Numerical Regge pole analysis of resonance structures in elastic, inelastic and reactive state-to-state integral cross sections

E. Akhmatskaya\textsuperscript{a,c}, D. Sokolovski\textsuperscript{b,c}, and C. Echeverría-Arrondo\textsuperscript{b}

\textsuperscript{a}Basque Center for Applied Mathematics (BCAM),
Alameda de Mazarredo, 14 48009 Bilbao, Bizkaia, Spain
\textsuperscript{b}Department of Physical Chemistry, University of the Basque Country, Leioa, 48940, Spain
\textsuperscript{c}IKERBASQUE, Basque Foundation for Science, E-48011 Bilbao, Spain

Abstract

We present a detailed description of a FORTRAN code for evaluation of the resonance contribution a Regge trajectory makes to the integral state-to-state cross section (ICS) within a specified range of energies. The contribution is evaluated with the help of the Mulholland formula [Macek \textit{et al} (2004)] and its variants [Sokolovski \textit{et al} (2007); Sokolovski and Akhmatskaya (2011)]. Regge pole positions and residues are obtained by analytically continuing $S$-matrix element, evaluated numerically for the physical values of the total angular momentum, into the complex angular momentum plane using the \texttt{PADE} \texttt{I} \texttt{I} program [Sokolovski \textit{et al} (2011)]. The code decomposes an elastic, inelastic, or reactive ICS into a structured, resonance, and a smooth, 'direct', components, and attributes observed resonance structure to resonance Regge trajectories. The package has been successfully tested on several models, as well as the $F + H_2 \rightarrow HF + H$ benchmark reaction. Several detailed examples are given in the text.

PACS:34.50.Lf,34.50.Pi

\textit{Key words:} Atomic and molecular collisions, integral cross sections, resonances, $S$-matrix; Padé approximation; Regge poles.

PROGRAM SUMMARY

\textit{Manuscript Title:} Numerical Regge pole analysis of resonance structures in elastic, inelastic and reactive state-to-state integral cross sections.
\textit{Authors:} E. Akhmatskaya, D. Sokolovski, and C. Echeverría-Arrondo

1 dgsokol15@gmail.com

Preprint submitted to Elsevier 19 May 2016
Program Title: ICS_Regge
Journal Reference: 
Catalogue identifier: 
Licensing provisions: Free software license
Programming language: FORTRAN 90
Computer: Any computer equipped with a FORTRAN 90 compiler
Operating system: UNIX, LINUX
RAM: 256 Mb
Has the code been vectorised or parallelised: no
Number of processors used: one
Supplementary material: PADE_II, MPFUN and QUADPACK packages, validation suites, script files, input files, readme files, Installation and User Guide
Keywords: Atomic and molecular collisions, integral crioss sections, resonances, S-matrix; Padé approximation; Regge poles
PACS: 34.50.Lf,34.50.Pi
Classification: Molecular Collisions
External routines/libraries: none
CPC Program Library subprograms used: N/A
Nature of problem: The package extracts the positions and residues of resonance poles from numerical scattering data supplied by the user. This information is then used for the analysis of resonance structures observed in elastic, inelastic and reactive integral cross sections.
Solution method: The S-matrix element is analytically continued in the complex plane of either energy or angular momentum with the help of Padé approximation of type II. Resonance Regge trajectories are identified and their contribution to the integral cross section are evaluated.
Restrictions: None.
Unusual features: Use of multiple precision MPFUN package.
Additional comments: none
Running time: from several minutes to hours depending on the number of energies involved, the precision level chosen and the number of iterations performed.
References:
1 Introduction

In the last fifteen years the progress in crossed beams experimental techniques has been matched by the development of state-of-the-art computer codes capable of modelling atom-diatom elastic, inelastic and reactive differential and integral cross sections (ICS) [1]-[10]. The ICS, accessible to measurements in crossed beams, are often structured, and so offer a large amount of useful information about details of the collision or reaction mechanism. This information needs to be extracted and analysed, which often presents a challenging task. One distinguishes two main types of collisions: in a direct collision the partners depart soon after the first encounter, while in a resonance collision they form an intermediate complex (quasi-molecule) which then breaks up into products (reactive case) or back into reactants (elastic or inelastic case). The resonance pathways may become important or even dominant at low collision energies. For this reason, accurate modelling and understanding of resonance effects gains importance in such fields as cold atom physics and chemistry of the early universe.

Once the high quality scattered matrix is obtained numerically, one needs to understand the physics of the reaction, often not revealed until an additional analysis is carried out. In particular, resonances invariably leave their signatures on the integral state-to-state cross sections. In this paper we propose and describe software for the analysis of such resonance patterns. Relevant information on the Regge poles can be found in Refs.[11]-[16]. Some applications of the poles to the angular scattering and integral cross sections are discussed in Refs.[17]-[36] and [37]-[42], respectively. For a description of the type-II Padé approximation, used by the software, the reader is referred to Refs.[43]-[50].

2 Background and theory

We start with a brief review of the techniques required for our analysis.

2.1 The integral, or total, scattering cross section

An integral (total) cross section, \( \sigma(E) \), gives the total number of incident particles with the energy \( E \), scattered in all possible directions per unit time, for unit incoming flux. This definition is valid for a single particle scattered by a central force, as well as for an atom \( A \) colliding with a diatomic molecule
In the latter case, the collision partners may part in the same arrangement $(A + BC)$, with the molecule in the same internal state described by the vibrational $(v)$, rotational $(j)$, and helicity $(\Omega)$ quantum numbers. This is elastic scattering.

In an inelastic scattering event, the arrangement $(A + BC)$ remains the same, yet the internal quantum numbers of the molecule $BC$ are changed, $(v, j, \Omega \rightarrow v', j', \Omega')$.

Finally, in a reactive event the atom $B$ becomes attached to the atom $A$ $(A + BC \rightarrow AB + C)$, and the quantum numbers of the newly formed diatomic may take any values allowed by the conservation laws.

The probability amplitude for each process is defined for an energy $E$ (total, if it includes the internal energy of the reactant diatomic, or collision, $E^{\text{coll}}$, if it does not), and a value of the total angular momentum $J$ (we use $\hbar = 1$). It is given by complex-valued scattering ($S$-) matrix element $S^\alpha_{\nu \leftrightarrow \nu'}(E, J)$, where $\alpha = \text{elastic, inelastic, reactive}$, and $\nu$ is the shorthand for $(v, j, \Omega)$.

In all cases, the integral cross section can be written as a partial wave sum (PWS) over all physical (i.e., integer) values of the total angular momentum. For the elastic channel, where the interference with the incoming wave must be taken into account, one has

$$
\sigma^\alpha_{\nu \leftrightarrow \nu'}(E) = \frac{2\pi}{k^2_\nu} \sum_{J=J_{\text{min}}}^{\infty} (J + 1/2)|S^\alpha_{\nu \leftrightarrow \nu'}(E, J) - 1|^2, \quad \alpha = \text{elastic},
$$

where

$$J_{\text{min}} = |\Omega|,$$

since the projection of the molecule’s angular momentum on the relative velocity cannot exceed the total angular momentum $J$.

For an inelastic or reactive process there is no such interference, and the ICS takes the form

$$
\sigma^\alpha_{\nu \leftrightarrow \nu'}(E) = \frac{2\pi}{k^2_\nu} \sum_{J=J_{\text{min}}}^{\infty} (J + 1/2)|S^\alpha_{\nu \leftrightarrow \nu'}(E, J)|^2, \quad \alpha = \text{inelastic or reactive},
$$

with

$$J_{\text{min}} = \text{max}(|\Omega|, |\Omega'|).$$

In Eqs. (1) and (3) $k_\nu = \sqrt{2\mu E^{\text{coll}}}$ is the reactant’s relative wave vector, which in the case of the single-channel potential scattering becomes the wave vector of the incoming plane wave, $k = \sqrt{2\mu E}$, $\mu$ being the corresponding reduced
Identification and quantitative analysis of structures produced in the ICS by the capture of collision partners into long lived metastable states is the main subject of this paper. Despite their simplicity, Eqs. (1)-(3) are not best suited for the task, since the $S$-matrix elements contain contributions from both the direct and the resonance mechanisms. A more convenient representation, designed to separate the two contributions, was proposed by Macek et al in their study of the proton scattering by hydrogen [37], and later used by Ovchinnikov et al who considered proton impact on inert gas atoms [38]. The method relies on the concept of Regge poles, which we will discuss here only briefly.

2.2 Regge poles and Regge trajectories

Regge poles are the poles of $S$-matrix element $S_{\nu'\rightleftarrows\nu}(E,J)$ in the first quadrant complex plane of the (continuous) variable $J$, evaluated at a fixed energy $E$ [11]. Sharp long-lived resonances manifest themselves as poles at $J = J_n(E)$, $n = 1, 2, ...$, close to the real axis,

$$S_{\nu'\rightleftarrows\nu}(E, J_n) = \infty, \quad \text{Re} J_n > 0, \quad \text{Im} J_n > 0. \quad (5)$$

Resonance Regge poles are closely related to the resonance complex energy poles [34], [14], [40], but are more convenient for the analysis of the quantities given by partial wave sums at a fixed energy, like the ICS in Eqs. (1)-(3). A Regge pole cannot disappear suddenly (except by coalescing with a complex zero of the $S$) and, as the energy changes, traces a continuous curve in the complex angular momentum (CAM) plane. These curves are known as Regge trajectories.

It is useful to distinguish between at least two types of Regge trajectories [15]: those which at zero angular momentum correspond to a bound state supported by the potential (type I), and those corresponding at $J = 0$ to a potential’s metastable state (type II). Regge trajectories [curves $\text{Im} J_n(E)$ vs. $\text{Re} J_n(E)$] of the type (I) typically start, at low energies, on the real $J$-axis, and then move into the complex $J$-plane in its first quadrant. The trajectories of the type (II) initially descend on the real $J$-axis, and affect an ICS in a different way [15]. Our test cases, provided below, contain examples of each type.

2.3 The Mulholland formula

Macek et al [37], [38] replaced the discrete sum over $J$ in Eq. (1) by an integral, and then deformed the contour of integration to run along the imaginary $J$-axis, thereby picking contributions from the Regge poles in the first quadrant.
The result is the decomposition

$$
\sigma_{\text{elastic}}(E) = \frac{2\pi}{k^2} \int_{0}^{\infty} f(E, \lambda) \lambda d\lambda - \text{Re} \int_{0}^{\infty} \frac{4\pi f(E, \lambda)/k^2}{1 + \exp(-2i\pi\lambda)} \lambda d\lambda - \frac{2\pi}{k^2} \text{Re} \sum_{n} \frac{4\pi i \text{Res}[f(E, \lambda_n)]\lambda_n}{1 + \exp(-2i\pi\lambda_n)}
$$

(6)

where $f(E, \lambda) \equiv |S(E, \lambda) - 1|^2$, $\lambda \equiv J + 1/2$, $\lambda_n = J_n + 1/2$ is the position of the $n$-th pole in the CAM plane, and $\text{Res}_n[f(E, \lambda_n)]$ is the residue at the $n$-th pole,

$$
\text{Res}[f(E, \lambda_n)] \equiv \lim_{\lambda \to \lambda_n} (\lambda - \lambda_n)f(E, \lambda).
$$

(7)

The integrals along the real and imaginary axis are expected to be slowly varying function of the energy, with the resonance structure given mostly by the last sum in Eq.(6). The authors of Refs. (1)-(3) traced their technique back to a 1928 paper by H. P. Mulholland [51] (whose subject, one must say, had very little to do with the present analysis), and we will follow them in calling the result (6) the Mulholland formula.

The practical use of the formula (6) is quite simple: one chooses a particular, say the $n$-th, Regge trajectory, evaluates along it the contribution to the ICS given by the $N$-th term in the sum,

$$
I_{\text{Mull}}^n(E) = -\frac{2\pi}{k^2} \text{Re} \frac{4\pi i \text{Res}[f(E, \lambda_n)]\lambda_n}{1 + \exp(-2i\pi\lambda_n)},
$$

(8)

and then subtracts the result from the full ICS. If what is left is smooth and structureless, one can identify the pattern in the full ICS as arising from this particular resonance trajectory. There are many possible patterns. From (8) one readily sees that a pole contribution is greatly enhanced whenever the trajectory passes in the vicinity of a positive integer, $J_n \approx N > 0$ [37]. This is hardly surprising since one recalls that a true bound state requires an integer value of $J$, and that a sharp resonance is in fact a bound state which is weekly connected to a continuum. Depending on the potential, a trajectory may follow in the vicinity of the real $J$-axis, passing by several integer $J$’s, and thus producing a series of peaks in the ICS [15], [41]. Or it can veer steeply deep into the CAM plane thus producing just one peak, or even no peaks at all [37]. A trajectory following the real axis at a larger distance would produce in the ICS not peaks, but sinusoidal oscillations [31]. A trajectory may start on the real axis, and then move deeper into the CAM plane, or descend to the real $J$-axis, and then follow it for some time. All these possibilities account a large variety of ways in which a resonance may produce a structure in an ICS.
2.4 Modifications of the Mulholland formula

The Mulholland decomposition of the ICS (6) is easily modified to the case of reactive and inelastic transitions as [31] (below $\alpha = \text{inelastic or reactive}$)

\[
\sigma_{\nu'\rightarrow\nu}(E) = \frac{2\pi}{k_{\nu}^2} \int_{J_{\min} - 1/2}^{\infty} |S_{\nu'\rightarrow\nu}(E, \lambda)|^2 \lambda d\lambda + \frac{8\pi^2}{k_{\nu}^2} Im \sum_{n} \frac{\lambda_n \text{Res}[S_{\nu'\rightarrow\nu}(E, \lambda_n)] S_{\nu'\rightarrow\nu}^*(E, \lambda_n^*)}{1 + \exp(-2i\pi\lambda_n)} + I_{\nu'\rightarrow\nu}(E),
\]

where $^*$ denotes complex conjugation, and $I_{\nu'\rightarrow\nu}$ contains integrals along the imaginary $\lambda$-axis, similar to the second term in Eq. (6). In the following we will not be interested in the explicit form of $I_{\nu'\rightarrow\nu}$. Equation (9) is exact, and if necessary, $I_{\nu'\rightarrow\nu}$ can be evaluated by subtracting from $\sigma_{\nu'\rightarrow\nu}(E)$, computed as a PWS, the first two terms in its r.h.s.

For an elastic transition in a multi-channel case, $\alpha = \text{elastic}$, $\nu = \nu'$, one replaces in Eq.(9)

\[
S_{\nu'\rightarrow\nu}(E, \lambda) \rightarrow S_{\nu'\rightarrow\nu}(E, \lambda) - 1.
\]

The formula (9), which reduces to (6) in the single-channel case, is to be used just as described at the end of the previous Subsection, and is employed in our computer code. Previously, it has been applied to analyse resonance structures in the ICS of the benchmark $F + H_2 \rightarrow HF + H$ reaction [31], [40].

Returning to the single-channel case, one notes [15], [41] that the factor

\[
[1 + \exp(-2i\pi\lambda_n)]^{-1} = - \sum_{M=1}^{\infty} \exp(2\pi i M J_n)
\]

is a sum of a geometric progression in which the first term, a unity, is missing. (For a detailed discussion of this point the reader is referred to Ref. [15]) The physical picture is that of an intermediate metastable complex which rotates and decays at the same time. The optical theorem [41] allows one to interpret the terms in the r.h.s of Eq.(11) as the contributions a rotating complex makes to the forward scattering amplitude after $M = 1, 2...$ complete rotations. The missing term in (11) is just what the complex contributes to forward scattering immediately after its formation, before completing even one rotation. Thus, the $M = 0$ term can be subtracted from the first, and added to the second term in Eq.(9). This yields a modified Mulholland formula [15],

\[
\sigma(E) = \frac{2\pi}{k^2} \int_{E} |S(E, \lambda) - 1|^2 \lambda d\lambda + \frac{8\pi^2}{k^2} Im \sum_{n} \frac{\lambda_n \text{Res}[S(E, \lambda_n)]}{1 + \exp(2i\pi\lambda_n)} + I(E),
\]

\[7\]
where the integration contour $\Gamma$ runs in the first quadrant of the CAM above all poles which make significant Mulholland contributions to the ICS. Decomposition (12) was shown to achieve a better separation of the ICS into direct and resonance parts for a simple single-channel model in [15], and we include the modified formula in the present code. However, we have yet been unable to extend its application to the multichannel case, and the option for using (12) remains open to single-channel potential scattering only.

2.5 Padé reconstruction of the scattering matrix element

The application of the decompositions (9)-(12) requires the knowledge of the pole positions $\lambda_n = J_n + 1/2$, and the corresponding residues, $\text{Res}[S(E, \lambda_n)]$. Since the analytic continuation of $|f(E, \lambda)|^2$ from the real axis into the complex $\lambda$-plane is $f(E, \lambda)f^*(E, \lambda^*)$, one also requires the values of $S^*(E, \lambda_n^*)$. These data are not usually available since a computer code used in modelling a chemical reaction typically evaluates the $S$-matrix elements, $S(E, J)$, for the physical integer values of $J = 0, 1, 2, \ldots, N$, with $N$ sufficiently large to converge the partial waves sums (1) and (3). Using these values, we construct a rational Padé approximant, $([x]$ stands for the integer part of $x$)

$$S_{\nu' \leftrightarrow \nu}^{\text{Pade}}(E, J) \equiv K_N \exp[i(aJ^2 + bJ + c)] \times \frac{\prod_{i=1}^{|N/2|} (J - Z_i)}{\prod_{i=1}^{|(N-1)/2|} (J - P_i)}, \quad (13)$$

where $P_i(E)$ and $Z_i(E)$ stand for poles and zeroes of the approximant, respectively, and $K_N(A), a(E), b(E)$ and $c(E)$ are energy dependent constants. The approximant is conditioned to coincide with $S(E, J)$ at the $N$ integer values of $J$,

$$S_{\nu' \leftrightarrow \nu}^{\text{Pade}}(E, J) = S_{\nu' \leftrightarrow \nu}(E, J), \quad J = 0, 1, \ldots, N, \quad (14)$$

and provides an analytic continuation of the exact function $S(E, J)$ in a region of the complex $J$-plane, containing the values (14). Inside this region, the poles and zeroes of the approximant coincide with the true Regge poles and zeroes of the $S$-matrix element. The remaining poles and zeroes tend to mark the border of the region, beyond which Padé approximation fails [47]-[50]. Thus, for a given pole $J_n = P_j$ we evaluate the required quantities as

$$\text{Res}[S_{\nu' \leftrightarrow \nu}(E, \lambda_n)] = K_N \exp[i(aP_j^2 + bP_j + c)] \times \frac{\prod_{i=1}^{|N/2|} (P_j - Z_i)}{\prod_{i=1, i \neq j}^{|(N-1)/2|} (P_j - P_i)}, \quad (15)$$
and

\[ S^\ast(E, \lambda_n^\ast) = K_N^\ast \exp[-i(aP_j^2 + bP_j + c)] \times \frac{\prod_{i=1}^{[N/2]}(P_j - Z_i^\ast)}{\prod_{i=1}^{[(N-1)/2]}(P_j - P_i^\ast)}, \]  

(16)

which provides the data needed for evaluating the resonance contributions in the Mulholland formulas (9) and (12).

The program used to construct the Padé approximant (13) is essentially the PADE II code reported in the Ref. [50]. The minor modifications made to adapt it to the problem in hand will be described below.

### 2.6 A brief summary

Our analysis consists in attributing to particular Regge trajectories resonance structures observed in an elastic, inelastic of reactive ICS obtained by numerical modelling of a scattering process. The behaviour of the scattering matrix element in the complex \( J \)-plane, required for evaluating the contribution a trajectory makes to the ICS, is reconstructed by using Padé approximants of type (II) [50]. The method gives a clear picture of how a resonance affects the ICS in a range of energies. It does not, however, reveal the physical origin of the resonance (e.g., its location in the entrance or exit channel on the potential surface), which must be established independently.

### 3 ICS_Regge package: Overview

#### 3.1 Installation

This version of ICS_Regge is intended for IA32 / IA64 systems running the Linux operating system. It requires Fortran and C compilers. The software is distributed in the form of a gzipped tar file, which contains the ICS_Regge source code, PADE.II 1.1 source code, QUADPACK source code, test suites for each package, as well as scripts for running and testing the code. The detailed structures of each subpackage, ICS_Regge, PADE.II 1.1 and QUADPACK, are presented in the Appendices C, D and E. For users benefits we supply a file README for each package in directories ICS, ICS/PADE and ICS/QUADPACK. The files provide a brief summary on the code structure and basic instructions for users. The ICS_Regge Manual, ICSManual.pdf, is located in ICS/ directory whereas PADE.II Manual, FManual.pdf, can be found in ICS/PADE directory. In addition, the documents describing PADE.II and QUADPACK are available.
from http://www.cpc.cs.qub.ac.uk/ (Catalogue identifier: AEHO_v1.0) and http://www.netlib.org/quadpack/ respectively. Once the package is unpacked the installation should be done in the following order:

1. Installing QUADPACK library

2. Installation of the PADE II package

3. Installation of the ICS Regge package.

Installation procedure for each subpackage is straightforward and can be successfully performed by following the instructions in the README files or and in ICS_Regge, PADE II Manuals. Here we just want to notice that two ways of building the ICS_Regge and PADE II executables are available. One is manual and is recommended for the first time users at the installation stage, whereas the fully automated way is useful at late stages of using the package. While a manual building assumes using a Makefile_UNIX with the adjusted environmental variables, an automated building relies on the run script runICS with the built-in instructions for making the packages. The appropriate values of environmental variables are passed, in this case, through the input file ICS/input/INPUT controlled by a user (see section 5 for details).

3.2 Testing

Three test suits are prepared for each subpackage to validate the installation procedure.

3.2.1 Running the QUADPACK test

To test the QUADPACK library, one has to run quadpack_prb.sh script in /QUADPACK/scripts directory.

The results of 15 tests can be found in quadpack_prb_output.txt file in ICS/QUADPACK/test directory. The message QUADPACK_PRB: Normal end of execution confirms that the code passed the validation test.

3.2.2 Running the PADE II test suite

The input for 4 jobs, test1, test2, test3 and test4, are provided in directories ICS/PADE/test/input/test_name, where test_name is either test1 or test2 or test3 or test4. To submit and run a test suite, first, one can
change the definition of the Fortran compiler, F77, C compiler, CC, their paths, COMP_PATH and CCOMP_PATH respectively, if necessary in Makefile test in directory ICS/PADE/src, and then go to the directory ICS/PADE/test and run a script ./run_TEST. All tests will be run in separate directories. Each test takes about 1 - 5 minutes to run on a reasonably modern computer. The message Test test_name was successful appearing on the screen at the conclusion of the testing process confirms that the code passed the validation test test_name. The results of the simulation can be viewed in the ICS/PADE/test/output/test_name directory. The message Your output differs from the baseline! means that the calculated data significantly differ from that in the baselines. The files PADE/test/output/test_name/diff_file can be checked to judge the differences.

3.2.3 Running the ICS_Regge test suite

For simplicity, the test suite for ICS_Regge is designed in the similar manner as the test suite for PADE II. The input for 3 jobs, test1, test2 and test3 are provided in directories ICS/test/input/test_name respectively, where test_name is either test1 or test2 or test3. The definitions of the FORTRAN and C compilers and their paths, as well as the path of the PADE II directory, can be changed in the INPUT files in each directory test_name, if required. The test suit can be run in the directory ICS/test using the following commands: ./run_TEST < test_data.txt. All tests are run in separate directories, ICS/test/output/test1, ICS/test/output/test2 and ICS/test/output/test3. Each test takes about 1 - 6 minutes to run on a reasonably modern computer. To analyse the test results, the message on the screen at the conclusion of the testing process has to be inspected. The message Test test_name was successful confirms that the code passed the validation test test_name. The results of the simulation can be viewed in the ICS/test/output/test_name directory. The message test_name output differs from the baseline!

Check your output in output/test_name /ics.mull

Baseline file: baselines/test_name /ics.mull

Differences: output/test_name /error_file

means that the calculated data significantly differ from that in the baselines. The differences can be found in the files ICS/test/output/test_name/error_file.
4 Computational modules of ICS_Regge

4.1 Padé reconstruction and *PADE\_II* options

As discussed above, an important part of the calculation consists in performing an analytical continuation of the $S$-matrix element into the CAM plane. The user has some flexibility in doing so. It concerns mostly the quadratic phase in Eq. (13), which must itself be determined in the course of the Padé reconstruction. The need for separating this rapidly oscillating term arises from the fact that the Padé technique used here, works best for slowly varying functions. Thus, by removing the oscillatory component, one expands the region of validity of the approximant in the CAM plane, which allows to correctly describe a larger number of poles. The extraction of the quadratic phase proceeds iteratively. Since sharp structures in the phase of the $S$-matrix element usually come from the Regge poles and zeroes located close to the real axis, one defines a strip $-\text{d}x > \text{Im} J \text{d}x$ around the real $J$-axis, and removes from the previously constructed approximant all poles and zeroes inside the strip. The smoother phase of the remainder is fitted to a quadratic polynomial, and this quadratic phase is then subtracted from the phase of the input values of $S$, after which a new approximant is constructed with these modified input data. The process is repeated $n_{\text{iter}}$ times resulting in a (hopefully) improved Padé approximant. There is no rigorous estimate of the improvement achieved, and the practice shows that in many cases using $n_{\text{iter}} > 1$ gives tangible benefits, while in some cases better results are achieved with $n_{\text{iter}}=1$ or 2. It is for the user to decide on the best values of $\text{d}x$ and $n_{\text{iter}}$ for a particular problem.

The input files required for running the Padé\_II code are stored in the input directory where they are labelled 1, 2...$N_E$. A typical input file is given in Appendix B. The file differs from the similar input used in Padé\_II package reported in [50] by one line added at the end, which should contain the collision energy in meV.

Other parameters for Padé reconstruction are read from the input/INPUT file. These include the pathways to the PADE directory, to the FORTRAN and C compilers, and the compiler options (see #16-#21). The next entry (#22) determines whether there should be a change of parity from the original data. This depends on the convention used in calculating the $S$-matrix elements, as explained in [50] (we use 0 for no and 1 for yes). Entry (#23) decides whether one should remove the guessed values of the quadratic phase prior to the construction of the first Padé approximant in a series of iterations. Its recommended value is 1 (yes).

Entry (#24) determines if multiple precision routines should be used in calculating the Padé approximant. The recommended value is 1 if the number of partial waves (PW) exceeds 40. It can also be used for a smaller number of (PW) to check the stability of calculations.
Entries (#25 and #27) allow us to repeat the calculations with added non-analytical noise of magnitude $\text{fac, nstime}$ times. This may be needed to check the sensibility of calculations to numerical noise. The recommended initial values are $\text{nstime}=0$ and $\text{fac}=0$, in which case no noise is added. Entry (#26) determines the number of points in the graphical output from Padé II [50].

Finally, the user has the options of changing the number of partial waves, the number of iterations and the value of $\text{dxl}$ for all files used in the current run by setting to 1 $\text{iover1}$, $\text{iover2}$ and $\text{iover3}$ in entries (#28, #29 and #30). The corresponding parameters are reset to the values $\text{nread1}$, $\text{niter1}$, and $\text{dxl1}$, specified in the entries (#31, #32 and #33), respectively.

### 4.1.1 Changes made to PADE II

The changes from the previous version [50] include (I) replacement of all the Numerical Algorithms Group (NAG) routines with ones available in the public domain, and (II) provision of additional controls allowing to change the parameters of Padé reconstruction for all energies in the run at once, without changing individual input files labelled 1,2,..., as discussed in the previous Section. A brief summary of the changes made to subroutines is given below.

**Subroutine FIT (fit.f)**

The NAG routine g05ccf has been replaced by a subroutine svdfit described in section 15x.4 of *Numerical Recipes in C: The Art of Scientific Computing (Second Edition)*, published by Cambridge.

**Subroutine ZSRND (zsrnd.f)**

The NAG routine E02ACF has been replaced by a sequence of calls to the system routines srand48 and drand48.

**Subroutine IDET_SEED (idet_seed.c)**

Added new routine generating the seed for srand48.

**Wrapper (wrapper.c)**

Added wrapper allowing for calling C-routines idet_seed.c, srand48 and drand48 in the Fortran code.

### 4.2 The structure of ICS_Regge

The ICS_Regge application is a sequence of 12 FORTRAN files. The files are listed below, and their functions are explained.
Program *ICS_Regge* *(ICS_Regge.f)*
Main program.

**Subroutine READ1** *(read1.f)*
Opens and reads the original input file at a given energy.

**Subroutine READ** *(read.f)*
Reads the parameters of the Padé reconstruction at a given energy.

**Subroutine SORT** *(sort.f)*
At a given energy, selects poles and zeroes in the specified region of the complex angular momentum \((J^-)\) plane and discards pole/zero pairs (Froissart doubles).

**Subroutine TCROSS** *(tcross.f)*
Evaluates the partial wave sum (PWS) for the integral cross section using the original data and also the Padé approximant for integer values of \(J\). Estimates the error of the Padé reconstruction.

**Subroutine TCINTE** *(tcinte.f)*
Replaces the discrete PWS by integration over continuous values of \(J\). Evaluates the integral using the Padé approximant.

**Subroutine MULLO** *(mullo.f)*
At a given energy, evaluates the resonance (Mullholland) contribution to the ICS from a pole chosen by hand from the list of available poles.

**Subroutine MULL** *(mull.f)*
At a given energy, evaluates the resonance (Mullholland) contribution to the ICS from a pole with the real part closest to that of the pole chosen at the previous energy.

**Subroutine ZPADE** *(zpade.f)*
Calculates the full Padé approximant.

**Subroutine ZPR** *(zpr.f)*
Evaluates the ratio of the two polynomials in the Padé approximant.

**Subroutine ZRES** *(zres.f)*
Calculates (part of) the residue for a chosen Regge pole from the Padé approximant.

**Subroutine FST3** *(fst3.f)*
Supplies the integrand for the integral evaluated in *tcinte.f*. 
There are two additional utilities.

**Subroutine SKIP (skip.f)**
Decides which of the input data/energies must be included in the current run.

**Subroutine SUBTR (subtr.f)**
Subtracts from the ICS the contribution from a given Regge pole trajectory. All files are located in the ICS/src directory.

### 4.3 The QUADPACK library


Currently one library subroutine, DQAGS, is used in the ICS_Regge. The subroutine estimates integrals over finite intervals using an integrator based on globally adaptive interval subdivision in connection with extrapolation [45] by the Epsilon algorithm [46]. The subroutine is called from the ICS_Regge subroutine tcinte. For users convenience the whole library is available in the package ICS_Regge. The link to the library is provided in Makefile and Makefile_UNIX in ICS/src.

### 5 Running the ICS_Regge code

#### 5.1 Creating input data

Three input files are required for running calculations: a parameter file, INPUT, located in ICS/input directory, and two input files for running PADE-II. The first file is the PADE-II parameter file, param.pade. It is created on the fly by running the script runICS. The second file or set of files has(ve) to be supplied in ICS/input/PADE_data directory and it (or they) contain(s) the data to be Pade approximated. The name of directory PADE_data can be chosen arbitrary and should be specified in the parameter file INPUT before the starting calculations. The names of the input files in the directory PADE_data are fixed to be 1, 2, ...NE, each of which contains previously computed values of $S_{\nu_\nu}(E,J)$ for $J = 0, 1, 2, ..., J_{max}$, for the energy $E_k$, $k = 1, 2, ..., NE$, and the
value of the energy itself. An example of input file is given in Appendix B and also provided in directory ICS/input. The file ICS/input/INPUT is self-explanatory and describes each input parameter to be specified. Please notice, that each input entry appears between colons (:). We provide input files for all test cases considered under the names ICS/input/INPUT.BOUND, ICS/input/INPUT.META and ICS/input/INPUT.FH2. For an example of the parameter file INPUT see Appendix A.

5.2 Executing ICS_Regge

The script runICS in ICS/ directory automates calculations. The following assumptions are made in the script:
* all binaries for ICS_Regge are placed in ICS/bin whereas the binaries for the PADE_II package are located in ICS/PADE/bin.
* input files are located in directory ICS/input. The names of the input files are chosen as described in section 4.1.
* output files can be found in ICS/output on completion of the calculation. We recommend running a calculation in directory ICS/. The command ./runICS immediately starts the calculation.

5.3 Understanding the run script runICS

The run script runICS located in ICS/ directory does not require any tuning, editing or corrections in order to start the calculation. Provided that the parameter file ICS/input/INPUT is prepared for calculations, the run script runICS takes care of the following steps in the following order:
1. INITIALIZATION
   * Edits input parameter file INPUT
   * Reads input parameters from INPUT
   * Prepares directories for runs
   * Sets the useful directories
   * Cleans the directories if necessary
2. BUILDING PACKAGES
   * PADE_II
   * ICS_Regge
   * Utilities
3. RUNNING ICS_Regge FOR ALL INPUT FILES OF INTEREST
   * Checks if the input file falls in the range of energies under investigation
   * Runs PADE_II with the current input file if it is in the considered range
   * Runs ICS_Regge if the current input file is in the considered range
*evaluates the non-resonance background by subtraction the Mulholland con-
4. OUTPUT DATA MANAGEMENT
* Stores the calculated data in the appropriate files

5.4 Using the code

Using the code involves at least two steps.

5.4.1 Step I

In the parameter file input/INPUT (see Appendix A) one sets

is this the first run? :yes:. The code evaluates the poles $P_i$ and the
gozeroes $Z_i$ of the Padé approximant (13) for each collisional energy $E_k$, for the
chosen set of files (see #5-#6 of Appendix A), in a region of the CAM plane,

$$x_{\text{min}} \leq \Re P_i, Z_i \leq x_{\text{max}}, y_{\text{min}} \leq \Im P_i, Z_i \leq y_{\text{max}},$$

with the values $x_{\text{min}}, x_{\text{max}}, y_{\text{min}},$ and $y_{\text{max}}$ specified by the user in the file input/INPUT (see

#12 -#15 in the Appendix A).

One has the option of not including in the Padé approximant (13) the Frois-
sart doublets, i.e., pole-zero pairs, with the distance $|P_i - Z_j| < \epsilon$, with the
value of $\epsilon$ set in #11 of Appendix A. Such pairs often represent non-analytical
noise present in the input data [48], and their removal may be beneficial.

Also, at this stage the program evaluates, for all energies, the exact ICS using

Eq.(1) or (3), and the first integral in (9). The results are written in the files

output/ics.exact and output/ics.int, respectively. Provided the energy is
entered in meV, and the reduced mass is in the unified atomic mass units,

(u.a.u or Daltons) (see #10 of Appendix A), the cross sections are in the units
of angstroms squared ($\text{Å}^2$).

One then identifies Regge trajectories by plotting the pole positions vs. en-
ergy from the output file output/ics.pole. (It is recommended to to use the
plot $\Re P(E)$ vs. $E$, as the real parts of the pole positions are less sensitive to
numerical noise). In the plot the trajectories appear as continuous strings of
poles, with additional poles scattered around them in a random manner.

5.4.2 Step II

In the file input/INPUT one sets is this the first run? :no:. The code
takes the first of the files in the energy range $E_{\text{min}} \leq E \leq E_{\text{max}}$ with $E_{\text{min}}$ and

$E_{\text{max}}$ specified by the user in #8-#9 of the file input/INPUT (see Appendix
A). It then displays all the poles at this energy within the specified range,
from which the user chooses the one lying on the Regge trajectory of interest. Then, if one sets follow trajectory by hand? :no: the code will follow the trajectory automatically, choosing at the next energy the pole whose real part is closest to that of the pole chosen at the previous energy.

If one chooses follow trajectory by hand? :yes: the program continues display the poles, from which the user must choose the desired one for all values of energy by hand. (The "by hand" option is useful, e.g., when working with a poorly defined trajectory from each some of the poles may be missing.) In both cases, the corresponding Mulholland contribution in Eq. (9),

\[
I_{\text{Mull}}^n(E) = \frac{8\pi^2}{k^2} Im \lambda_n Res \left[ S_{\nu' \nu}(E, \lambda_n) \right] S^*_\nu(E, \lambda_n^*) \frac{1}{1 + \exp(-2i\pi \lambda_n)}
\]

is written down in the file output/ics.mull.

The pole position and the corresponding residue (in meV) are stored in the files output/ics.traj and output/ics.resid.

Once the energy \( E_{\text{max}} \) is reached, the program stops and \( I_{\text{Mull}}^n(E) \) is subtracted from the exact ICS in the file output/ics.exact, the result written in output/ics.smooth.

Step II can be repeated several times, thus making the program follow different Regge trajectories, while choosing \( E_{\text{min}} \) and \( E_{\text{max}} \) as is convenient. At the end, the Mulholland contributions from all trajectories considered are subtracted from the exact ICS, and the much less structured non-resonance part of the ICS is stored in the file output/ics.smooth.

Prior to operating the code one must specify whether the transition is elastic or not, by answering elastic channel? with either :yes: or :no:. For single channel scattering one may also choose to use the modified Mulholland formula (12) by setting in the input/INPUT: modified Mulholland? :yes:.

In this case the Mulholland contribution stored in output/ics.mull is given by the pole term in (12),

\[
I_{\text{Mull}}^n(E) = \frac{8\pi^2}{k^2} Im \lambda_n Res \left[ S(E, \lambda_n) \right] \frac{1}{1 + \exp(2i\pi \lambda_n)}
\]

(18)

Note that to use the modified formula (12) after using the one given in Eq.(9), one must also repeat the step I first. This is necessary to recalculate the exact ICS, from which the contribution (17) will then be subtracted.

6 Examples of using ICS_Regge

Three examples of using the ICS_Regge package are provided. The input and output files for these examples are included in the package.
6.1 Example 1: The hard sphere model (Regge trajectory of the type I)

The first example involves the $S$-matrix element for potential (single channel) scattering off a hard sphere of a radius $R - d$ surrounded by a thin semi-transparent layer of a radius $R$. The spherically symmetric potential $V(r)$ is infinite for $r < R - d$, has a rectangular well of a depth $V$ for $R - d < r < R$, a zero range barrier $\Omega \delta(r - R)$ ($\delta(x)$ is the Dirac delta), and vanishes elsewhere [15]. In this example the energy of a non-relativistic particle of a mass $\mu = 1$ u.a.u varies from $1 \text{meV}$ to $100 \text{meV}$, the radii of the hard sphere and the width of the well $d$ are 2.045Å and 0.592Å, respectively, $V = 165 \text{meV}$, and $\Omega = 1.023 \text{meV} \cdot \text{Å}$. In this range, there is a single resonance Regge trajectory, originating at $J = 0$ in the bound state of the well at about $-14 \text{meV}$. A detailed discussion of this model can be found in Refs.[15] where the Regge trajectories were obtained by direct integration of the radial Schroedinger equation for complex values of $J$. Here we seek to repeat the results of [15] by evaluating the $S$ matrix element for integer $J$’s, and then using the Padé reconstruction. The data files are in the directory input/BOUND.

Step I

In the directory input, copy the file INPUT.BOUND into the file INPUT. Run the code to completion. Use the file output/ics.pole to plot real parts of the poles vs. energy, and identify the Regge trajectory of interest. Use the file output/ics.exact to plot the integral cross section in the specified range of energies.

Step II

In the file INPUT change is this the first run? :yes: to is this the first run? :no:. In the entry #5 set modified Mulholland? :no: or modified Mulholland? :yes: to use Eq.(9) or Eq.(12), respectively. Run the code. When prompted to choose a pole, choose the one at the beginning of the trajectory, with $ReJ \approx 4.85$ and $ImJ \approx 0.0043$. After completion, use the files in the directory output to plot the results. The correct results for the Regge trajectory, the residues of the resonance pole, the Mulhollolland contribution, and the background cross section are shown in Fig.1.

6.2 Example 2: The hard sphere model (Regge trajectory of the type II)

This is the same model as in Example 1, but with $\Omega = 66.463 \text{meV} \cdot \text{Å}$, considered in the range of collision energies from $40 \text{meV}$ to $100 \text{meV}$. In this case, there is a single resonance Regge trajectory, originating at $J = 0$ in a metastable state with the real part of about $48 \text{meV}$. The data files are in the directory input/META.

Step I

In the directory input copy the file INPUT.META into the file INPUT. Run
Fig. 1. Example 1: (a) The full ICS (solid) and the smooth background obtained by subtracting from it the resonance contribution $\sigma_{\text{res}}$ given by the pole term in Eq.(6) (dashed) and in Eq.(12) (dot-dashed); (b) the real (dashed) and the imaginary (solid) parts of the resonance Regge trajectory which affects the ICS in the shown energy range; (c) the modulus (solid) and the real part (dashed) of the residue of the Regge pole, $\rho \equiv \text{Res}[S]$, which traces the trajectory in (b).

the code to completion. Use the file output/ics.pole to plot real parts of the poles vs. energy, and identify the relevant Regge trajectory. Use the file output/ics.exact to plot the integral cross section in the specified range of energies.

Step II
In the file INPUT, change is this the first run? :yes: to is this the first run? :no:. In the entry #5 set modified Mulholland? :no: or modified Mulholland? :yes: to use Eq.(9) or Eq.(12), respectively. Run the code. When prompted to choose a pole, choose the one at the beginning of the trajectory, with $\text{Re}J \approx 0.195$ and $\text{Im}J \approx 3.22$. After completion, use the files
in the directory output to plot the results. The correct results for the Regge trajectory, the residues of the resonance pole, the Mulholland contribution, and the background cross section are shown in Fig. 2.

Fig. 2. Example 2: (a) The full ICS (solid) and the smooth background obtained by subtracting from it the resonance contribution $\sigma_{\text{res}}$ given by the pole term in Eq.(6) (dashed) and in Eq.(12) (dot-dashed); (b) the real (dashed) and the imaginary (solid) parts of the resonance Regge trajectory which affects the ICS in the shown energy range; (c) the modulus (solid) and the real part (dashed) of the residue of the Regge pole, $\rho \equiv \text{Res}[S]$, which traces the trajectory in (b).
6.3 Example 3: The $F + H_2(v = 0, j = 0, \Omega = 0) \rightarrow HF + H(v' = 2, j' = 0, \Omega' = 0)$ reaction. (Two pseudo-crossing Regge trajectories)

This example uses realistic numerical data obtained in Ref. [9], and analysed in Refs. [34]-[35], [40]. In the specified energy range there are two resonance Regge trajectories, both contributing to the Regge oscillations seen in the state-to-state integral cross section. At the collision energy of about 38meV the imaginary parts of the trajectories cross, while the real parts do not (for details see Ref.[39]). The data files are in the directory input/FH2.

Step I
In the directory input copy the file INPUT.FH2 into the file INPUT. Run the code to completion. Use the file output/ics.pole to plot real parts of the poles vs. energy, and identify two relevant Regge trajectories. Use the file output/ics.exact to plot the integral cross section in the specified range of energies.

Step II
In the file INPUT, change is this the first run? :yes: to is this the first run? :no:. In the entry #5 keep modified Mulholland? :no: respectively. Run the code. When prompted to choose a pole, choose the one at the beginning of first trajectory, with $ReJ \approx 1.54$ and $ImJ \approx 1.17$. After completion, if necessary, save the data in files output/ics.traj, output/ics.mull and output/ics.resid, as they will be overwritten.

Run the code again. When prompted to choose a pole, this time choose the one at the beginning of second trajectory, with $ReJ \approx 5.54$ and $ImJ \approx 1.43$. After completion, use the files in the directory output to plot the results. The correct results for the Regge trajectories, the residues of the resonance poles, their respective Mulholland contributions, and the background cross section are shown in Fig.3.
Fig. 3. **Example 3:** (a) The full ICS (solid) and the smooth background obtained by subtracting from it the resonance contribution $\sigma_{\text{res}}$ given by the pole sum term in Eq.(6) (dashed); (b) the real (dashed) and the imaginary (solid) parts of the two resonance Regge trajectories which affect the ICS in the shown energy range; (c) the Mulholland contributions [pole terms in Eq.(6)] for the trajectories shown in (b): (I) (solid) and (II) (dashed). (d) the moduli (solid) and the real parts (dashed) of the residues of the Regge poles, $\rho \equiv \text{Res}[S]$, which trace trajectories (I) and (II) in (b).

### 7 Summary

In summary, we present a user friendly computer code which evaluates the contribution a resonance Regge trajectory makes to an integral cross section.
Regge poles positions and residues are evaluated from numerical values of the corresponding scattering matrix element by Padé reconstruction. The code can be used for analysing elastic, inelastic and reactive transitions.

8 Acknowledgements:

We acknowledge support of the Basque Government (Grant No. IT-472-10), and the Ministry of Science and Innovation of Spain (Grant No. FIS2009-12773-C02-01). The SGI/IZO-SGIker UPV/EHU is acknowledged for providing computational resources.
Appendix A: Example of ICS_Regge parameter file INPUT

```
# ICS RUN PARAMETER
$1) directory for input
 : PR2 :
$2) is this the first run? (yes or no)
 : yes :
$3) follow trajectory by hand? (yes or no)
 : no :
$4) elastic channel? (yes or no)
 : no :
$5) modified Mulholland? (yes or no)
 : no :
$6) total number of input files to read
 : 223 :
$7) the first input file to read - 1, i.e. 0 corresponds to the first file
 : 0 :
$8) the lowest energy (meV) to consider
 : 0 :
$9) the highest energy (meV) to consider
 : 200 :
$10) the reduced mass (in unified atomic mass units)
 : 1.023 :

# ICS INTERNAL PARAMETER
$11) threshold for filtering Froissart doublets (in J)
 : 0.0 :
$12) real lower limit for poles/zeros (in J)
 : 0.0 :
$13) real upper limit for poles/zeros (in J)
 : 20.0 :
$14) imaginary lower limit for poles/zeros (in J)
 : -0.0 :
$15) imaginary upper limit for poles/zeros (in J)
 : 0.0 :

# PADE PARAMETERS
$16) PADE directory
 : /home/gfpsod/ELKERA/guinnes/ICS/PADE/ :
$17) Fortran compiler’s path
 : /opt/intel/compiler.xe_2011.spl.11.3.39/bin/intel164/ :
$18) C compiler’s path
 : /usr/bin :
$19) Fortran compiler
 : ifort :
$20) C compiler
 : cc :
$21) Fortran compilation’s options
 : - : 
$22) ipar
 : 0 :
$23) iprec
 : 1 :
$24) imult
 : 0 :
$25) nstime
 : 0 :
$26) nprint
 : 2000 :
$27) fac
 : 0.20000001 :
$28) iover1 to change No of partial waves
 : 0 :
$29) iover2 to change No of iterations in Pade
 : 0 :
$30) iover3 to change dal
 : 0 :
$31) nreadl
 : 15 :
$32) niter1
 : 3 :
$33) dal
 : 1.5 :
```

Fig. 4. An example of the INPUT file for an inelastic or reactive transition.
The contents of the file, also described in [50], include:

- **nread**: the number of partial waves read,
- **niter**: the number of iterations to remove the quadratic phase,
- **sht**: this shifts the input grid points and may be used to avoid exponentiation of extremely large number when evaluating the polynomials involved.

The value \( sht = \frac{nread}{2} \) is suggested for a large number of partial waves.

- **jstart** and **jfin**: with all input points numbered by \( j \) between 1 and \( N \), determine a range \( j_{\text{start}} \leq j \leq j_{\text{fin}} \) to be used for the Padé reconstruction.
- **inv**: set to \(-1\), not used in present calculations,
- **dxl**: determines the width of the strip in which poles and zeroes are removed while evaluating the quadratic phase in Eq. (12).

The table below shows the input parameters for the file:

<table>
<thead>
<tr>
<th>j</th>
<th>nread</th>
<th>niter</th>
<th>sht</th>
<th>jstart</th>
<th>jfin</th>
<th>inv</th>
<th>dxl</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>20</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>18</td>
<td>-1</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td>0.</td>
<td>-1.99870601</td>
<td>0.0508557074</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.</td>
<td>-0.555148566</td>
<td>-0.896604378</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.</td>
<td>-0.025640943</td>
<td>-0.224998729</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.</td>
<td>-0.000284346738</td>
<td>-0.0238455997</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.</td>
<td>-2.39786402E-06</td>
<td>-0.00218991376</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.</td>
<td>-7.80826626E-10</td>
<td>3.95177561E-05</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.</td>
<td>-1.14352972E-14</td>
<td>1.50673028E-07</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7.</td>
<td>0.</td>
<td>6.74187776E-10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8.</td>
<td>0.</td>
<td>2.03219359E-12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.</td>
<td>0.</td>
<td>-1.06243235E-15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.</td>
<td>0.</td>
<td>-6.73667676E-17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11.</td>
<td>0.</td>
<td>-6.00290847E-19</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12.</td>
<td>0.</td>
<td>-4.19847314E-21</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13.</td>
<td>0.</td>
<td>-3.95924061E-23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>14.</td>
<td>0.</td>
<td>1.23879594E-25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15.</td>
<td>0.</td>
<td>9.01172959E-29</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16.</td>
<td>0.</td>
<td>1.0179967E-31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>17.</td>
<td>0.</td>
<td>0.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>18.</td>
<td>0.</td>
<td>0.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>19.</td>
<td>0.</td>
<td>0.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 5. The input file 1, which contains the input parameters required to run Padé II, the values of the S-matrix element for different values of \( J \), and the value of collision energy in meV.
11 Appendix C: Structure of the ICS directory

Fig. 6. Detailed structure of the ICS directory.
Fig. 7. Detailed structure of the PADE directory.
Appendix E: Structure of the QUADPACK directory

Fig. 8. Detailed structure of the QUADPACK directory.
References


Please note that there is a typo in Eq.(5), which should read

\[ S(k, \lambda) = \frac{\Delta^{(2)}(E, \lambda)}{\Delta^{(1)}(E, \lambda)} \equiv \frac{-H^{(2)}_\lambda(kr)\{[\ln H^{(2)}_\lambda(kr)]' - [\ln \phi_{well}(r, k, \lambda)]' - 2\Omega\}}{H^{(1)}_\lambda(kr)\{[\ln H^{(1)}_\lambda(kr)]' - [\ln \phi_{well}(r, k, \lambda)]' - 2\Omega\} |_{r=R}}, \]

with \( \phi(r) = A(k, \lambda)\sqrt{r}\phi_{well}(r) \) and \( q \equiv (k^2 + 2V)^{1/2} \). This regrettable misprint does not, however, affect the results or conclusions of the paper, as the correct values have been used in all calculations.


