JOURNAL OF COMPUTATIONAL PHYSICS 88, 15–30 (1990)

Symbolic Computation and Automatic Fortran Code Generation for Eigenvalue Determination by Phase Integral Method

JUSSI LUPPI AND PETRI PAJUNEN

Department of Physical Chemistry, University of Helsinki,
Meritullinkatu 1 C, 00170 Helsinki, Finland

Received November 18, 1988; revised April 25, 1989

An automatic FORTRAN code generation and determination of energy eigenvalues for periodic potentials with wells and barriers are presented. These problems serve as test cases in developing tools and techniques for generation of numerically efficient FORTRAN software and for more general, efficient and flexible ways of programming. Several problems in different areas of science are described, and the advantages of the approach as well as limitations of the tools presently available are discussed.

I. INTRODUCTION

The standard procedure for many scientific computation problems, such as determining eigenvalues of Schrödinger equation, is to combine numerical FORTRAN library routines with the scientist's own program to perform numerically intensive subtasks such as numerical integration. By the mere availability of subroutines for a large variety of different purposes, FORTRAN has secured its status as the predominant language in scientific computations. However, FORTRAN has some serious drawbacks which are difficult to cure. It is admittedly somewhat tedious to program and, more importantly, inclusion of branching such as IF–THEN structures in the code must be paid for by loss of efficiency and, indeed, in many cases is totally impractical. The present article addresses these problem in the form of practical applications to the computation of eigenvalues for periodic potentials by phase integral methods. However, the aim is to develop techniques to combine the computational power of numerical FORTRAN software with more efficient and more general ways of programming provided by symbolic software. The general characterization of the problems that are relevant here is numerically intensive; that is, the most expensive part of the work is the actual execution of the code. If interactive development and programming are the time consuming parts of the task, there already exist better alternatives to FORTRAN such as Matlab or Speakeasy, for instance.

The phase integral quantization conditions for periodic potentials clearly exhibit the weaknesses of FORTRAN mentioned above. While the phase integral method [1–5] is a computationally efficient technique for calculating the eigenvalues of the one-dimensional Schrödinger equation, the phase integral quantization condition

0021-9991/90 $3.00

for a periodic potential function, such as that for torsional vibration, is a transcendental equation involving phase integrals across the wells and barriers of the potential function, and is therefore potential-dependent; each different number of wells and barriers requires a new slightly different quantization condition. Therefore, a FORTRAN computer code for the present task, determination of eigenvalues of arbitrary problem that has any number of wells and barriers would be impossible in principle and in practice, with the number of wells and barriers limited to realistic maximum values, it would at best be very clumsy and extensive. However, there is an elegant solution to this dilemma provided by recent developments in symbolic computer languages and based on the fact that all various quantization conditions for problems with different number of wells and barriers can be written in a form of a matrix equation. A combination of a symbolic language, MACSYMA in the present case, and the existing efficient quadrature routines for evaluation of the phase integrals [6–8] written in FORTRAN, allow most of the logic to be handled under MACSYMA, while the final time consuming numerical computation is performed by FORTRAN. Furthermore, the FORTRAN window facility of MACSYMA, supported by some locally developed auxiliary routines, provides tools for automatic FORTRAN code generation. While the project started from the above-mentioned difficulty with the phase integral quantization conditions, it is now clear that by transferring all possible logic under MACSYMA, extremely efficient FORTRAN code can be generated. In practice, the number of logical decisions and the number of DO loops can be reduced considerably, a result which yields very simple code which compilers and optimisers can fully exploit. While the FORTRAN code can be saved for later use, it is actually treated as a disposable utility and the programmer never writes a single line of FORTRAN code, apart from the work in the past when the FORTRAN codes for the phase integral quadrature routines were generated.

In the following, the physical problem of determining eigenvalues for various potentials by the phase integral method, is considered in Section II and the results are presented in Section III. Section IV then describes in some detail the structure and organization of the FORTRAN code generation. The discussion in Section V finally points out the benefits of employing symbolic computer languages for FORTRAN code generation.

II. PHASE INTEGRAL QUANTIZATION CONDITIONS

The phase integral quantization condition for present problem, that is for a potential $V(\phi)$ of periodicity of 2π with m different wells and barriers, is well known [9], and is obtained by following changes in the coefficients (C_+, C_-) in the phase integral wave function of the form

$$\Psi(\phi) = [q(\phi)]^{-1/2} \left[C_+ \exp\left(+i \int_c^{\phi} q(\phi)\, d\phi \right) + C_- \exp\left(-i \int_c^{\phi} q(\phi)\, d\phi \right) \right], \quad (1)$$

where $q(\phi)$ is the phase function and c the phase reference point, subject to boundary conditions

$$\begin{aligned}\Psi(0) &= \Psi(2\pi),\\ \Psi'(0) &= \Psi'(2\pi).\end{aligned} \tag{2}$$

The phase reference point is always chosen to be a classical turning point, defined as a zero of $(E - V(\phi))$. For a potential well with classical turning points at c and b, Eq. (1) may also be written as

$$\Psi(\phi) = [q(\phi)]^{-1/2} \left[B_+ \exp\left(+i \int_b^{\phi} q(\phi)\, d\phi \right) + B_- \exp\left(-i \int_b^{\phi} q(\phi)\, d\phi \right) \right] \tag{3}$$

so that the connection from b to c is given by

$$\begin{pmatrix} C_+ \\ C_- \end{pmatrix} = \mathbf{B} \begin{pmatrix} B_+ \\ B_- \end{pmatrix}, \tag{4}$$

where

$$\mathbf{B} = \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{-i\alpha} \end{pmatrix} \tag{5}$$

and

$$\alpha = i \int_b^c q(\phi)\, d\phi. \tag{6}$$

In the phase integral approximation, the function $q(\phi)$ consists of contributions of various orders

$$q(\phi) = q^{(1)}(\phi) + q^{(3)}(\phi) + \cdots. \tag{7}$$

The first-order term is

$$q^{(1)}(\phi) = [(E - V(\phi))/F(\phi)]^{1/2}, \tag{8}$$

where $F(\phi)$ is the effective rotational constant, and the functional form of all higher order terms is such that the first-order and all higher order phase integrals of Eq. (6) may be expressed in the general form

$$I_k(E) = \int_\Gamma f_k(\phi)/[E - V(\phi)]^{k+(1/2)}\, d\phi \tag{9}$$

where k is an integer, -1, 0, 1, 2,..., and $f_k(\phi)$ some analytic combination of $F(\phi)$, $V(\phi)$, and their derivatives. The integration path Γ is a closed loop in the complex plane which encircles the two turning points under consideration and avoids the turning points where the integrand has nonintegrable singularities for $k > 0$. The single valuedness of the integrand requires branch cuts from each of the turning points and for the integrals of Eq. (9) the cut joins the two turning points. The evaluation of these, in general complex, phase integrals by an efficient and reliable quadrature procedure is described in Refs. [6–8].

The matrix connecting the coefficients of the wave functions with reference points at two turning points associated with a potential barrier, c and d say, may be written as [9]

$$\mathbf{C} = e^{\beta} \begin{pmatrix} [1 + e^{-2\beta}]^{1/2} & -i \\ i & [1 + e^{-2\beta}]^{1/2} \end{pmatrix}, \tag{10}$$

where again the phase integrals across the barrier

$$\beta = i \int_c^d q(\phi)\, d\phi \tag{11}$$

may be computed routinely.

Repeated application of the connection formulae above and application of the boundary condition, Eq. (2), then allow the general quantization condition to be written as

$$\det |\mathbf{C}_1 \mathbf{B}_1 \mathbf{C}_2 \mathbf{B}_2 \cdots \mathbf{C}_m \mathbf{B}_m - \mathbf{1}| = 0, \tag{12}$$

where the matrices **B** and **C** depend on the phase integrals across the wells and barriers, respectively, of the potential. However, while Eq. (12) is very general and compact, it is not amenable for numerical computation. For this purpose, for a fixed m, the determinant can be transformed into a transcendental expression involving the relevant phase integrals. For illustration, Eqs. (13)–(15) display the quantization condition function for a problem with two different wells and barriers, three different wells and barriers, and a symmetric case of three wells and barriers, respectively.

$$e^{(-\beta(2)-\beta(1))}(\sqrt{e^{(2\beta(1))}+1}\,\sqrt{e^{(2\beta(2))}+1}\cos(\alpha(2)+\alpha(1))+\cos(\alpha(2)-\alpha(1)))-1 \tag{13}$$

$$\begin{aligned} &e^{(\beta(3)-\beta(2)-\beta(1))}(\sqrt{e^{((2\beta(1))}+1}\,\sqrt{e^{(2\beta(2))}+1}\,\sqrt{e^{(2\beta(3))}+1}\cos(\alpha(3)+\alpha(2)+\alpha(1)) \\ &\quad +\sqrt{e^{(2\beta(3))}+1}\cos(\alpha(3)+\alpha(2)-\alpha(1))+2\sqrt{e^{(2\beta(1))}+1}\cos(\alpha(3)-\alpha(2)+\alpha(1)) \\ &\quad +\sqrt{e^{(2\beta(2))}+1}\cos(\alpha(3)-\alpha(2)-\alpha(1))-1 \end{aligned} \tag{14}$$

$$e^{(-3\beta(1))}(\sqrt{e^{(2\beta(1))}+1}(e^{(2\beta(1))}+1)\cos(3\alpha(1)))+3\sqrt{e^{(2\beta(1))}+1}\cos(\alpha(1))-1, \tag{15}$$

where the indices 1, 2, 3 denote each additional different well or barrier.

It is obvious that a FORTRAN code that contained quantization conditions for all possible variations of symmetry and values of m would clearly be very clumsy indeed. To avoid this, a very compact MACSYMA program based on Eq. (12) can have the desired value of m and appropriate information about symmetry as input.

It would then generate the quantization condition and the FORTRAN code required to compute the desired eigenvalues, thus reducing the programming work considerably.

The present scheme carries these ideas even further, as there is no need to specify the value of m. The program starts by reading the input information about the potential function under MACSYMA, which is subsequently utilized to analyze the potential and the first task is to determine the number of wells and barriers. Equations (14) and (15), which have the same number of wells and barriers but different periodicity, show that, apart from the periodocity of 2π, any additional periodicity or symmetry of the potential simplifies the quantization condition and reduces final numerical computation. Therefore, symbolic tests for these properties, such as if $V(\phi)=V(\phi+\pi)$ is true for a higher periodicity or, if $V(\phi)=V(-\phi)$ is true for even symmetry, are performed in order to simplify the quantization condition as much as possible.

On the other hand, the graphical output for the potential, determination of the appropriate energy range based on the positions and maxima and minima values of wells and barriers of the problem, and initial values for classical turning points, which are crucial for the phase integral method, require numerical calculations which at this stage are also carried out under MACSYMA. Good initial values for the classical turning points are obtained by choosing the energy slightly on top of all barriers and starting a Newton–Raphson search for complex turning points with small imaginary component and the real part equal to the position of the barrier top.

The number of wells and barriers and the symmetry of the problem determine the appropriate phase integral quantization condition for the problem, which is then generated with MACSYMA. The FORTRAN window facility and some locally developed auxiliary tools are used to automatically generate the FORTRAN code which performs major part of numerical computation. One crucial choice in the whole procedure is the allocation of the work between symbolic and numerical software. In the present application, MACSYMA is used to calculate, simplify, and generate the quantization condition and the FORTRAN code but the final quadrature for the phase integrals is more efficiently performed by numerical quadrature routines written in FORTRAN.

While the FORTRAN code may be saved for later use, in principle it is treated as a disposable utility; for each execution of the program a new code is generated, compiled, linked, and executed. This is justified because, for numerically intensive tasks, the final execution of the code is responsible for the major part of the required computer time. For codes such as in the present applications, the required total CPU time is not significant and the main advantage of the approach is the speed and ease of programming as the user never has to write a single line of FORTRAN by himself. The output from the program consists of numerical tables of energy eigenvalues or transition frequencies for spectroscopical applications, together with graphical outputs showing the potential and all the energy levels, and the phase integral quantization condition as a function of energy. The latter provides an illustration of the pattern of energy levels which facilitates interpretations regarding the degeneracies of the states.

III. RESULTS

Vibrational spectroscopy features many potentials with wells and barriers, such as those associated with restricted rotation [10], or those involved with unimolecular conformer interconversion reactions between species which differ by a rotation around a single C–C, C–O, or C–S bond [11]. While more extensive applications to torsional vibrations are published elsewhere [12], a sample problem of each type, illustrating the versatility and clarity of presentation of the present procedure, is presented here. The first potential considered is the asymmetric torsional potential function governing internal rotation about the C–C bond of chloroacetyl fluoride [13]. The model potential for this problem is of the form

$$V(\phi) = \sum_{i=1}^{6} (V_i/2)(1 - \cos(i\phi)), \tag{16}$$

where the torsional coordinate is ϕ and the foldness of the barrier is i. For an asymmetric potential function, the internal rotation constant $F(\phi)$ varies as a function of the internal rotation angle and this dependence on the dihedral angle is approximated by a Fourier series,

$$F(\phi) = F_0 + \sum_i F_i \cos(i\phi). \tag{17}$$

The values for parameters used in Eqs. (16) and (17) are listed in Table I, and the potential energy function and the energy levels are shown in Fig. 1, with energy eigenvalues and transition frequencies listed in Table II. It is clear that all the lower energy levels of the problem are located in the deep well around the torsion angle equal to zero and are, therefore, nondegenerate levels associated with the *trans*

TABLE I

Potential Parameters V_i $(i = 1, 6)$ and Internal Rotation Constant Parameters F_i $(i = 0, 6)$ (in cm^{-1}) for Chloroacetyl Fluoride

V_i	Value	F_i	Value
		F_0	1.066340
V_1	350.0	F_1	0.136518
V_2	306.0	F_2	0.102467
V_3	420.0	F_3	−0.000174
V_4	44.0	F_4	0.008006
V_5	0.	F_5	−0.000576
V_6	2.0	F_6	0.000696

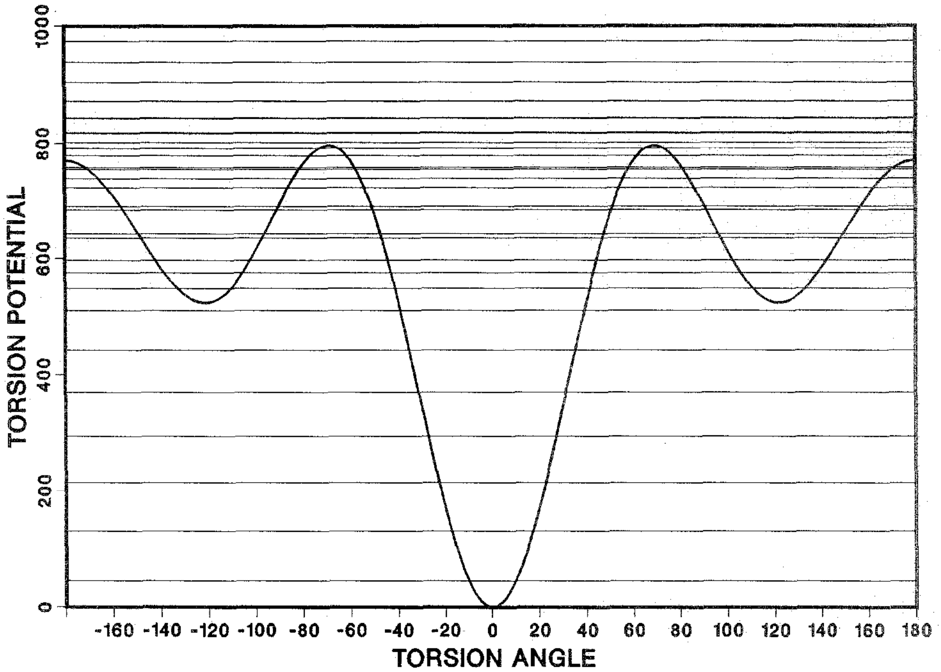

FIG. 1. Schematic drawing of the potential and computed energy levels (in cm^{-1}) for chloroacetyl fluoride.

conformer. This can be verified from Fig. 2 which plots the left-hand side of the appropriate phase integral quantization condition function as function of energy; a zero value of the function plotted corresponds to an eigenlevel. The lower zeros of this function at energies listed in Table II are clearly simple zero associated with sin-like function crossings associated with the simple phase integral quantization condition for a potential well. However, at higher energies the quantization condition is a function involving phase integrals across all the wells and barriers of the problem implying more variety in the behaviour of the higher levels. For instance, at energies above 540 cm^{-1}, there is a possibility of finding doubly degenerate levels associated with the *gauche* conformer lying in the two shallow wells. These levels can easily be identified from the behaviour of the quantization condition function. See, for instance, the levels at 575 and 549 cm^{-1}. The zero of the quantization condition function for the former level is a normal crossing associated with a simple zero but the quantization condition at 549 cm^{-1} is clearly a parabola corresponding to double zero and thus a *gauche* level lying in the two shallow wells. A detailed investigation of the quantization condition function clearly allows the identification of nondegenerate *trans* levels and doubly degenerate *gauche* levels.

The second potential considered here is the symmetric threefold potential governing the internal rotation of the methyl group in acetaldehyde, obtained on

TABLE II

Energy Levels E_v and Transition Frequencies (in cm^{-1}) for Chloroacetyl Fluoride

v	E_v	Conformer	Transition	Frequency
0	44.23	Trans		
1	130.70	Trans	86.48	
2	213.87	Trans	83.16	
3	293.61	Trans	79.74	
4	369.80	Trans	76.19	
5	442.28	Trans	72.48	
6	510.85	Trans	68.57	
7	549.00	Gauche		
8	575.68	Trans	64.83	
9	596.86	Gauche		47.86
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
12	684.11	Gauche		42.09
13	689.87	Trans	54.76	
14	722.20	Gauche		38.08
15	738.57	Trans	48.70	
16	745.07	Gauche		31.87

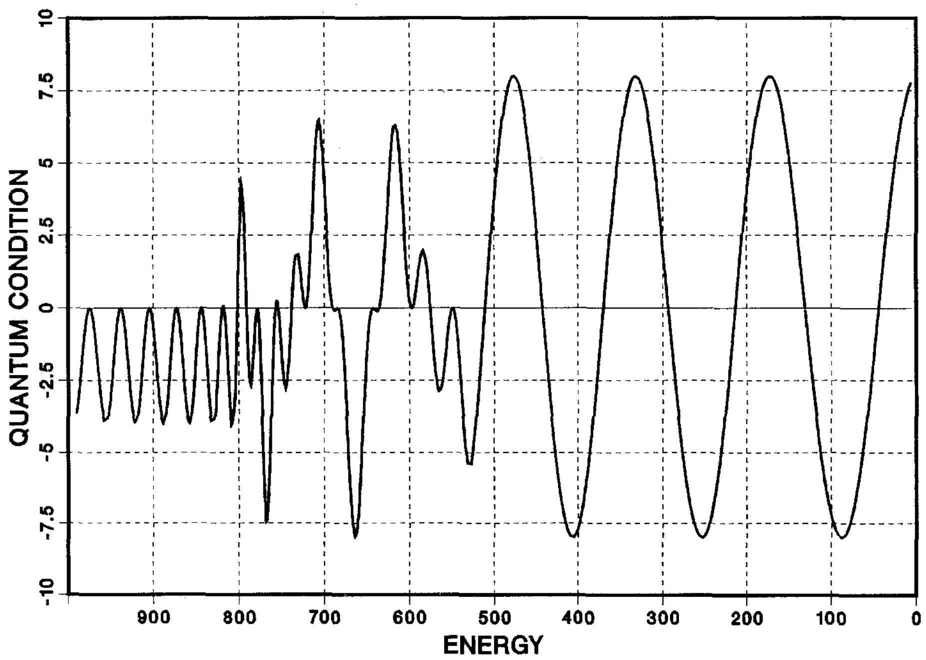

FIG. 2. Schematic drawing of the phase integral quantization condition as function of energy (in cm^{-1}) for chloroacetyl fluoride.

TABLE III

Energy Levles E_v (in cm^{-1}) for Acetaldehyde

v	E_v	Level	Symmetry
0	77.52	A	
1, 2	77.52	E	
3, 4	221.93	E	
5	223.77	A	
6	332.79	A	
7, 8	347.09	E	
9, 10	426.07	E	R
11	482.97	A	R
12	496.92	A	R
13, 14	581.36	E	R

substituting $i = 3$ in Eq. (16). The energy levels listed in Table III computed from the parameter values $V_3 = 400.5$ cm^{-1} and $F_0 = 7.476$ cm^{-1} agree very well with experimental results [14]; this potential and the resulting energy levels are shown in Fig. 3. Due to the threefold symmetry of the potential, it is now expected that the lowest level should be triply degenerate, and indeed the quantization condition function of Fig. 4 clearly exhibits third-order polynomial behaviour at 77.52 cm^{-1}. The next two groups of zeros show how the splitting due to tunneling sets in. The

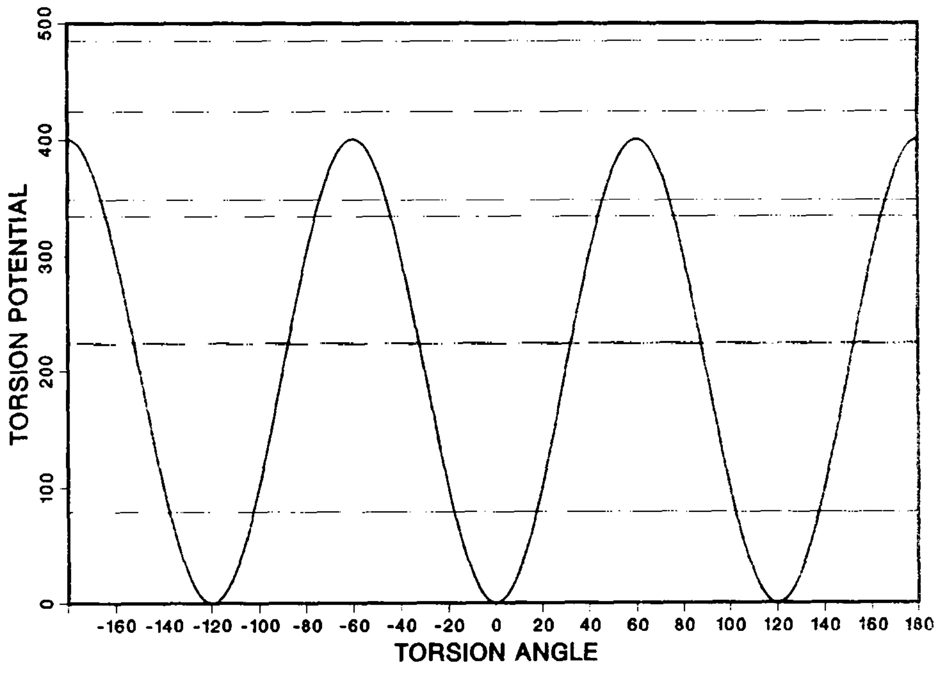

FIG. 3. Schematic drawing of the potential and computed energy levels (in cm^{-1}) for acetaldehyde.

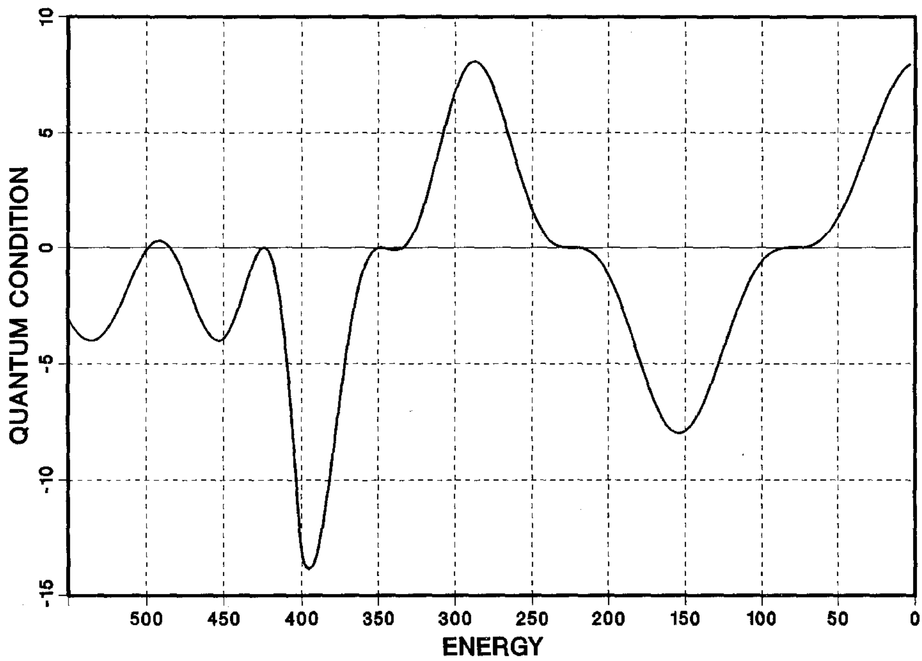

FIG. 4. Schematic drawing of the phase integral quantization condition as function of energy (in cm^{-1}) for acetaldehyde.

energy levels in Fig. 3 at 222 cm^{-1} are very close to each other and consequently the quantization condition curve in Fig. 4 looks very much like the lowest levels. The group of levels near 350 cm^{-1}, on the other hand, clearly consists of a non-degenerate *A* level, a line crossing at lower energy, followed by a doubly degenerate pair of *E* levels, the now familiar parabola. While strictly speaking, the splitting due to tunneling must already exist for the lowest group of levels, the splitting is too small to be detected experimentally, and the present calculation which is based on the phase integral approximation which is a medium-accuracy procedure designed to strike a balance between efficiency and accuracy consistent with the experimental spectral resolution. Finally, above all the barriers, the alternating doubly degenerate *E* and non-degenerate *A* free rotor levels *R* with the structure *E*, *A*, *A*, *E*, *E*,... can be recognized.

The third potential studied here is the Coffey–Evans potential used to model dipolar coupling in polarizable liquids such as crystal displays [15]. This potential has the form

$$V(\phi) = -2b\cos(2\phi) + b^2 \sin^2(2\phi), \tag{18}$$

where the parameter b, typically in the range 0 to 50, measures the strength of the coupling. For $b = 20$, the potential and the resulting eigenlevels are shown in Fig. 5 while the energies are listed in Table IV. From the potential, it is again expected

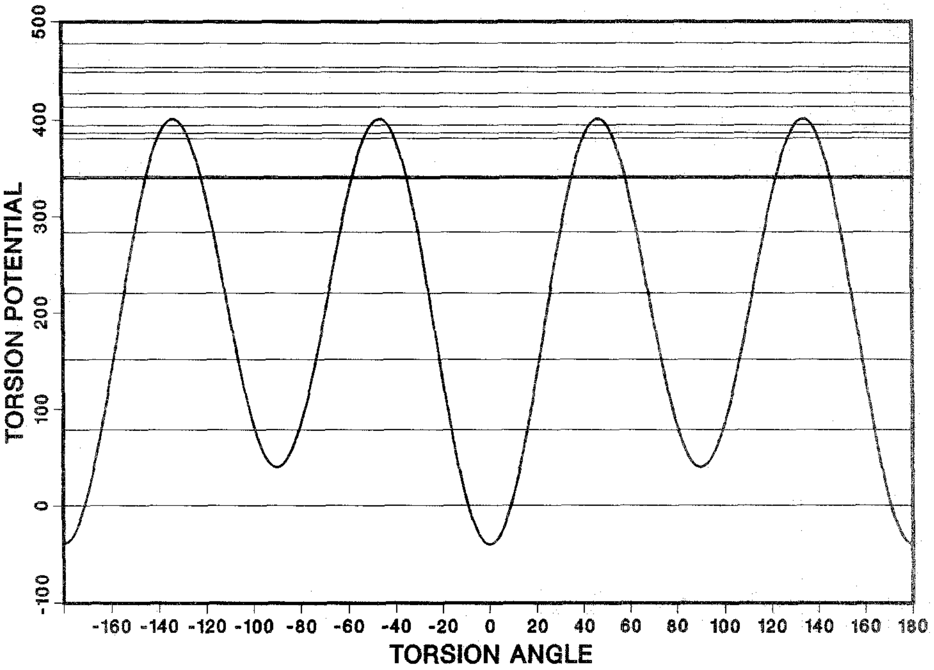

FIG. 5. Schematic drawing of the potential and computed energy levels for Coffey–Evans equation with $b=20$.

that the lowest energy level should be a doubly degenerate level lying in the deeper wells, but associating the other energy levels with the correct member of the two different pairs of wells requires investigation of the quantization condition function shown in Fig. 6. From this figure, however, it is immediately evident that all the other levels at energies below 350, are actually fourfold degenerate, within the present resolution. The difference between the parabola for the lowest level and the quartic curve for the others is obvious. While two sets of doubly degenerate levels are expected, this fourfold degeneracy is an accidental result of the coincidence of

TABLE IV

Eigenvalue Clusters E_n of Coffey–Evans Equation with $b=20$

n	E_n	n	E_n
0	0.00	7	384.92
1	77.83	8	390.99
2	151.36	9	426.53
3	220.06	10	452.50
4	283.15	11	477.64
5	339.49	12	507.51
6	379.94	13	540.63

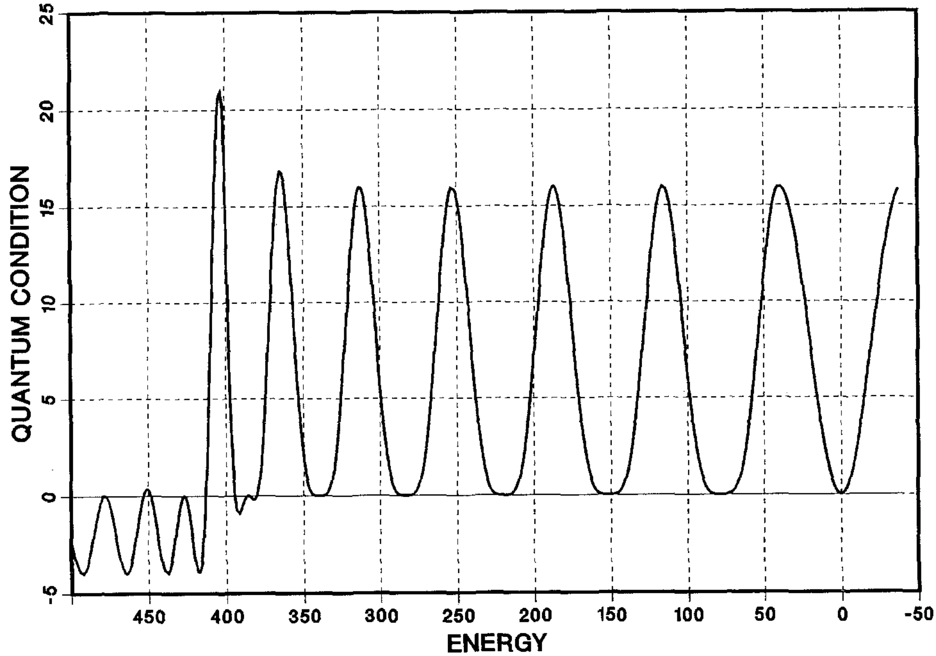

FIG. 6 Schematic drawing of the phase integral quantization condition as function of energy for Coffey–Evans equation with $b = 20$.

the positions of the eigenvalues in the two different types of wells [16]. The exact eigenvalues for this problem with slightly more restricted boundary conditions [17] were computed by two general purpose routines [18–19] up to $b = 30$, and although the clustering of eigenvalues was noticed no qualitative explanation for the clustering which is immediately available from the present calculation was provided by the exact method. Table IV lists the mean positions of those level clusters. A calculation with $b = 50$, shown in Fig. 7 and in Table V, illustrates the robustness of the phase integral calculation. In contrast, the NAG routine starts to

TABLE V

Eigenvalue Clusters E_n of Coffey–Evans Equation with $b = 50$

n	E_n	n	E_n
0	0.50	8	1458.94
1	198.15	9	1618.80
2	391.71	10	1772.29
3	581.31	11	1919.52
4	766.45	12	2059.29
5	947.01	13	2190.81
6	1122.84	14	2310.64
7	1293.82	15	2413.33

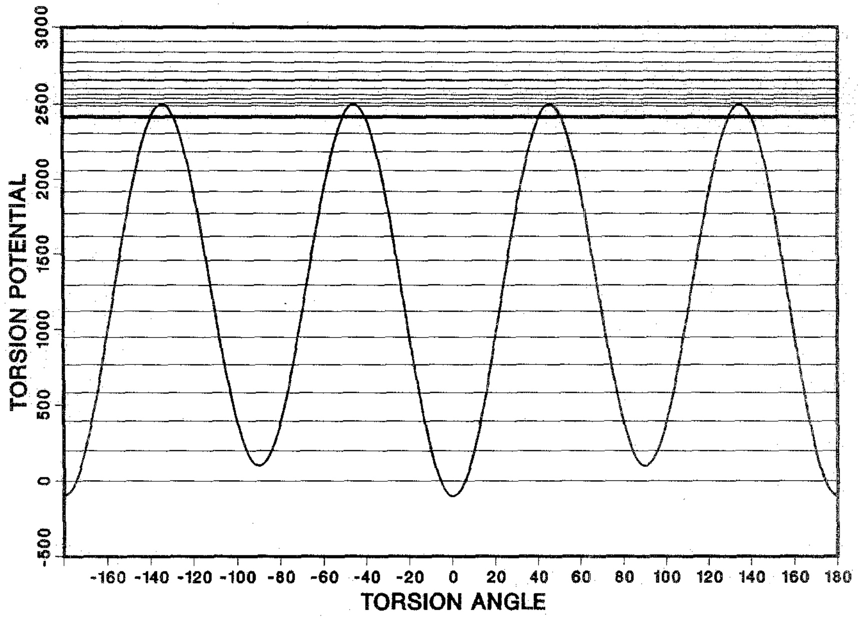

FIG. 7. Schematic drawing of the potential and computed energy levels for Coffey–Evans equation with $b = 50$.

experience numerical difficulties at $b = 30$ due to high barriers [16]. This is not the case for he phase integral calculation which is actually very accurate and efficient for levels deep in the wells or high above the tops of all barriers. The only regions where the present method experiences difficulties are the immediate neighbourhoods of barrier maxima or well minima where the accuracy is somewhat reduced and phase correction functions must be introduced. However, the present test cases show that the accuracy, typically four to five significant figures, often is sufficient for all practical purposes, and that the phase integral method due to its [illegible] and reliable, although not a high-accuracy picture of the whole range of spectrum.

In the above, no reference to the actual order of the phase integral approximation used has been made because this is based solely on computational considerations; the cost of computation versus the accuracy obtained. While calculations were carried out in first and third order, all the results presented here are first-order results.

IV. THE FORTRAN CODE GENERATION

The FORTRAN window facility available for our use in this work is rather primitive. The FORTRAN command of MACSYMA [20], when applied to its argument, indents the line six or more columns and transforms MACSYMA expressions into Fortran expressions, adding continuation lines as necessary. It recognizes the usual mathematical functions and ordinary mathematical operations, but has no intimate knowledge of FORTRAN syntax. For instance, while it understands 1.0, it does not consider 1.d0 to be a number, and a command such as FORTRAN(GOTO 11) "breaks" the window. The indentation of at least six columns is annoying because addresses cannot be placed directly in the correct columns.

As a result of the above, for automatic FORTRAN code generation, several auxiliary MACROS were developed to manipulate the MACSYMA code and transform it into a form accepted by the FORTRAN command. Due to these tools, the programming which is now carried out under MACSYMA is much quicker and more flexible than previously under FORTRAN. The general structure of these tools can be written as COMMAND(LIST)$ where LIST, in the spirit of LISP programming lanquage, consists of members which, in turn, may be numbers, strings, lists themselves, MACROS, etc. A simple example of a useful tool is the one used to analyse input. Some lists are allocated for input, and a special tool investigates them. If a particular member of the list is symbolic, the tool creates a READ command to appropriate place in the FORTRAN code to be generated but if the entry is numeric it is used as a numerical constant possibly generating a PARAMETER statement in the FORTRAN code.

As examples of more sophisticated MACROS, a DO LOOP MACRO will create a DO loop. The argument list for this MACRO may contain a DO loop by itself, and hence a flexible generation of stacking DO loops is possible. Other necessary tools are MACROS associated with GOTO and OPEN commands, and MACROS to generate data files. The addresses which are vital parts of FORTRAN programming can have logical names in MACSYMA, and the actual replacement of the logical names by FORTRAN type numeric addresses, which is carried out by another MACRO, has to be made at a late stage of the procedure due to possible stacking of DO loops. The MACSYMA program is now a combination of ordinary MACSYMA commands and the tools described above. These lines are first arranged in a list, the members of which may be lists as well. In order to prepare this list for final FORTRAN code generation, the recursive lists are expanded to yield a simple list. The result from this procedure carried out by a tool collected from MATRIXMAP, FORTRAN, and WRITEFILE commands of MACSYMA, is a FORTRAN source file which after some housekeeping, such as moving the lines containing addresses to their proper columns, can be compiled, linked, and executed.

V. Discussion

In order to develop tools and techniques for efficient and flexible generation of FORTRAN software for numerically intensive problems, a test case of the determination of eigenvalues for periodic potentials has been presented. Extension to other boundary condition problems is straighforward. A combination of facilities provided by MACSYMA and locally developed auxiliary tools allow automatic generation of FORTRAN code without any need to resort to FORTRAN programming. The existing, efficient and tested FORTRAN library subroutines can then be employed to carry out the numerically intensive subtasks of the problem, while logical decisions and mathematical manipulations are performed under MACSYMA. As a result, the generation of very simple but computationally extremely efficient FORTRAN code becomes possible.

The idea to perform the mathematical manipulations in MACSYMA prior to the generation of the FORTRAN code is not new [21, 22]. However, the present approach of developing auxiliary routines for all parts of programming work provide completely new levels of possibilities for FORTRAN programming. It is possible to write extremely complicated mathematical expressions with full confidence that the expressions and the final FORTRAN code are free from errors. If editing is required, the changes can take place in MACSYMA and are generated effortlessly to all appropriate places. The ease of the present style of programming also makes it possible to generate very straightforward code which the compilers and optimisers can make the most of. At the same time, this may present new challenges to the compilers and optimisers which are designed to work with the present style of codes. For example, during this work, several times compilers have refused to handle FORTRAN statements because of their complexity. This is trivial but understandable, because in the past it would have been diffult to code such expressions without making mistakes.

While the present work deals with numerically intensive problems, development of other similar tools will certainly make any kind of FORTRAN programming much quicker, saving programming effort. For example, FORTRAN libraries can have the codes written in symbolic form so that editing so as to implement changes in updating the library can be implemented with just a few changes in the symbolic language level.

The combination of the possibilities of symbolic computation and the efficiency of numerical FORTRAN software provides an enormous computing power and, as with the advent of FORTRAN which superceded actual manipulation of machine instructions, the new approach reduces the explicit use of FORTRAN, thus setting up a new hierarchy level of computer languages.

REFERENCES

1. H. JEFFREYS, *Proc. London Math. Soc* [2] **23**, 428 (1925).
2. G. WENTZEL, *Z. Phys.* **39**, 518 (1926).
3. H. A. KRAMERS, *Z. Phys.* **39**, 828 (1926).
4. L. BRILLOUIN, *Compt. Rend.* **183**, 24 (1926); *J. Phys. Radium* **7**, 353 (1926).
5. N. FRÖMAN, *Ark. Fys.* **32**, 541 (1966).
6. J. LUPPI AND P. PAJUNEN, *J. Chem. Phys.* **77**, 1505 (1982).
7. J. LUPPI AND P. JAJUNEN, *J. Chem. Phys.* **81**, 1836 (1984).

8. J. Luppi, *J. Comput. Phys.* **71**, 111 (1987).
9. W. H. Miller, *J. Chem. Phys.* **48**, 1951 (1968).
10. *Internal Rotation in Molecules*, edited by W. J. Orville-Thomas (Wiley, London, 1974).
11. M. Räsänen, H. Kunttu, and J. Murto, *Laser Chem.* **8**, 123 (1988).
12. T. Raaska, H. Kunttu, M. Räsänen, J. Luppi, and P. Pajunen, to be published.
13. J. R. Durig, W. Zhao, D. Lewis, and T. S. Little, *J. Chem. Phys.* **89**, 1285 (1988).
14. A. Bauder and Hs. H. Günthard, *J. Mol. Spectrosc.* **60**, 290 (1976).
15. M. W. Evans, W. T. Coffey, and J. D. Pryce, *Chem. Phys. Lett.* **63**, 133 (1979).
16. M. S. Child and A. V. Chambers, *J. Chem. Phys.* **92**, 3122 (1988).
17. J. D. Pryce, *NAG Newsletter (Oxford)* **3**, 4 (1986).
18. Numerical Algorithms Group, Fortran Library, Routine Documents D02KAF, D02KDF, D02KEF, par. 11, 1984 (unpublished).
19. P. B. Bailey, M. K. Gordon, and L. F. Shampine, *ACM Trans. Math. Software* **4**, 193 (1978).
20. *VAX UNIX MACSYMA Reference Manual*, Version 11 (Symbolics, Boston, 1985).
21. P. S. Wang, *J. Symb. Comput.* **2**, 305 (1986).
22. J. Čížek and F. Vinette, *Int. J. Quant. Chem. Sympos.* **22**, 537 (1988).