of its simplicity: the rates are just exponential functions of the potential. By contrast, in the case of the MDoS theory, whose justification comes from the well-established theory of electron transfer proposed by Marcus, the rates are expressed in the form of indefinite integrals, which are harder to compute. A number of algorithms have been presented in the literature, with various merits in terms of accuracy and computation time; however, these algorithms have never been compared under the same conditions, which prevent making informed choices. We systematically compared the algorithms, both in terms of accuracy and computation time, and also propose a new algorithm which is very fast and reasonably accurate.

1. Introduction

Two distinct models predict the rate of electron transfer between a redox species (in solution or adsorbed) and an electrode, like in the following reaction:

$$\mathrm{Ox} + \mathrm{e}^- \underset{k_{\mathrm{ox}}(E)}{\overset{k_{\mathrm{red}}(E)}{\rightleftharpoons}} \mathrm{Red} \tag{1}$$

The first one is the Butler-Volmer equation, which is based on the assumption that the rate of electron transfer increases exponentially as a function of driving force. Although purely empirical, it is remarkably useful in the general case[1]. The second one is based on Marcus' theory of electron transfer. Marcus predicted that the rate of electron transfer between two moieties in solution can be written under the following form[2]:

$$k_{\mathrm{ox/red}} = k_0 \exp \frac{(\Delta G \pm \lambda)^2}{4RT\lambda} \tag{2}$$

in which ΔG is the driving force for the reaction, R the gas constant, T the temperature, k_0 a prefactor and λ is the so-called reorganization energy, which corresponds to the changes in the configuration of the molecules and the solvent before and after the electron transfer.

This equation can be extended to interfacial electron transfer by considering the electrode as a distribution of states following a Fermi-Dirac statistics, hence the commonly used name "Marcus Distribution of States" or "Marcus Density of States" (MDoS hereafter) for this theory. The theory is also known under the names "Marcus-Hush-Levich[3]" or "Marcus-Hush-Chidsey", as Chidsey was the first to show experimentally the validity of the model, by measuring the rates of electron transfer across thiol layers[4].

URL: `vincent.fourmond@imm.cnrs.fr` (V. Fourmond*)

The MDoS theory states that the rates of electron transfer in reaction (1) can be written under the following form[3]:

$$k_{\mathrm{ox}}(E) = A \int_{-\infty}^{\infty} \frac{\exp\left[-\frac{RT}{4F\lambda}\left\{\frac{F}{RT}\left[\lambda - (E-E^0)\right] - \xi\right\}^2\right]}{1+\exp\xi}\,\mathrm{d}\xi \tag{3a}$$

$$k_{\mathrm{red}}(E) = A \int_{-\infty}^{\infty} \frac{\exp\left[-\frac{RT}{4F\lambda}\left\{\frac{F}{RT}\left[\lambda + (E-E^0)\right] - \xi\right\}^2\right]}{1+\exp\xi}\,\mathrm{d}\xi \tag{3b}$$

in which E is the electrode potential (the Fermi level of the electrode), E^0 is the standard potential of the Ox/Red couple, λ is the reorganization energy, F the Faraday constant, R the gas constant, T the temperature, and A a prefactor. The prefactor can be put into two forms, depending on whether one wishes to focus on rates at infinite or zero driving force:

$$A = \frac{k_\infty}{\sqrt{4\pi F\lambda/RT}} \tag{4}$$

$$A = \frac{k_0}{\displaystyle\int_{-\infty}^{\infty} \frac{\exp\left[-\frac{RT}{4F\lambda}\left\{\frac{F\lambda}{RT} - \xi\right\}^2\right]}{1+\exp\xi}\,\mathrm{d}\xi} \tag{5}$$

Indeed, with (4), $\lim_{E\to\infty} k_{\mathrm{ox}}(E) = k_\infty$, while with (5), $k_{\mathrm{ox}}(E_0) = k_{\mathrm{red}}(E_0) = k_0$. The equations relating k_∞ or k_0 to the molecular details of the interaction between the electrode and the redox species are out of the scope of this paper and discussed elsewhere[3, 5–7].

The main issue with the formulas in (3) is that there is no closed-forms expression for this integral; a number of algorithms have been published to compute the rates, varying in their approaches and goals in terms of accuracy/computation time trade off. However, it is not easy to compare their merits.

The purpose of this article is to review the numerical methods that have been proposed to compute the integrals in (3), implement them in a publicly available computer program (written in C), and compare them both in terms of accuracy and computing time. Both are particularly important, for example when integrating differential equations that involve MDoS rates. The aim of this article is to arm the reader with various read-to-use tools they could use in their own computations, making informed choices based on their accuracy and cost. We also provide a new algorithm (`ts1` below), which gives single precision accuracy at remarkable speed.

2. The Marcus-Hush-Chidsey integral, and the methods to evaluate it

In this paper, we focus on the evaluation of the integrals in equations (3). Two properties simplify the evaluation. First of all, the expressions in (3) follow the de Donder relationships (i.e. they are compatible with the Nernst equation):

$$\frac{k_{\mathrm{ox}}(E)}{k_{\mathrm{red}}(E)} = \exp\frac{F\left(E-E^0\right)}{RT} \tag{6}$$

In addition, it can be shown that the rate constants of oxidation and reduction are symmetric of one another around the standard potential of the couple:

$$k_{\mathrm{ox}}(E^0+\delta E) = k_{\mathrm{red}}(E^0-\delta E) \tag{7}$$

This relationship only holds for the symmetric case, i.e. when the reorganization energy is the same for both the oxidation and the reduction processes. This is not necessarily the case, see below for more information. The rest of the article assumes that equation (7) holds. Under this assumption, it is only necessary to concentrate our efforts on the oxidation rate constant for potentials greater than the standard potential, as the other values can all be deduced from equations (6) and (7).

We rewrite (3a) into a dimensionless form:

$$\kappa(\eta,\Lambda) = \int_{-\infty}^{\infty} \frac{\exp\left[-\frac{(\Lambda-\eta-\xi)^2}{4\Lambda}\right]}{1+\exp\xi}\,\mathrm{d}\xi \tag{8}$$

in which we removed the prefactor A, and used the following dimensionless variables:

$$\eta = \frac{F\left(E-E^0\right)}{RT} \qquad \Lambda = \frac{\lambda}{RT} \tag{9}$$

η is the dimensionless overpotential and Λ the dimensionless reorganisation energy. A dimensional value of $\lambda = 1$ eV corresponds to a value of $\Lambda = 38.9$ at 298 K. Similarly, an overpotential of 1 V corresponds to $\eta = 38.9$ at 298 K.

The purpose of this article is to discuss the different ways to compute or approximate $\kappa(\eta,\Lambda)$. In the light of the above symmetry considerations, we restrict ourselves to the region $\eta > 0$, without loss of generality. This region is numerically the easiest, since the values of the rate constants are not exceedingly small, unlike for $\eta < 0$.

2.1. Direct computation of the integral

Conceptually, the simplest way to compute $\kappa(\eta,\Lambda)$ is to evaluate the integral using trapezoid rules or more advanced integration techniques. For $\eta > \Lambda$, the integrand of equation (8) is bell-shaped, symmetric, centered on $\xi = \Lambda - \eta$ and has a width of about $\sqrt{\Lambda}$. For $\eta < \Lambda$, the integrand is centered around $\xi = 0$, is more asymmetric and narrower (see [8] for a more precise evaluation of the position of the peak).

As the integrand of (8) decreases very fast, the indefinite integral can in practice be replaced by a finite integral over an interval centered on the peak and whose size is a small number of peak widths.

2.2. Exact summation formula

By introducing the hyperbolic secant function $\operatorname{sech}\xi$ into the integrand and using the development of $\operatorname{sech}\xi$ in powers of $\exp\xi$, Oldham and Myland showed that equation (8) can be written under the form of the following series[9]:

$$\begin{aligned}\kappa(\eta,\Lambda) = \sqrt{\pi\Lambda}\times\exp-\frac{(\Lambda-\eta)^2}{4\Lambda}\times \\ \sum_{i=0}^{\infty}(-1)^i\left\{\operatorname{eerfc}\left[\sqrt{\Lambda}\left(\frac{2i+1}{2}+\frac{\eta}{2\Lambda}\right)\right]+\right. \\ \left.\operatorname{eerfc}\left[\sqrt{\Lambda}\left(\frac{2i+1}{2}-\frac{\eta}{2\Lambda}\right)\right]\right\}\end{aligned} \tag{10}$$

in which the function $\operatorname{eerfc} y$ is defined by equation (11) where $\operatorname{erfc} y$ is the complementary error function:

$$\operatorname{eerfc} y = \exp y^2 \times \operatorname{erfc} y \tag{11}$$

As they noted themselves, this series is exact, but very slowly converging, especially for low η. Therefore, they principally used this representation to derive a number of properties of the integral, including exact values when η is an integer multiple of Λ, and approximations at low and high η[10].

Other authors have used various strategies to accelerate the convergence of series (10), notably Migliore and Nitzan[11, 12], and Bieniasz[13]. In both cases, the series take the following form:

$$\begin{aligned}\sqrt{\pi\Lambda}\times\exp-\frac{(\Lambda-\eta)^2}{4\Lambda}\times \\ \sum_{i} w_i\left\{\operatorname{eerfc}\left[\sqrt{\Lambda}\left(\frac{2i+1}{2}+\frac{\eta}{2\Lambda}\right)\right]+\right. \\ \left.\operatorname{eerfc}\left[\sqrt{\Lambda}\left(\frac{2i+1}{2}-\frac{\eta}{2\Lambda}\right)\right]\right\}\end{aligned} \tag{12}$$

in which w_i is a weight factor. For the original series, (10), $w_i = (-1)^i$. In the work of Migliore and Nitzan[12], for a total number of terms n, the coefficient w_i is given by:

$$w_i = \frac{1}{2}\times(-1)^i\times\sum_{j=i}^{n}\frac{1}{2^j}\binom{j}{i} \tag{13}$$

The series derived by Bieniasz contains only 22 terms, and the value of the coefficients are given in the original publication[13]. Mamedov, using the binomial development of $1/(1+y)$, also obtained similar series, but their convergence properties appear inferior to those described above[14].

2.3. *Step function approximations*

Other groups have derived simpler, approximate formulas for computing (8). For instance, Finklea[15] derived polynomial expressions for $\ln\kappa(\eta,\Lambda)$ for a number of values of Λ by simply fitting low-order polynomials to values of $\ln\kappa(\eta,\Lambda)$ computed using numerical integration. However, since this approach does not provide means to compute $\kappa(\eta,\Lambda)$ for arbitrary values of Λ, we will not consider it further here.

The other approaches are based on the decomposition of the integrand of equation (8) in two parts:

$$\frac{\exp\left[-\frac{(\Lambda-\eta-\xi)^2}{4\Lambda}\right]}{1+\exp\xi} = f(\xi,\eta,\Lambda)\times g(\xi) \tag{14}$$

with f a Gaussian and g the Fermi-Dirac distribution:

$$f(\xi,\eta,\Lambda) = \exp\left[-\frac{(\Lambda-\eta-\xi)^2}{4\Lambda}\right] \tag{15}$$

$$g(\xi) = \frac{1}{1+\exp\xi} \tag{16}$$

For large values of Λ, the function f varies more slowly than g, which varies from ≈ 1 for $\eta<0$ to ≈ 0 for $\eta>0$. Therefore, a first approximation is simply to replace g by a step function, which gives[16]:

$$\kappa(\eta,\Lambda) = \int_{-\infty}^{\infty} f(\xi,\eta,\Lambda)\times g(\xi)\,\mathrm{d}\xi \tag{17}$$

$$\approx \int_{-\infty}^{0} f(\xi,\eta,\Lambda)\,\mathrm{d}\xi \tag{18}$$

$$\approx \sqrt{\pi\Lambda}\,\mathrm{erfc}\,\frac{\Lambda-\eta}{2\sqrt{\Lambda}} \tag{19}$$

Two approaches have been used to improve the accuracy. Using a well-established technique for evaluating the product of Fermi-Dirac distribution with smooth functions, Nahir derived the following improved formula[8]:

$$\kappa(\eta,\Lambda) \approx \sqrt{\pi\Lambda}\,\mathrm{erfc}\,\frac{\Lambda-\eta}{2\sqrt{\Lambda}} + \frac{\pi^2}{12}\frac{\Lambda-\eta}{\Lambda}\exp-\frac{(\Lambda-\eta)^2}{4\Lambda} \tag{20}$$

Zeng and coworkers proposed another formula, obtained by extending the formula to $\eta<0$ and modifying the argument so that the final function is continuous[17]:

$$\kappa(\eta,\Lambda) \approx \frac{\sqrt{\pi\Lambda}}{1+\exp-\eta}\,\mathrm{erfc}\,\frac{\Lambda-\sqrt{1+\sqrt{\Lambda}+\eta^2}}{2\sqrt{\Lambda}} \tag{21}$$

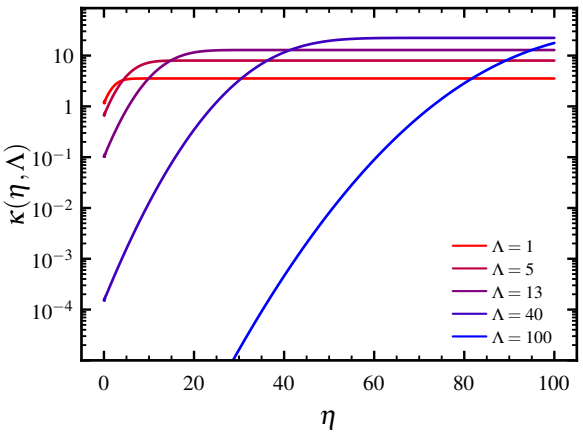

Figure 1: Plot of the function $\kappa(\eta,\Lambda)$ as a function of η for a number of values of Λ (values obtained from the `qag` algorithm). The results demonstrate the typical exponential increase for $\eta \ll \Lambda$ and the saturation for $\eta \gg \Lambda$.

3. Comparison of the methods

We have implemented the formulas described in the previous section in a C program and we have used them to generate a large number of values of $\kappa(\eta,\lambda)$. Table 1 lists the algorithms, along with the names of the file generated by the C program and the time it takes to generate the file[18].

For the numerical integration of (8), we have used the adaptive Gauss-Kronrod finite integration techniques from the Quadpack package[19] as implemented in the GNU Scientific Library [20] (the dataset is called `qag` in table 1), and simple trapezoid implementations. For the latter, we have used two different step sizes, $h=1$ and $h=1/4$, and tried a simple, non-optimized calculation (dataset `tn1` for step $h=1$). We have also rearranged the terms allowing a faster implementation, see Appendix A (datasets `ts1` and `ts4`).

For the summation formulas based on equation (10), we have used 50 terms for the original Oldham and Myland series[10] (dataset `oldham`). Bieniasz's series only has 22 terms (dataset `bieniasz`), so we have chosen to compare the other accelerated series, that of Migliore and Nitzan, using the same number of terms (dataset `migliore`).

We implemented the three "step function approximations" described above; they are datasets `step`, `zeng` and `nahir`.

To compare the various algorithms, we have taken a systematic approach, and computed κ for values of η and Λ in the range $0.01<\eta<100$ and $1<\Lambda<100$, which correspond, at room temperature, to overpotentials of up to 2.5 V and reorganisation energies of up to 2.5 eV. We have used a Maple program to compute the reference values to a relative precision of 10^{-16} (see details in Supplementary section S1; the reference data are avalailable as supplementary files). Figure 1 shows the value of $\kappa(\eta,\Lambda)$ computed by the reference implementation. They show the saturation behaviour for $\eta \geq \Lambda$ that is typical of the MDoS theory and sets it apart from the Butler-Volmer theory.

Figure 2 shows the maximum relative error as a function of Λ for all algorithms (the numbers on the right indicate the rel-

Name	Eq.	Ref.	Time	Notes	Relative time
`qag`	(8)	4, 19	4900 ms	adaptive Gauss-Kronrod integration	117
`oldham`	(10)	10	1850 ms	50 terms	44
`migliore`	(12) and (13)	12	900 ms	22 terms	22
`bieniasz`	(12)	13	900 ms	full summation (22 terms)	22
`tn1`	(A.1)		1000 ms	naive trapezoids, step $h = 1$	25
`ts4`	(A.3)		500 ms	optimized trapezoids, step $h = 1/4$	12
`ts1`	(A.3)		150 ms	optimized trapezoids, step $h = 1$	3.6
`step`	(19)	16	42 ms		1
`nahir`	(20)	8	47 ms		1.1
`zeng`	(21)	17	45 ms		1.1

Table 1: The various algorithms compared in this study, together with the time it took to compute the points used for comparison. The name in the first column is also the name of the data file generated by the C program. The last column is the time relative to the fastest implementation, `step`. The reference data took about 100 hours of computation time to generate.

ative computation times for generating the whole dataset). The errors are plotted as a function of both Λ and η in supplementary information (figures S1 to S9).

According to figure 2, the different algorithms can be roughly divided in three groups. The simple approximations `step`, `zeng` and `nahir`, together with the `oldham` series are in the upper part of the graph, with maximal errors in the order of 1% to 70%.

The second group corresponds to `migliore` series and the `ts1` trapezoid, with maximum errors in the range 10^{-8} to 3×10^{-7} (which corresponds to single precision).

The last group corresponds to the `bieniasz` series, the `ts4` trapezoids, and the Quadpack `qag` numerical integration procedure, with maximum errors below 10^{-11}. The 22-terms series of Bieniasz[13], achieves a 10^{-13} accuracy. The `ts4` dataset, although significantly worse overall than `bieniasz`, achieves maximum errors between 10^{-12} and 10^{-11}. The `qag` numerical integration procedure is on the whole more precise than the other two, but have "spikes" corresponding to a few points where the integration procedure does not give satisfactory results.

In terms of performance (table 1), the three "simple formula approximation" (`step`, `zeng` and `nahir`) perform similarly. Then comes the smart trapezoid with step 1, `ts1`, at only about 3.6 times the computational cost, the trapezoid with step $1/4$, `ts4`, at 12 times. The 22-terms summation formulas `migliore` and `bieniasz` are 22 times slower, and the non-optimized trapezoids `tn1` 25. Finally the slowest methods, the original 50 terms oldham series `oldham` and the quadpack integrations `qag` are between 40 and 120 times slower than the simple approximations.

Not unsurprisingly, the approximations given by equations (19), (20) and (21) are significantly worse than the direct integration, although for all but the smallest values of Λ, the approximation by Nahir[8] has overall better accuracy than the formula proposed by Zeng and coworkers[17]. The simple step function approximation of equation (19) appears on the whole worse than the more elaborate approximations. We have compared the three approximations for selected values of Λ in fig-

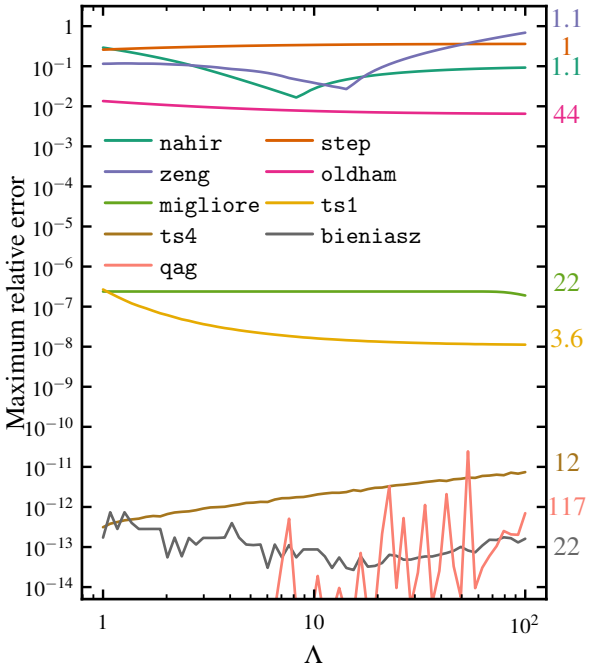

Figure 2: Maximum of the absolute value of the relative error $(|(\kappa - \kappa_{\text{ref}})/\kappa_{\text{ref}}|$, in which κ_{ref} is a reference value computed to a relative precision of 10^{-16}, see Supplementary Section S1) over the range $0.01 \leq \eta \leq 100$ for all the algorithms in table 1. The parts that are not visible correspond to maximum precision better than 10^{-14}. We have not plotted the `tn1` data set, because it is identical to `ts1`. The colored number on the right is the relative computation time, deduced from table 1.

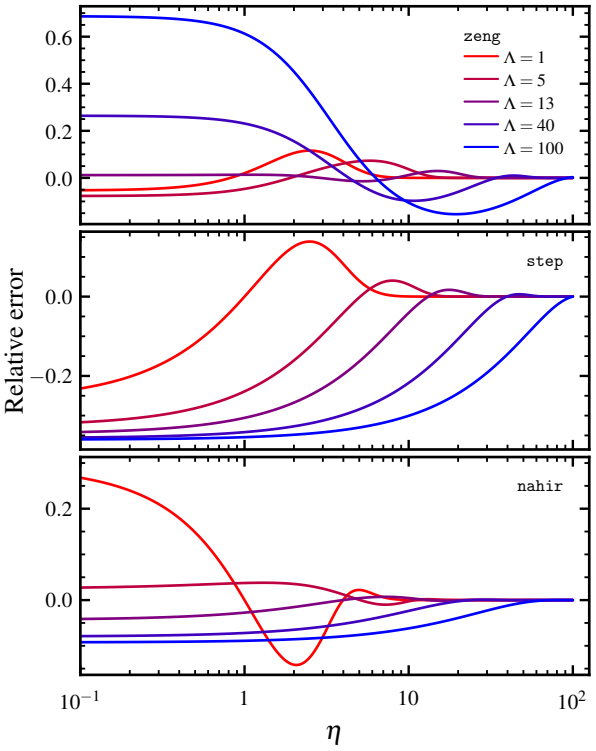

Figure 3: Relative errors with respect to the reference value as a function of η for the three different approximations, for different values of Λ. Data sets, from top to bottom: `zeng`, `step`, `nahir`.

ure 3. The graphs show that, for all the three approximations, the errors are all localized in the low η region. However, as Λ increases, the range in which the approximation has important errors increases to larger values of η than at lower Λ; this is true for all three approximations, but is particularly visible in the `step` approximation (these features are even more clear in supplementary figures S5–7).

The top panel of figure 3 shows in particular that the approximation of equation (21) does a particularly bad job at representing the low η region at high Λ, with relative errors in excess of 60% at $\Lambda = 100$.

We also used our computations to compare the convergence of the series based on equation (12). Figure 4 represents the relative accuracy of the sum over $1 \leq i \leq n$ of the various series as a function of n[21], for two pairs of values of (η, Λ). Unsurprisingly, the original series by Oldham and Myland converges extremely slowly; a log/log plot (Supplementary figure S10) shows that the accuracy scales as n^{-1}. The `migliore` series is much better, with an error that decreases exponentially with the number of terms, until it reaches a value a bit larger than the machine precision, probably because of error accumulation. The errors for the `bieniasz` series decrease slowly at first, and then abruptly go to 10^{-15} when the series is complete with the 22 terms. This is due to the finite nature of the minimax approximation used by Bieniasz[13]: the minimax polynomials are not meant as series that converge towards the target function, but as approximations with a fixed order [22]. Removing any term leads to drastic changes in accuracy.

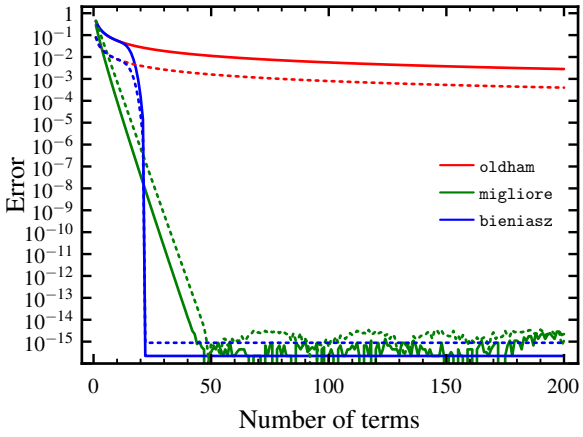

Figure 4: Relative error as a function of the number of terms for the different series based on equation (12). Solid lines: $\Lambda = \eta = 1$; dashed lines: $\Lambda = \eta = 50$.

4. Discussion

We have compared several different ways to compute the MDoS integral in equation (8). We have compared the values for the different algorithms for a large range of values of η and Λ, in a much more systematic fashion than previously reported. We also provide the C code of the program in order for others to assess the formulas against their own needs and directly reuse the code for their own implementations.

The fastest implementations are those based on "simple formulas". They have similar computation times, but are not very accurate, especially at low η and high Λ values. This could lead to important accuracy problems when computing Tafel plots (like in ref 17 for instance), as the lower overpotential region would be misrepresented, especially at high Λ. Among the three methods, the one derived by Nahir (equation (20)) appears to be the most accurate over the largest range.

Many have used trapezoid integrations to compute values of the integral[23–26]. Our results show that even with relatively large step sizes, such as $h = 1$, the accuracy is excellent, better than 10^{-7} over the whole range. Dividing the step size by 4 yields accuracies better than 10^{-10}. This seems at first counter-intuitive, since in general the accuracy for the trapezoid rule scales as the square of the step size: one would expect a precision of roughly 10^{-8}. However, integral (8) falls under the case of the so-called "exponentially convergent trapezoidal rule[27]". Theorem 5.1 in ref. 27 states that one can expect exponential convergence of the trapezoid sum as a function of the reciprocal of the step size for an integral over the real line of a function analytical over a stripe around the real line. This theorem applies since the integrand of equation (8) is analytical over the stripe:

$$|\mathrm{Im}(z)| < \pi \tag{22}$$

Under these conditions, the theorem in [27] predicts an exponential decrease of the error:

$$|\Delta\kappa| < \frac{\alpha}{\exp\left(\frac{2\pi a}{h}\right) - 1} \tag{23}$$

in which $\Delta\kappa$ is the error, α a constant, h the step size and a a constant close to π. This explains the excellent precision obtained by the trapezoid formula.

We have designed a very fast implementation of the trapezoids, a factor of 8 times faster than the "naive" one (compare the computation times for `tn1` and `ts1`), which yields single precision values (for $h = 1$) only 3.6 times slower than the simple formulas.

For the most precise evaluations, the series proposed by Bieniasz[13], based on the series of Oldham and Myland[10] perform extremely well, reaching 10^{-13} relative accuracy with only 22 times the cost of the simplest formulas, greatly outperforming the similar series proposed by Migliore and Nitzan[12]. The original work by Bieniasz[13] showed that the accuracy of the series should be close to 10^{-16}, the differences here may be due to the particular implementation of the eerfc$y$ function, or to accumulation of errors. The numerical integration by the QUADPACK algorithms does not appear to be a good choice, because of the long computation time and the precise of isolated points with errors significantly larger than the average precision (see supplementary figure S9).

Based on the computation speed, we suggest the use of `ts1` when only simple-precision accuracy is enough, and that of `bieniasz` for double-precision.The accuracy of both should be enough for numerical differentiation. We have implemented both methods in our open-source data analysis software QSoas[28] (`qsoas.org`), together with the approximations of Zeng and of Nahir.

In spite of a number of experimental sucesses[4], a growing number of experimental data pinpoint the inability of the MDoS to describe experimental data[15, 25, 29]. This lies in the simplifying assumption of equal force constants for reactants and products in the drawing of the Marcus energy parabolas[30]. Taking this asymmetry into consideration leads to the so-called "asymmetric Marcus-Hush-Chidsey" or "asymmetric MDoS" theory, which was found to be more appropriate to describe a number of experimental results[15, 25, 29–31]. Although the present article only deals with the symmetric MDoS electron transfer rates, Zeng and coworkers have shown that, under a large range of conditions, the asymmetric MDoS rates can be obtained from the symmetric ones by multiplying by a simple factor[32]. The results herein can therefore easily be extended to the asymmetric MDoS computations. We therefore hope that the algorithms described here can also help using asymmetric MDoS rates.

Acknowledgments

The authors acknowledge support from CNRS, Agence Nationale de la Recherche (grants ANR-14-CE05-0010, ANR-15-CE05-0020, ANR-17-CE11-002) and Région PACA. The project leading to this publication has received funding from Excellence Initiative of Aix-Marseille University - A*Midex, a French "Investissements d'Avenir" programme. The authors are members of the French Bioinorganic Chemistry group (`http://frenchbic.cnrs.fr`).

Supplementary data

The source code for the program generating the all the data analyzed in the article is available as supplementary data. The precision of all the algorithms as a function of both η and Λ is available as supplementary figures S1 to S8.

References

[1] M. C. Henstridge, E. Laborda, N. V. Rees, R. G. Compton, Marcus–Hush–Chidsey theory of electron transfer applied to voltammetry: A review, Electrochim. Acta 84 (2012) 12–20. doi:10.1016/j.electacta.2011.10.026.

[2] R. A. Marcus, Chemical and electrochemical electron-transfer theory, Annual Review of Physical Chemistry 15 (1) (1964) 155–196. doi:10.1146/annurev.pc.15.100164.001103.

[3] J. M. Savéant, C. Costentin, Elements of molecular and biomolecular electrochemistry, 2nd Edition, John Willey & sons, Inc., 2019.

[4] C. E. Chidsey, Free energy and temperature dependence of electron transfer at the metal-electrolyte interface, Science 251 (4996) (1991) 919–922. doi:10.1126/science.251.4996.919.

[5] S. W. Feldberg, Implications of Marcus-Hush theory for steady-state heterogeneous electron transfer at an inlaid disk electrode., Anal. Chem. 82 (12) (2010) 5176–5183. doi:10.1021/ac1004162.

[6] M. Honeychurch, Effect of electron-transfer rate and reorganization energy on the cyclic voltammetric response of redox adsorbates, Langmuir 15 (15) (1999) 5158–5163. doi:10.1021/la990169u.

[7] J.-M. Savéant, Effect of the electrode continuum of states in adiabatic and nonadiabatic outer-sphere and dissociative electron transfers. use of cyclic voltammetry for investigating nonlinear activation-driving force laws, J. Phys. Chem. B 106 (36) (2002) 9387–9395. doi:10.1021/jp0258006.

[8] T. M. Nahir, On the calculation of rate constants by approximating the Fermi–Dirac distribution with a step function, Journal of Electroanalytical Chemistry 518 (1) (2002) 47–50. doi:10.1016/S0022-0728(01)00688-X.

[9] The exact series used by Oldham and Myland is slightly different because they are computing a slightly different integral, but this formula was derived using their approach[10].

[10] K. B. Oldham, J. C. Myland, On the evaluation and analysis of the Marcus–Hush–Chidsey integral, Journal of Electroanalytical Chemistry 655 (1) (2011) 65–72. doi:10.1016/j.jelechem.2011.01.044.

[11] A. Migliore, A. Nitzan, Nonlinear charge transport in redox molecular junctions: a Marcus perspective., ACS Nano 5 (8) (2011) 6669–6685. doi:10.1021/nn202206e.

[12] A. Migliore, A. Nitzan, On the evaluation of the Marcus–Hush–Chidsey integral, Journal of Electroanalytical Chemistry 671 (2012) 99–101. doi:10.1016/j.jelechem.2012.02.026.

[13] L. K. Bieniasz, A procedure for rapid and highly accurate computation of Marcus-Hush-Chidsey rate constants, Journal of Electroanalytical Chemistry 683 (2012) 112–118. doi:10.1016/j.jelechem.2012.08.015.

[14] B. A. Mamedov, Analytical evaluation of the Marcus–Hush–Chidsey function using binomial expansion theorem and error functions, Journal of Mathematical Chemistry 51 (10) (2013) 2699–2703. doi:10.1007/s10910-013-0231-y.

[15] H. O. Finklea, Consequences of a potential-dependent transfer coefficient in AC voltammetry and in coupled electron–proton transfer for attached redox couples, Journal of Electroanalytical Chemistry 495 (2) (2001) 79–86. doi:10.1016/S0022-0728(00)00399-5.

[16] J. Hale, The potential-dependence and the upper limits of electrochemical rate constants, J. Electroanal. Chem. Interfacial Electrochem. 19 (1968) 315–318. doi:10.1016/S0022-0728(68)80131-7.

[17] Y. Zeng, R. B. Smith, P. Bai, M. Z. Bazant, Simple formula for Marcus-Hush-Chidsey kinetics, J. Electroanal. Chem. 735 (0) (2014) 77–83. doi:10.1016/j.jelechem.2014.09.038.

[18] The times were measured using a single core of and Intel Xeon CPU E5620 running at 2.40GHz under Linux. The C program was compiled by `gcc` version 6.2.1, using the `-O3` optimization option. We report them only to compare the relative performances of the various algorithms.

[19] R. Piessens, E. de Doncker-Kapenga, C. W. Überhuber, D. K. Kahaner, Quadpack, Vol. 1 of Springer Series in Computational Mathematics, Springer Berlin Heidelberg, 1983, p. 304.
[20] M. Galassi, J. Davies, J. T. B. Gough, G. Jungman, P. Alken, M. Booth, F. Rossi, GNU Scientific Library Reference Manual, Network Theory Ltd., 2003.
[21] Note that, since the terms are very similar, they take the same time to compute, so this is also a plot of the accuracy as a function of computation time.
[22] W. Fraser, A survey of methods of computing minimax and near-minimax polynomial approximations for functions of a single independent variable, Journal of the ACM (JACM) 12 (3) (1965) 295–314. doi:10.1145/321281.321282.
[23] H. A. Heering, J. Hirst, F. A. Armstrong, Interpreting the catalytic voltammetry of electroactive enzymes adsorbed on electrodes, J. Phys. Chem. B 102 (35) (1998) 6889–6902. doi:10.1021/jp981023r.
[24] J. Hirst, F. Armstrong, Fast-scan cyclic voltammetry of protein films on pyrolytic graphite edge electrodes: Characteristics of electron exchange, Anal. Chem. 70 (23) (1998) 5062–5071. doi:10.1021/ac9805571.
[25] M. C. Henstridge, E. Laborda, R. G. Compton, Asymmetric Marcus–Hush model of electron transfer kinetics: Application to the voltammetry of surface-bound redox systems, J. Electroanal. Chem. 674 (2012) 90–96. doi:10.1016/j.jelechem.2012.04.006.
[26] K. Weber, S. E. Creager, Voltammetry of redox-active groups irreversibly adsorbed onto electrodes. treatment using the Marcus relation between rate and overpotential, Analytical Chemistry 66 (19) (1994) 3164–3172. doi:10.1021/ac00091a027.
[27] L. N. Trefethen, J. A. C. Weideman, The exponentially convergent trapezoidal rule, SIAM Review 56 (3) (2014) 385–458. doi:10.1137/130932132.
[28] V. Fourmond, QSoas: a versatile software for data analysis, Anal. Chem. 88 (10) (2016) 5050–5052. doi:10.1021/acs.analchem.6b00224.
[29] D. Suwatchara, N. V. Rees, M. C. Henstridge, E. Laborda, R. G. Compton, Experimental comparison of the Butler-Volmer and Marcus-Hush-Chidsey formalisms of electrode kinetics: The reduction of cyclooctatetraene at mercury hemispherical electrodes via cyclic and square wave voltammetries, Journal of Electroanalytical Chemistry 665 (2012) 38–44. doi:10.1016/j.jelechem.2011.11.009.
[30] E. Laborda, M. C. Henstridge, R. G. Compton, Asymmetric Marcus theory: application to electrode kinetics, Journal of Electroanalytical Chemistry 667 (2012) 48–53. doi:10.1016/j.jelechem.2011.12.011.
[31] E. Laborda, M. C. Henstridge, C. Batchelor-McAuley, R. G. Compton, Asymmetric Marcus–Hush theory for voltammetry, Chem. Soc. Rev. 42 (12) (2013) 4894–4905. doi:10.1039/c3cs35487c.
[32] Y. Zeng, P. Bai, R. B. Smith, M. Z. Bazant, Simple formula for asymmetric Marcus–Hush kinetics, Journal of Electroanalytical Chemistry 748 (2015) 52–57. doi:10.1016/j.jelechem.2015.04.018.

Appendix A. Efficient computation of the trapezoid formula

Here, we derive the formula used in the C code to compute the trapezoid integral:

$$S_n = \sum_{i=-n}^{n} \frac{\exp\left[-\frac{(\Lambda-\eta-\xi_i)^2}{4\Lambda}\right]}{1+\exp\xi_i} \tag{A.1}$$

with:

$$\xi_i = \Lambda - \eta + ih \tag{A.2}$$

Thus:

$$S_n = \sum_{i=-n}^{n} \frac{\exp\left[-\frac{(ih)^2}{4\Lambda}\right]}{1+\exp(\eta-\Lambda)\times\exp ih} = \sum_{i=-n}^{n} \frac{\alpha^{i^2}}{1+\gamma\times\beta^i} \tag{A.3}$$

with:

$$\alpha = \exp\left[-\frac{h^2}{4\Lambda}\right] \qquad \beta = \exp h \qquad \gamma = \exp(\Lambda-\eta) \tag{A.4}$$

One can efficiently compute the α^{i^2} term by expressing it as:

$$\alpha^{i^2} = \alpha^{(i-1)^2} \times \alpha^{2i+1} \tag{A.5}$$

and updating the α^{2i+1} term at each iteration.