ELSEVIER

Computer Physics Communications 144 (2002) 224–241

Computer Physics Communications

www.elsevier.com/locate/cpc

GTOBAS: fitting continuum functions with Gaussian-type orbitals ☆

Alexandre Faure, Jimena D. Gorfinkiel, Lesley A. Morgan, Jonathan Tennyson *

Department of Physics and Astronomy, University College London, Gower Street, London WC1E 6BT, UK

Received 21 May 2001; received in revised form 12 September 2001

Abstract

GTOBAS is a program for fitting Gaussian-type orbitals (GTOs) to Bessel and Coulomb functions over a finite range. The exponents of the GTOs are optimized using the method of Nestmann and Peyerimhoff [J. Phys. B 23 (1990) L773]. The appended module NUMCBAS provides the numerical Bessel and Coulomb functions required as input for the program. The use of GTO continuum basis sets is particularly important in electron–molecule scattering calculations when polyatomic targets are involved. Sample results for such calculations are also discussed. © 2002 Elsevier Science B.V. All rights reserved.

PACS: 02.60.Ed; 34.80.-i; 52.20.Fs

Keywords: Gaussian basis sets; Bessel and Coulomb functions; R-matrix calculations

PROGRAM SUMMARY

1. GTOBAS

Title of program: GTOBAS

Catalogue identifier: ADPV

Program Summary URL: http://cpc.cs.qub.ac.uk/summaries/ADPV

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

Computer for which the program is designed and others on which it has been tested: Compaq Alpha-DEC, IBM RS/6000

Operating systems or monitors under which the program has been tested: Digital UNIX V5.0, IBM AIX 4.3.2.0

Programming language used: Fortran 90

Memory required to execute with typical data: less than 0.25 Mwords

No. of bits in a word: 32

No. of processors used: 1

Has the code been vectorized?: no

No. of bytes in distributed program, including test data, etc.: 8376

☆ The programs can be downloaded from the CPC Program Library under catalogue identifiers: http://cpc.cs.qub.ac.uk/summaries/ADPV and http://cpc.cs.qub.ac.uk/summaries/ADPW

* Corresponding author.
E-mail address: j.tennyson@ucl.ac.uk (J. Tennyson).

PII: S0010-4655(02)00141-8

Distribution format: tar gzip file

Keywords: Gaussian basis sets, Bessel and Coulomb functions, R-matrix calculations

Nature of physical problem
Optimizing GTO basis sets to represent continuum functions.

Method of solution
Numerical continuum functions are read from an external file and the GTOs set is optimized using the method proposed by Nestmann and Peyerimhoff [1].

Restrictions on the complexity of the problem
The limitation for obtaining satisfactory fits is linked to the intrinsic difficulty of representing a large number of nodes with nodeless Gaussian functions expanded about a single centre.

Typical running time
1 to 10 minutes (depending on the number of needed GTOs) plus the time taken by the subroutine used to generate the numerical functions.

Unusual features of the program
The program makes use of subroutines from Numerical Recipes [2]. We also append a module, NUMCBAS, for generating Bessel and Coulomb functions (see below).

2. NUMCBAS

Title of program: NUMCBAS

Catalogue identifier: ADPW

Program Summary URL: http://cpc.cs.qub.ac.uk/summaries/ADPW

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

Computer for which the program is designed and others on which it has been tested: Compaq Alpha-DEC, IBM RS/6000

Operating systems or monitors under which the program has been tested: Digital UNIX V5.0, IBM AIX 4.3.2.0

Programming language used: Fortran 90

Memory required to execute with typical data: less than 0.1 Mwords

No. of bits in a word: 32

No. of processors used: 1

Has the code been vectorized?: no

No. of bytes in distributed program, including test data, etc.: 7463

Distribution format: tar gzip file

Keywords: Bessel and Coulomb functions

Nature of physical problem
Evaluation/calculation of Bessel and Coulomb functions. Provides input for GTOBAS.

Method of solution
Numerical integration of the Schrödinger equation from both boundaries and matching using de Vogelaere’s algorithm.

Restrictions on the complexity of the problem
The program could in principle solve a model scattering problem with any kind of potential, although in its present form, only Coulomb potentials can be input.

Typical running time
Typically, 0.03 s.

References

[1] B.M. Nestmann, S. Peyerimhoff, J. Phys. B: At. Mol. Opt. Phys. 23 (1990) L773.
[2] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes in Fortran, Cambridge University Press, 1995.

LONG WRITE-UP

1. Introduction

Standard quantum chemistry packages routinely use Gaussian-type orbitals (GTOs) to describe the electronic structure of nonlinear molecules. Although Slater-type orbitals (STOs) have the proper “cusp” near nuclei and the correct asymptotic behaviour, their use is almost totally confined to atomic and linear molecule calculations because the multicenter integrals that arise in nonlinear calculations cannot be performed efficiently (see, e.g., [1]). In contrast, such integrals can routinely be evaluated (in closed form) when GTOs are used. This fundamental advantage has lead to the dominance of GTOs in molecular electronic structure calculations.

A huge number of GTO basis sets have been made available for bound-state calculations [2]. On the other hand, few basis set generation procedures exist for the representation of Rydberg and continuum orbitals [3]. These functions cover a large spatial domain and their radial part is characterised by a great number of nodes. In the context of electron–molecule scattering calculations, the use of numerical functions to represent the continuum has proved very successful for both atomic and linear targets [4]. For nonlinear molecules, however, there are at present no adequate numerical procedures available. An alternative approach, explored originally in the context of R-matrix calculations, involves the use of GTOs to represent both the continuum and target electrons. Nestmann and Peyerimhoff [5] developed a method to fit GTO basis sets to Bessel functions within the finite region of an R-matrix sphere. In spite of the intrinsic difficulty of representing a large number of nodes with (nodeless) Gaussian functions expanded about a single centre, this method was found to give excellent results for Bessel functions with eigenenergies up to 16 eV. The method has been used to obtain a number of GTO continuum basis sets, all for neutral targets and to be used in calculations with an R-matrix radius of $R_{\text{mat}} = 10\,a_0$ [5–7]. A general discussion on approximating functions with non-orthogonal basis sets can be found in [8].

The aim of this work is to provide a program which is able to construct adequate GTO continuum basis sets for representing both Bessel and Coulomb functions using the procedure described by Nestmann and Peyerimhoff and to briefly discuss the behaviour of these basis sets in actual calculations. We also append the module NUMCBAS, based on a program by Salvini [9] and which is a cut-down version of the module NUMBAS used in the UK molecular R-matrix codes [10] to generate the numerical continuum orbitals. The paper is organised as follows: Section 2 summarises the method used to optimise the Gaussian exponents while Section 3 reports and discusses sample results. The program organisation and input data description are given in Section 4. Test data is given in Section 5.

2. Method

We summarise here the procedure used in GTOBAS to optimise the Gaussian exponents. A brief description of the module NUMCBAS, used to generate numerical Bessel and Coulomb functions, is also given. It should be noted that any other user-supplied program can be used to produce these functions, or indeed other functions that one may wish to fit.

2.1. Generation of numerical continuum orbitals

The module NUMCBAS defines an adaptive grid of radial coordinates r_k to generate numerical continuum orbitals $u_{hl}(r_k)$ by solving the model, single channel scattering equation:

$$\left[\frac{\mathrm{d}^2}{\mathrm{d}r_k^2} - \frac{l(l+1)}{r_k^2} + 2V_0 + k_h^2\right]u_{hl}(r_k) = 0, \tag{1}$$

where l is the angular momentum quantum number, k_h^2 are the eigenenergies and V_0 is a model potential, both in Rydbergs. This equation is solved subject to the fixed boundary conditions:

$$u_{hl}(0) = 0 \quad \text{for } l \neq 0, \qquad \frac{1}{u_{hl}(r_k)}\left[\frac{\mathrm{d}u_{hl}(r_k)}{\mathrm{d}r_k}\right]_{r_k=R_{\text{lim}}} = 0, \tag{2}$$

where R_{lim} is the boundary radius in Bohrs ($r_k \leqslant R_{\text{lim}}$). For the special case $l = 0$ the value of $u_{hl}(0)$ is obtained by a two point Lagrange interpolation. Only those eigenfunctions whose energy k_h^2 is smaller than the parameter E_{up} (typically a few Rydbergs) are evaluated. NUMCBAS finds solutions to Eq. (1) by integrating functions from both boundaries and matching them using de Vogelaere's algorithm [11,12]. It has been found that, in practice, V_0 in Eq. (1) can be replaced by Z/r_k where Z is the effective charge of the target. In this case, the eigensolutions become numerical representations of spherical Bessel ($Z = 0$) or Coulomb functions over a finite range.

2.2. Optimisation procedure

The procedure used to obtain the Gaussian exponents was originally proposed by Nestmann and Peyerimhoff [5]. The main idea of this scheme is to fit the continuum functions $u_{hl}(r_k)$ by a set of Gaussian functions with exponents α_i. This is performed by minimising the function:

$$F_{l,N} = \sum_{h=1}^{N} \frac{\sum_k \left[\sum_{i=1}^{n} c_{hi} r_k^l \exp(-\alpha_i r_k^2) - u_{hl}(r_k)\right]^2}{\sum_k [u_{hl}(r_k)]^2} + D(\alpha_1, \ldots, \alpha_n), \tag{3}$$

where N is the number of continuum functions and n is the number of Gaussian exponents. The minimum of $F_{l,N}$ is obtained by Powell's method [13] (prototype of multidimensional direction-set methods) using $\ln(\alpha_i)$ as variational parameters. The coefficients c_{hi} are determined by a least-squares fit.

The term $D(\alpha_1, \ldots, \alpha_n)$:

$$D(\alpha_1, \ldots, \alpha_n) = \sum_{i=2}^{n} \sum_{j=1}^{i-1} \exp\left(-g \times \left|\frac{\alpha_i}{\alpha_j} - \frac{\alpha_j}{\alpha_i}\right|\right), \tag{4}$$

is added to avoid the convergence of two different α_i towards the same value and hence problems with linear dependence. In Eq. (4) the real number g has been set to R_{lim}, the boundary radius defined in NUMCBAS (see Eq. (2)).

It is important to note that the logarithmic boundary condition defined in Eq. (2) leads to an artificial constraint on the numerical continuum functions at $r_k = R_{\mathrm{lim}}$. Employing the R-matrix technique, the continuum basis set must be able to describe both maxima and nodes at the boundary radius of the R-matrix sphere, R_{mat}. As a consequence, R_{lim} must be chosen larger than R_{mat}. In practice, we found for R_{mat} in the range 10–13 a_0 (the only values thoroughly tested) that $R_{\mathrm{lim}} = R_{\mathrm{mat}} + 2\,a_0$ is the best choice for both Bessel and Coulomb functions (see details in Section 3).

2.3. Initial selection of exponents

The minimisation problem is highly non-linear and characterised by having many local minima. This makes the final fit sensitive to the starting point. In our procedure, the initial Gaussian exponents α_i can be read directly as input data. Alternatively, we implemented two different methods of selection. In the first one, the initial exponents are selected randomly within an appropriate range ([0.01, 0.5] is the default). In the second, recommended method, a geometric series is used to generate the initial, even-tempered [14], set of exponents:

$$\alpha_i = \beta \times \gamma^i; \quad i = 1, 2, 3, \ldots, n. \tag{5}$$

In the present work, the default values are $\beta = 0.016$ and $\gamma = 1.39$ for both Bessel and Coulomb functions. These values were obtained by fitting and averaging (over the different l values) typical final sets of exponents. The use of Eq. (5) as an initial guess was found to minimise the required CPU time.

3. Sample results and discussion

In this section we present GTO basis sets optimised with GTOBAS for the representation of Bessel and Coulomb functions. These continuum basis sets have been employed in electron–molecule scattering calculations [15–18] using the UK R-matrix polyatomic codes [10]. Bessel and Coulomb functions with values $l = 0, 1, \ldots, 4$ were

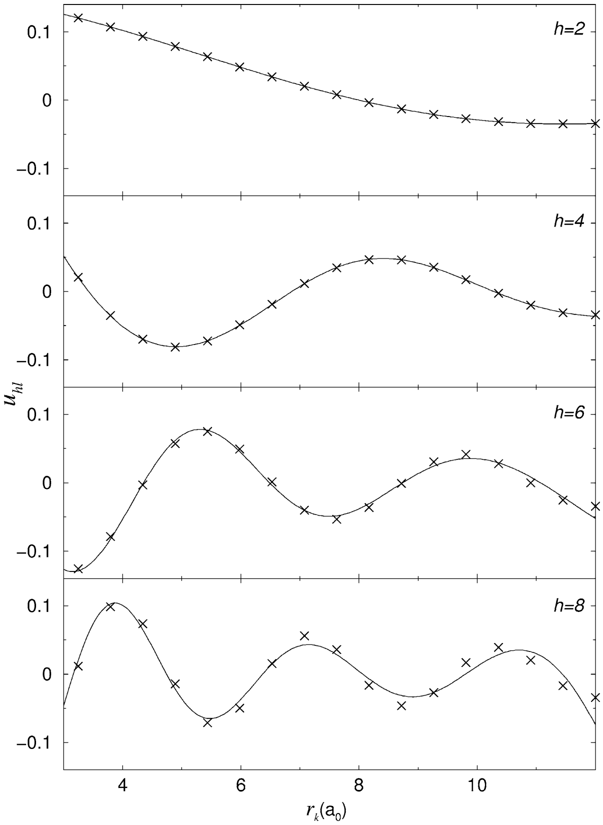

Fig. 1. Sample Bessel functions with $l = 0$ (crosses) and their fit by a linear combination of 9 Gaussian functions (full curves) for $R_{\mathrm{lim}} = 12\,a_0$ and $E_{\mathrm{up}} = 4$ Ryd. h indicates the specific function being plotted; the total number of nodes for each function is $h - 1$. The exponents of the GTOs are those in the first column in Table 1.

generated with NUMCBAS. The main input parameters, namely R_{lim}, E_{up} and the number n of GTOs, have been determined by the needs of the molecular systems studied and by the limitations of the computer resources. The radial mesh was chosen to be finer at the shorter ranges. Sensible changes in the mesh have little influence on the final value of the Gaussian exponents (typically less than 5%). The value of R_{lim} is fixed by the value of the R-matrix radius (see below). The upper energy, E_{up}, then determines the number of numerical functions to be fitted and therefore, the number of GTOs which must be bigger than or equal to the number of numerical functions. Note that there are no restrictions on the upper values of R_{lim} and E_{up} other than the required computational time (e.g., up to 70 minutes for $R_{\mathrm{lim}} = 20\,a_0$, $E_{\mathrm{up}} = 7$ Ryd and 17 GTOs).

Figs. 1 and 2 show, respectively, $l = 0$ Bessel and Coulomb functions and their corresponding fits. The region of best fit, $0 < r_k < 3$, is omitted from the figures for clarity. Although functions with $l = 0$ are the hardest to fit, it can be seen that, in general, the agreement is very good. As expected, when the number of nodes increases, the quality of the fit decreases. Besides, as r_k increases, the quality of the fit decreases too. A number of weighting procedures were tested to improve the fits at large r_k, without success. However, in R-matrix calculations, only the region $r_k \leqslant R_{\mathrm{mat}}$ is considered; therefore, the quality of the fit for $r_k > R_{\mathrm{mat}}$ is not relevant. The accuracy of the fit can be assessed by checking the final value of the minimisation function $F_{l,N}$ (see Eq. (3)): for the fits presented in Figs. 1, 2, $F_{l,N}$ was found to be smaller than 4×10^{-3}.

When using the R-matrix method, the set of continum functions has to be complete within the finite region of the R-matrix sphere ($0 < r_k < R_{\mathrm{mat}}$). A basic difficulty is that this can cause problems of linear dependence with

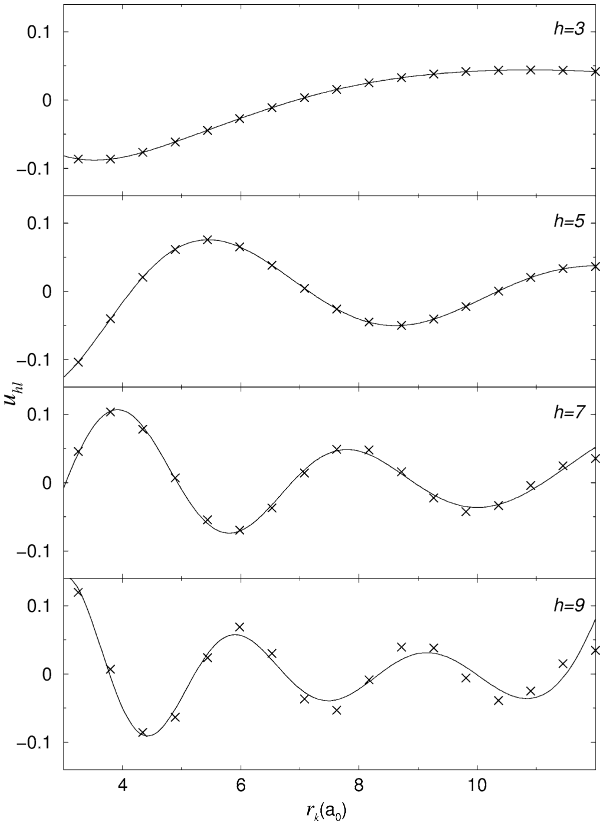

Fig. 2. Sample Coulomb functions with $l = 0$ (crosses) and their fit by a linear combination of 12 Gaussian functions (full curves) for $R_{\mathrm{lim}} = 12\,a_0$ and $E_{\mathrm{up}} = 5$ Ryd. h and number of nodes as in Fig. 1. The exponents of the GTOs are those in the first column in Table 3.

the functions representing the target [4]. If the boundary radius R_{lim} defined in NUMCBAS is much larger than the R-matrix sphere radius R_{mat}, the continuum basis set can become overcomplete in the inner region defined by this sphere, leading to strong linear dependence problems. After several tests with R_{mat} in the range 10–13 a_0, we concluded that the best choice for the boundary radius (in terms of linear dependence) is $R_{\mathrm{lim}} = R_{\mathrm{mat}} + 2\,a_0$, for both neutral and singly charged targets. Using a smaller R_{lim} would produce a poor fit for $r_k \simeq R_{\mathrm{lim}}$. In their paper, Nestmann and Peyerimhoff suggested $R_{\mathrm{lim}} = 20\,a_0$ for $R_{\mathrm{mat}} = 10\,a_0$ but we found that using this value lead to severe linear dependence.

GTO basis sets for the representation of Bessel and Coulomb functions are given in Tables 1–3. Using a much larger number of GTOs than numerical functions can lead to some exponents being zero and/or more than one converged set. If the latter is the case, the quality of the different converged sets is very similar, at least in terms of its adequacy for scattering calculations. Finally, it should be noted that the fitting procedure given here also provides contraction coefficients for the GTO continuum basis. However, to maximise the flexibility of the basis in the R-matrix calculations, our practice, and that of Nestmann and co-workers [5–7], has been to leave these functions uncontracted.

3.1. Neutral targets

Table 1 compares our basis set for a neutral target and $R_{\mathrm{mat}} = 10\,a_0$ with that obtained by Sarpal et al. [6] who also used the method of Nestmann and Peyerimhoff [5]. The differences between the two sets may arise from the

Table 1
Optimised GTO exponents for Bessel functions with $R_{\text{lim}} = 12\,a_0$ and $E_{\text{up}} = 4$ Ryd. N is the total number of Bessel functions for each l. Numbers in parentheses correspond to the basis set obtained by Sarpal et al. [6]

	s $(l = 0)$	p $(l = 1)$	d $(l = 2)$	f $(l = 3)$	g $(l = 4)$
$i \backslash N$	8	7	7	6	6
1	0.285726	0.122916	0.125926	0.172844	0.109047
	(0.108108)	(0.130267)	(0.111252)	(0.170320)	(0.111363)
2	0.192361	0.090430	0.094783	0.125071	0.083773
	(0.095095)	(0.102083)	(0.089412)	(0.130520)	(0.089501)
3	0.133124	0.067262	0.072171	0.092858	0.064982
	(0.078078)	(0.080605)	(0.072361)	(0.111586)	(0.072433)
4	0.092965	0.049879	0.054821	0.069374	0.050149
	(0.063430)	(0.063430)	(0.058496)	(0.089591)	(0.058555)
5	0.064850	0.036582	0.041223	0.051579	0.038123
	(0.049698)	(0.049698)	(0.047005)	(0.072505)	(0.047052)
6	0.044862	0.026327	0.030441	0.037788	0.028129
	(0.038336)	(0.038336)	(0.037397)	(0.058614)	(0.035035)
7	0.030582	0.018347	0.021766	0.026876	
	(0.029180)	(0.029180)	(0.029225)	(0.047099)	
8	0.020403				
	(0.021923)				
9	0.013159				
	(0.013013)				

Table 2
Optimised GTO exponents for Bessel functions with $R_{\text{lim}} = 15\,a_0$, $E_{\text{up}} = 5$ Ryd for $l = 0, 1, 2$, $E_{\text{up}} = 4$ Ryd for $l = 3$ and $E_{\text{up}} = 2.4$ Ryd fort $l = 4$. N is the total number of Bessel functions for each l

	s $(l = 0)$	p $(l = 1)$	d $(l = 2)$	f $(l = 3)$	g $(l = 4)$
$i \backslash N$	11	10	10	8	6
1	0.396150	0.200280	0.127610	0.092326	0.065228
2	0.282361	0.151623	0.101449	0.073837	0.051981
3	0.206334	0.117236	0.081731	0.059756	0.041790
4	0.152887	0.091465	0.066052	0.048426	0.033499
5	0.114144	0.071562	0.053288	0.039083	0.026564
6	0.085455	0.055913	0.042273	0.031270	0.020594
7	0.063902	0.043480	0.034058	0.024670	
8	0.047570	0.033545	0.026815	0.018994	
9	0.035148	0.025590	0.020778		
10	0.025705	0.019210	0.015690		
11	0.018550	0.014051			
12	0.013150				
13	0.009071				

use of a different boundary condition. Besides, Sarpal et al. use the same E_{up} but do not specify their value of R_{lim}. The most significant differences between the two sets are: (i) for $l = 0$, our basis set has 3 exponents that are bigger than their biggest one; (ii) for the other l values, the smallest exponents in our basis set are smaller than in their basis set.

In order to assess the validity of our basis set and how it compares with that of Sarpal et al., we used both sets to study the electron-impact electronic excitation of H_2O [16] and CF_2 [17]. Our basis set was found to give better eigenphases for both systems, as seen for the electron–H_2O collision in Fig. 3, where we have plotted the eigenphase sum of symmetry $^2A_1$ (in a variational calculation, a higher eigenphase indicates better results). The three plotted eigenphases are very similar in the energy region $0 < E < 5$ eV but differences increase at energies

Table 3
Optimised GTO exponents for Coulomb functions with $R_{\mathrm{lim}} = 12a_0$, $E_{\mathrm{up}} = 5$ Ryd for $l = 0, 1, 2, 3$ and $E_{\mathrm{up}} = 3$ Ryd for $l = 4$. N is the total number of Coulomb functions for each l

	s ($l = 0$)	p ($l = 1$)	d ($l = 2$)	f ($l = 3$)	g ($l = 4$)
$i\backslash N$	9	9	8	8	5
1	40.9539	4.73096	0.186838	0.176575	0.096767
2	3.19022	0.559667	0.136427	0.133689	0.071821
3	0.460592	0.381904	0.101837	0.102897	0.053743
4	0.292295	0.268948	0.076389	0.079368	0.039814
5	0.194782	0.193043	0.057060	0.060885	0.028686
6	0.132316	0.140037	0.042150	0.046164	
7	0.090364	0.101932	0.030561	0.034350	
8	0.061483	0.073983	0.021460	0.024755	
9	0.041395	0.053241			
10	0.027417	0.037780			
11	0.017749	0.026258			
12	0.011096	0.017649			

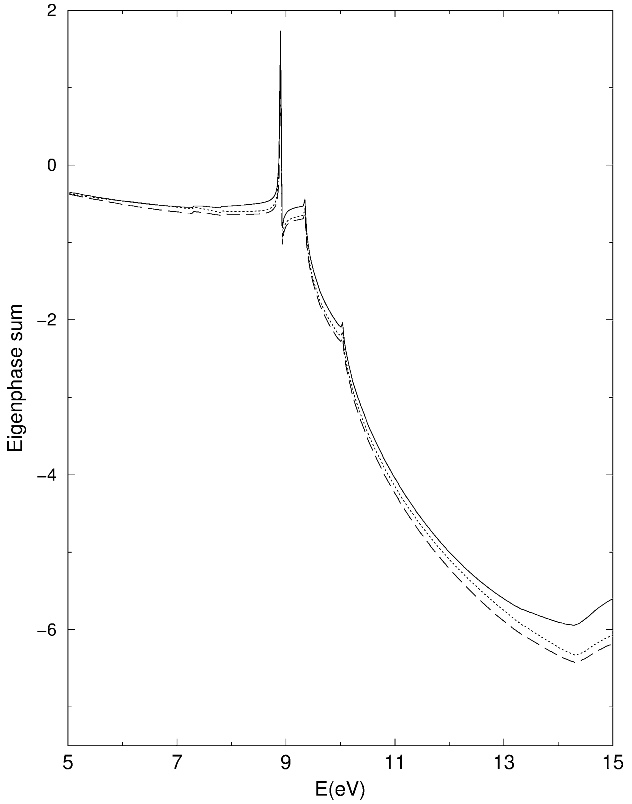

Fig. 3. Eigenphase sum of symmetry $^2\mathrm{A}_1$ for electron–$\mathrm{H_2O}$ collisions. Full line: calculation for $R_{\mathrm{mat}} = 10a_0$ using basis set from Table 1; long-dashed line: $R_{\mathrm{mat}} = 10a_0$, using Sarpal et al.'s basis set; dotted line: calculation for $R_{\mathrm{mat}} = 13a_0$ using basis set from Table 2. For a detailed description of these calculations, see [16].

above the first excitation threshold. Additionally, after performing several tests with different basis sets for both targets, we found that Sarpal et al.'s is more prone to linear dependence with the target basis set. This behaviour is, of course, target dependent.

Table 2 gives Gaussian exponents for the representation of Bessel functions generated with $R_{\mathrm{lim}} = 15\,a_0$. To adapt the size of the basis set to the computational limits imposed by the R-matrix code, we chose different E_{up} for different l values. In this way, we reduced the number of GTOs required for fitting the functions with higher l. The exponents, with exception of the first ones for $l = 0$ and $l = 1$, are smaller than those obtained with $R_{\mathrm{lim}} = 12\,a_0$, as expected. This basis set has been used for H_2O calculations with an R-matrix radius of $13\,a_0$ [16]. The resulting eigenphase sum and cross sections were found to be in very good agreement with those obtained for $R_{\mathrm{mat}} = 10\,a_0$ using the basis set from Table 1 (see Fig. 3), thus proving the adequacy of this basis set. The increasing discrepancies at higher energies are probably due to the incompleteness of the basis set of Table 2 for $R_{\mathrm{mat}} = 13\,a_0$.

3.2. Ionic targets

Gaussian exponents for singly charged targets are listed in Table 3. It can be noticed that for $l = 0$ and $l = 1$ some exponents are larger than unity. Such large values are consistent with the high amplitudes of $l = 0, 1$ Coulomb functions at short-range. Note that in this case, an even-tempered initial selection is not the most appropriate.

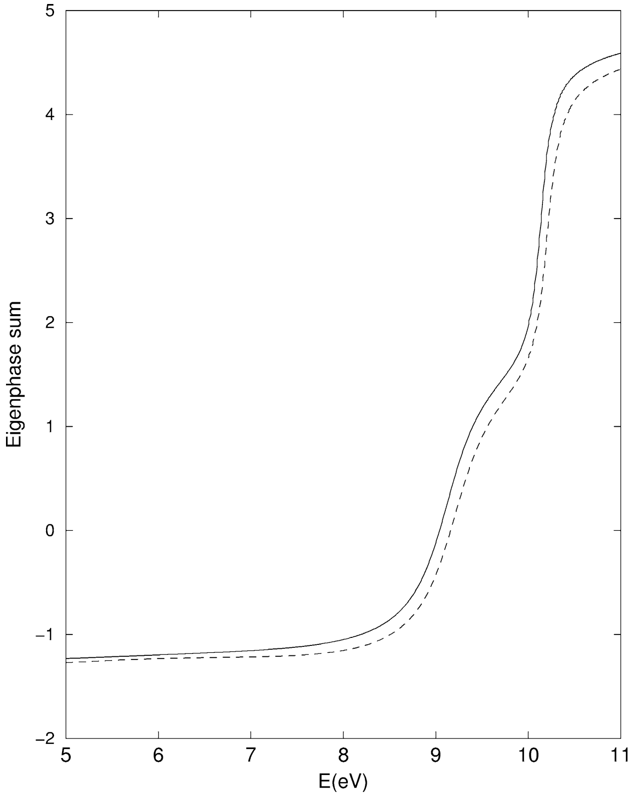

Fig. 4. Eigenphase sum of symmetry $^2A_1$ for electron–H_3^+ collisions. Full line: calculation for $R_{\mathrm{mat}} = 10\,a_0$ using basis set from Table 3; dashed line: $R_{\mathrm{mat}} = 10\,a_0$, using Sarpal et al.'s basis set. For a detailed description of these calculations, see [18].

The basis set presented in Table 3 has been used to study the electron-impact excitation of HCO^+ [15], H_3^+ and H_3O^+ [18]. In contrast to Bessel functions, there are no previous studies on the representation of Coulomb functions by GTOs. There is, however, a study on the dissociative recombination of H_3^+ by Orel and Kulander [19], where the scattering calculations are based on the complex Kohn variational method. In this approach, analytic Coulomb functions can be used. At the equilibrium geometry of the ion (for $^2A_1$ symmetry), Orel and Kulander found that the two first resonance energies (widths) are 9.1 (0.64) and 10.3 (0.18) eV. We computed these resonances using the R-matrix method and we obtained, respectively, 9.12 (0.64) and 10.14 (0.19) eV. This very good agreement with the results of Orel and Kulander indicates the reliability of our representation of the continuum. In order to further illustrate the influence of the continuum basis set on a scattering calculation, we have computed the eigenphase sum for the e–H_3^+ collision in $^2A_1$ symmetry using our basis set (Table 3) and that of Sarpal et al. (1996), which was optimised to represent Bessel functions. The results, plotted in Fig. 4, show that our basis set gives better (higher) eigenphases, as expected. They also show that Sarpal et al.'s basis set gives somewhat oscillatory behaviour at lower energies (not plotted).

4. Programs organisation and data input

4.1. Organisation of GTOBAS

The subroutines in GTOBAS are mainly taken from Numerical Recipes [20]. Further comments are given within the code.

`READNUM` reads the numerical continuum functions produced by NUMCBAS. It can be user-adapted to read any other input continuum functions.
`FUNCS` computes a Gaussian-type function.
`FUNCP` computes the function $F_{l,N}$ (Eq. (3)) to minimise.
`SVDFIT` performs a least-square fit by use of a singular value decomposition technique to obtain the coefficients c_{hi} of the Gaussian expansion (see Section 2.2). It is taken from Numerical Recipes [20].
`SVBKSB` and `SVDCMP` are used by `SVDFIT` to perform the singular value decomposition [20].
`POWELL` performs the minimisation of the function $F_{l,N}$ (Eq. (3)). It is taken from Numerical Recipes [20].
`LINMIN` implements the one-dimensional line minimisation used by the `POWELL` subroutine [20].
`MNBRAK` is used by `LINMIN` to bracket a one-dimensional minimum [20].

4.2. Organisation of NUMCBAS

`BASIS` performs the calculation of the numerical basis.
`SEARCH` is used by `BASIS` to search for the eigensolutions to the differential equations (see Eq. (1)).
`BASFUN` controls the numerical integration and normalises the resulting wavefunctions.
`DEVGL` is a de Vogelaere integration routine [11] used by `BASFUN`.
`FINDER` locates the eigenvalue with a given number of nodes.
`WRHEAD` writes the header of file **lunumb** which will contain the numerical basis in a form suitable for GTOBAS.

4.3. Input data for GTOBAS

Input data [with defaults in brackets] is read from standard input via a namelist /FIT/. The data type is indicated in the following way: variables starting with (a–h,o–z) are double precision and those starting with (i–n) are integers.

beta [1.6D–02]
Even-tempered β coefficient, see Eq. (5) (used if **iguess**=2).

expo Vector containing initial exponents. Only needed if **iguess**=0. Maximum size is 20.

ftol [1.D–09]
Convergence parameter used in `POWELL` subroutine.

gamma [1.39D0]
Even-tempered γ coefficient, see Eq. (5) (used if **iguess**=2).

iguess [2]
Determines whether initial exponents will be read as input data (0), generated randomly (1), or generated using an appropriate function (2) (see Section 2.3).

iprint [0]
Print flag for additional output. 1: All iteration data; 2: iteration data plus mesh.

iswmol3 [18]
Logical unit for the output of final exponents. The format is that of the Sweden-Molecule program [21, 22] used by the UK molecular R-matrix polyatomic code.

lunumb [13]
Logical unit for the input of numerical functions to be fitted.

luplot [17]
Logical unit for the output fitted functions. Numerical functions are also written to this file. The format allows plotting with xmgr.

noexp [0]
Number of exponents to be used. Must be bigger or equal to the number of numerical functions to be fitted. Only needed if **iguess**=1 or 2.

nplot [0]
Print flag: no functions (0) / all functions (1) (numerical and fitted) are saved to unit luplot.

rdlow [1.D–02]
Lower limit of the random selection (used if **iguess**=1).

rdup [0.49D0]
Upper limit of the random selection (used if **iguess**=1).

4.4. Input data for NUMCBAS

Input data is read from standard input via a namelist /INPUT/. The size of the array parameters is indicated by the number of default values in brackets. The data type is indicated in the same way as for GTOBAS.

charge [0.D0]
Effective charge of the target (see Section 2.1).

ecmax [10.D0]
E_{up}, upper bound for eigenvalues in Rydberg.

hrx [0.01, 0.02, 0.02605, 7*0.D0]
Vector of size **nix** containing the step length to be used in each subrange.

ibug [3*0]
Vector containing switches for extra printed output: ibug(1)=1 potential as a function of radial coordinate r_k; ibug(2)=1 final eigensolutions as function of r_k; ibug(3)=1 brief summary of data written to **lunumb**.

irx [30, 120, 500, 7*0]
Vector of size **nix** defining the number of mesh points to end of each subrange (must be divisible by 2). The program checks that the last point in the last subrange is $\simeq$ **rlim**. If it is not, a new **irx**(**nix**) is calculated.

Table 4
Tests data input for the numerical functions and corresponding fits represented in Figs. 1 and 2

	BESSEL functions	COULOMB functions

NUMCBAS	&INPUT
	title = ‘Neutral target’,
	lval = 0,
	ecmax = 4.00D0,
	rlim = 12.0,
	charge = 0.0D0, /
		&INPUT
		title = ‘Ionic target’,
		lval = 0,
		ecmax = 5.00D0,
		rlim = 12.0D0,
		charge = 1.0D0, /

GTOBAS	&FIT
	expo = 6.0, 3.0, 0.7, 0.3, 0.1, 0.07, 0.05, 0.02, 0.01,
	iguess = 0,
	nplot = 1,
	iprint = 0, /
		&FIT
		noexp = 12,
		iguess = 2,
		nplot = 1,
		iprint = 0, /

lunumb [13]
Logical unit for output of numerical functions.

lval [0]
Angular momentum of the numerical functions to be calculated.

nix [3]
Number of subranges with different step. Maximum value is 10.

rlim [10.D0]
R_{lim}, radius where the boundary condition is applied.

tiny [1.D–11]
Convergence threshold for eigenvalues calculation.

title Character*80 variable containing title for output.

5. Test data

Table 4 gives sample test data. Use of these data should give the exponents for $l = 0$ presented in Tables 1 and 3.

Acknowledgements

We thank Bernd Nestmann for helpful discussions on his work, and Peter Chocian and Andrew Richardson for work on the early stages of this project. This work was supported by a Marie Curie Fellowship of the European Community programme Human Potential (contract number HPMF-CT-1999-00415) and by EPSRC.

References

[1] P.W. Atkins, R.S. Friedman, Molecular Quantum Mechanics, 3rd edn., Oxford University Press, New York, 1997.
[2] See Extensible Computational Chemistry Environment Basis Set Database, http://www.emsl.pnl.gov:2080/forms/basisform.html, mantained by the Environmental and Molecular Sciences Laboratory, Pacific Northwest Laboratory.
[3] K. Kaufmann, W. Baumeister, M. Jungen, J. Phys. B: At. Mol. Opt. Phys. 22 (1989) 2223.
[4] J. Tennyson, L.A. Morgan, Philos. Trans. Roy. Soc. London A 357 (1999) 1161.
[5] B.M. Nestmann, S. Peyerimhoff, J. Phys. B: At. Mol. Opt. Phys. 23 (1990) L773.
[6] B. Sarpal, K. Pfingst, B.M. Nestmann, S.D. Peyerimhoff, J. Phys. B: At. Mol. Opt. Phys. 29 (1996) 857.
[7] T.M. Beyer, B.M. Nestmann, B.K. Sarpal, S.D. Peyerimhoff, J. Phys. B: At. Mol. Opt. Phys. 30 (1997) 3431.

[8] W.H. Klink, G.L. Payne, J. Comput. Phys. 21 (1976) 208.
[9] S.A. Salvini, Thesis, Queen's University of Belfast, 1983.
[10] L.A. Morgan, J. Tennyson, C.J. Gillan, Comput. Phys. Commun. 114 (1998) 120.
[11] R. de Vogelaere, J. Res. Nat. Bur. Standards B 54 (1955) 119.
[12] J.P. Coleman, J. Mohamed, Comput. Phys. Commun. 17 (1979) 283.
[13] M.J.D. Powell, Comput. J. 7 (1964) 155.
[14] D.L. Cooper, S. Wilson, J. Chem. Phys. 77 (1982) 5053.
[15] A. Faure, J. Tennyson, Monthly Notices Roy. Astronom. Soc. 325 (2001) 443.
[16] J.D. Gorfinkiel, J. Tennyson, J. Phys. B: At. Mol. Opt. Phys. (2002), in press.
[17] I. Rozum, J. Tennyson, J. Phys. B: At. Mol. Opt. Phys., submitted.
[18] A. Faure, J. Tennyson, J. Phys. B: At. Mol. Opt. Phys., submitted.
[19] A.E. Orel, K.C. Kulander, Phys. Rev. Lett. 71 (1993) 4315.
[20] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes in Fortran, Cambridge University Press, 1995.
[21] J. Almlöf, P.R. Taylor, in: C.E. Dykstra (Ed.), Advanced Theories Computational Approaches to Electronic Structure of Molecules, Reidel, Dordrecht, 1984.
[22] J. Almlöf, P.R. Taylor, in: E. Clementi (Ed.), Modern Techniques in Computational Chemistry: MOTECC-91, Escom, Leiden, 1991, Ch. 6A.

```
              Program NUMCBAS


**********************************************************************
*                                                                    *
*  ANY  NEUTRAL MOLECULE (L=0) RMAT =12.0                            *
*                                                                    *
**********************************************************************


  LUNUMB  = 13     Output file for the basis


       INTEGRATION MESH INPUT DATA

  NIX   =  3     No. of integration regions with different step-sizes

      I       IRX                HRX
      1        30          0.1000000000D-01
      2       120          0.2000000000D-01
      3       500          0.2605000000D-01


       NUMERICAL BASIS CALCULATION INPUT DATA

  LVAL   =       0     Angular Momentum
  CHARGE =     0.0     Effective charge
  RMAT   =    12.0     R-matrix boundary radius


       SEARCHING PROCEDURE PARAMETERS

  ECMAX  =    4.00     Maximum energy for the eigensolutions
  BTOL   =0.20D+00     Iteration starting tolerance
  TINY   =0.10D-10     Eigensolution convergence parameter


                 SUMMARY TABLE

  Partial wave L =  0   No. of eigensolutions  =   8

              Nodes    Eigenenergy (Ryd.)
      1         0       0.17137586D-01
      2         1       0.15423828D+00
      3         2       0.42843966D+00
      4         3       0.83974176D+00
      5         4       0.13881446D+01
      6         5       0.20736482D+01
      7         6       0.28962526D+01
      8         7       0.38559578D+01
```

```
            Program GTOBAS

 Numerical basis read by READNUM

 Angular momentum LVAL = 0

 Selected poles

      Seqno  Lval  Energy
    1    1    0    0.0171
    2    2    0    0.1542
    3    3    0    0.4284
    4    4    0    0.8397
    5    5    0    1.3881
    6    6    0    2.0736
    7    7    0    2.8963
    8    8    0    3.8560

 Boundary radius                      = 12.0
 Number of radial mesh points         =  500
 Number of exponents to be optimized  =    9
--------------------------------------------

 Initial exponents
  1  0.222400D-01
  2  0.309136D-01
  3  0.429699D-01
  4  0.597282D-01
  5  0.830222D-01
  6  0.115401D+00
  7  0.160407D+00
  8  0.222966D+00
  9  0.309923D+00

 Total Number of iterations = 14

 Final minimisation function =  0.380402D-02

 Final exponents
  1  0.285726D+00
  2  0.192361D+00
  3  0.133124D+00
  4  0.929651D-01
  5  0.648495D-01
  6  0.448617D-01
  7  0.305822D-01
  8  0.204030D-01
  9  0.131594D-01

 Writing   8 functions to unit 17 for plotting

 Saving exponents in SWMOL3 format to unit 18
```

```
        Program NUMCBAS

**********************************************************************
*                                                                    *
*  Ionic target (L=0) RMAT =12.0                                     *
*                                                                    *
**********************************************************************


  LUNUMB  = 13     Output file for the basis


       INTEGRATION MESH INPUT DATA

  NIX   =  3     No. of integration regions with different step-sizes

      I       IRX                 HRX
      1        30          .1000000000D-01
      2       120          .2000000000D-01
      3       500          .2605000000D-01


       NUMERICAL BASIS CALCULATION INPUT DATA

  LVAL   =       0     Angular Momentum
  CHARGE =     1.0     Effective charge
  RMAT   =    12.0     R-matrix boundary radius


       SEARCHING PROCEDURE PARAMETERS

  ECMAX  =    5.00     Maximum energy for the eigensolutions
  BTOL   = .20D+00     Iteration starting tolerance
  TINY   = .10D-10     Eigensolution convergence parameter
  NODMAX =       8     Last eigensolution within the energy range


                 SUMMARY TABLE

  Partial wave L =  0   No. of eigensolutions  =   9

            Nodes    Eigenenergy (Ryd.)
      1         0      -.10000044D+01
      2         1      -.25686944D+00
      3         2      -.96844407D-01
      4         3       .23281734D+00
      5         4       .73260852D+00
      6         5       .13822880D+01
      7         6       .21762751D+01
      8         7       .31120134D+01
      9         8       .41881205D+01
```

```
            Program GTOBAS

 Numerical basis read by READNUM

 Angular momentum LVAL = 0

 Selected poles

      Seqno  Lval  Energy
    1    1    0   -1.0000
    2    2    0    -.2569
    3    3    0    -.0968
    4    4    0      .2328
    5    5    0      .7326
    6    6    0    1.3823
    7    7    0    2.1763
    8    8    0    3.1120
    9    9    0    4.1881

 Boundary radius                      = 12.0
 Number of radial mesh points         =  500
 Number of exponents to be optimized  =   12
--------------------------------------------

 Initial exponents
  1   .222400D-01
  2   .309136D-01
  3   .429699D-01
  4   .597282D-01
  5   .830222D-01
  6   .115401D+00
  7   .160407D+00
  8   .222966D+00
  9   .309923D+00
 10   .430792D+00
 11   .598801D+00
 12   .832334D+00

 Total Number of iterations = 20

 Final minimisation function =   .301797D-02

 Final exponents
  1   .409539D+02
  2   .319022D+01
  3   .460592D+00
  4   .292295D+00
  5   .194782D+00
  6   .132316D+00
  7   .903639D-01
  8   .614834D-01
  9   .413946D-01
```

```
10   .274171D-01
11   .177489D-01
12   .110964D-01

Writing   9 functions to unit 17 for plotting

Saving exponents in SWMOL3 format to unit 18
```