1984

A general transformation method to solve the Schroedinger equation.

Edward Neilson. Brewer
University of Windsor

CANADIAN THESES

THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

Canada

A GENERAL TRANSFORMATION METHOD TO SOLVE THE SCHROEDINGER EQUATION

by

Edward Neilson Brewer

A Thesis
submitted to the Faculty of Graduate Studies
through the Department of
Physics in partial fulfillment
of the requirements for the Degree
of Master of Science at
The University of Windsor

Windsor, Ontario, Canada

802086

ABSTRACT

Some numerical methods of solution of the radial Schroedinger equation approximate the potential as a simple polynomial in numerous small intervals. In each interval, the solution to the Schroedinger equation is simple to find analytically. These interval solutions are then linked together by the continuity condition on the wave function and its derivative to form the overall solution. The accuracy of this solution depends on small interval size.

In the method proposed in this work, the Schroedinger equation is transformed exactly to a directly solvable 'model' equation (often the Airy equation). Although a given transformation may not be valid for the whole range of x, it will be valid over a large interval. Only a few such intervals need be linked together to form the overall solution. Note that the solution is theoretically exact regardless of the size of the intervals.

Solution of the transformation equation gives the transformation function z(x) which can be shown to be a monotonic function. A close relationship between the formulation of this method and the JWKB and uniform JWKB approximation methods comes from this result.

Since the solution of the model equation is known, effort is concentrated on the solution of the transformation equation. Three methods of solution of this equation are presented; the iteration method, the power series method, and the coupled differential equation method with the latter currently preferred.

Examples of results using this method of solution are given

for the harmonic oscillator potential, the Eckart potential, and the N-fold periodic ring potential. An explanation and a full listing of a FORTRAN computer program implementing this method follow.

Acknowledgements

I would like to thank Dr. R.K.B. Helbing for hooking me on this problem and for the enormous amount of patience, respect, and understanding he has given me in the course of work. I would also like to thank Mr. Tim Scholl for his frustrating work obtaining the Airy functions to high accuracy and Mr. Murray Darrach for his valiant attempt to do the same for the Parabolic cylinder functions.

Financial support from the University of Windsor and the Natural Sciences and Engineering Research Council is gratefully acknowledged.

Table of Contents

List of Tables

List of Figures

A General Transformation Method to Solve the Schroedinger Equation

Chapter 1

Introduction

A common method of solution of the radial Schroedinger equation involves approximating the potential in an interval by some polynomial[1]. These polynomials are chosen so that the solution to their corresponding Schroedinger equation is analytically soluble. Solutions in each interval are then linked together using the continuity conditions to form the overall wave function. Any desired accuracy can be achieved by the choice of sufficiently small intervals.

The simplest choice of a polynomial is a constant. In this case, the potential is replaced by a series of constant steps, the height of which is the average of the potential over that interval. In each interval, the Schroedinger equation becomes

$$\frac{\hbar^2}{2\mu}\frac{d^2}{dr^2}\Psi_0(r) + (E-V_0)\Psi_0(r) = 0 \qquad (1.1.1)$$

where $\Psi_0(r)$ is the radial wave function, μ is the reduced mass, and E is the energy. The quantity V_0 is the potential $V_{effective}(r)$ averaged over the interval. The solution is:

for $E>V_0$,

$$\Psi_0(r)=\Psi(r_1)\cos[k(r-r_1)] + \Psi'(r_1)k^{-1}\sin[k(r-r_1)] \qquad (1.1.2)$$

for $E<V_0$,

$$\Psi_0(r)=\Psi(r_1)\cosh[k(r-r_1)] + \Psi'(r_1)k^{-1}\sinh[k(r-r_1)] \qquad (1.1.3)$$

and for $E=V_0$

$$\Psi_0(r)=\Psi(r_1) + (r-r_1)\Psi'(r_1). \qquad (1.1.4)$$

where

$$k=(2\mu/\hbar^2)^{1/2}(E-V_0)^{1/2} \qquad (1.1.5)$$

and

$$k=(2\mu/\hbar^2)^{1/2}(V_0-E)^{1/2} \qquad (1.1.6)$$

The solution $\Psi_0(r)$ has a value of $\Psi(r_1)$ and a derivative value of $\Psi'(r_1)$ at one endpoint of the interval. These quantities are obtained from the solution of the previous interval. Thus the overall solution of the wave function is obtained by solving the intervals sequentially, starting at some $\Psi(r_B)$,$\Psi'(r_B)$ determined by a boundary condition at r_B.

Higher order polynomonials can be used to better approximate the potential. Linear polynomials are a common choice[2],[3]. In this case, the functions involved in the interval solutions are the independent Airy functions Ai and Bi. Once again, the overall wave function is found by solving the intervals outward from a boundary condition point.

The main advantage of these types of methods over other numerical methods[4],[5] is that fewer points and hence intervals need be evaluated to obtain the same accuracy in the wave function. This is a result of the oscillatory functions used in each interval which better approximate the frequently oscillatory wave function.

It seems then that the way to further improve the numerical solution is by reducing the number of intervals. Intuitively this would imply a "better" fit is needed for the potential in each interval.

In the method presented in this work, the potential in each interval is, in effect, fitted exactly. This is done by the

transformation of the Schroedinger equation into another equation with known oscillatory solutions. In this way, fewer intervals are needed to obtain the the overall wave function. This drastic improvement is partly offset by the extra calculations needed to solve the transformation.

Chapter 2

Theory of the Method

2.1 The Transformation Equations

Since the radial equation after the transformation $y=\Psi(r)/r$ and the one dimensional Schroedinger equation have the same form $y''(r)=f(r)y(r)$ the latter is used for simplicity. Calling the independent variable x, the one dimensional equation is

$$-\frac{\hbar^2}{2m}\frac{d^2}{dx^2}y(x) - (E-V(x))y(x) = 0 \qquad (2.1.1)$$

with E the energy and V(x) the potential. This can be written as

$$y'' = W(x)y \qquad (2.1.2)$$

where $W(x) = (2m/\hbar^2)(V(x)-E)$. This function W(x) has the same form as the potential V(x) but has been 'energy adjusted' (the zeros of W are the classical turning points at energy E of V). Because of this relationship, W(x) will be referred to as the 'potential' from now on.

For a general potential W(x), equation (2.1.2) is transformed into a 'model' equation with known solutions

$$\Psi(z)'' = g(z)\Psi(z) \qquad (2.1.3)$$

where z is a function of x and ' refers to differentiation by z. The function g(z) is chosen so that the solutions $\Psi(z)$ are well known. For example, g(z)=z has Airy functions as solutions.

To transform (2.1.2) to (2.1.3) first change the variable x to the variable z using

$$y' = dy/dx = (dz/dx)(dy/dz) = z'y' \qquad (2.1.4)$$

$$y'' = z''y' + (z')^2y'' \qquad (2.1.5)$$

so (2.1.2) becomes

$$(z')^2y'' + z''y' - W(x)y = 0 \qquad (2.1.6)$$

or

$$y'' + (z''/(z')^2)y' - (W(x)/(z')^2)y = 0 \qquad (2.1.7)$$

if $z' \neq 0$.

A general equation of the form $y''(z) + P(z)y'(z) + Q(z)y = 0$ can be transformed[6] to the 'normal equation' form of (2.1.3) by using the substitution

$$y(z) = \varphi(z)u(z) \qquad (2.1.8)$$

where

$$u(z) = \exp\left[-\tfrac{1}{2}\int P(z)\,dz \right] \qquad (2.1.9)$$

In this case $P(z) = (z''/(z')^2)$ so

$$u(z) = \exp\left[-\tfrac{1}{2}\int (z''/(z')^2)\,dz \right] \qquad (2.1.10)$$

and since $dz = z'dx$

$$u(z) = \exp(-\tfrac{1}{2} \ln z') \qquad (2.1.11)$$

so

$$u(z) = (z')^{-1/2} \qquad (2.1.12)$$

Thus using

$$y(z) = \varphi(z)(z')^{-1/2} \qquad (2.1.13)$$

and noting that

$$d(z')/dz = z''/z' \qquad (2.1.14)$$

the other derivatives of y are obtained

$$y' = \varphi' - \tfrac{1}{2}\varphi z''(z')^{-5/2} \qquad (2.1.15)$$

$$y'' = \varphi''(z')^{-1/2} - \varphi' z''(z')^{-5/2} - \tfrac{1}{2}\varphi z'''(z')^{-7/2} + (5/4)\varphi(z'')^2(z')^{-9/2} \qquad (2.1.16)$$

Substituting into (2.1.7) gives

$$0 = \psi'' + (z')^{-2}[-\tfrac{1}{2}((z'''/z') - (3/2)(z''/z')^2\} - W(x)]\psi \quad (2.1.17)$$

The quantity in the curly brackets is known as the Schwarzian[7] and denoted (z,x) .

A comparison of equation (2.1.3) with equation (2.1.17) yields the 'transformation' equation

$$W(x) + \tfrac{1}{2}(z,x) - g(z)(z')^2 = 0 \quad (2.1.18)$$

Using the function $u(z)$ defined in (2.1.12) the Schwarzian becomes

$$(z,x) = -2u''/u \quad (2.1.19)$$

so equation (2.1.18) can be expressed as

$$W(x) - u''/u - g(z)u^{-4} = 0 \quad (2.1.20)$$

The transformation function $z(x)$ is a solution of eqs. (2.1.12) and (2.1.20) simultaneously.[7] The transformation will be valid as long as $z'(x)\neq 0$. Within this region, the two indepenent solutions of (2.1.2) from (2.1.8) become

$$y_1(x) = u(x)\psi_1(z)$$
$$y_2(x) = u(x)\psi_2(z) \quad (2.1.21)$$

if ψ_1, ψ_2 are the two independent solutions of (2.1.3). These solutions are selected so that, asymptotically, y_1 decreases while y_2 increases 'exponentially' in the classically forbidden region. Thus within the region the general solution can be expressed as

$$y(x) = a_1y_1(x) + a_2y_2(x) \quad (2.1.22)$$

Changing the starting conditions yields solutions of $y(x)$ valid over different ranges of x. In this way, the entire x-axis can be covered by an overlapping set of solutions of $y(x)$ each valid only in a finite region and having different solutions of

$z(x)$, $u(x)$, $u'(x)$, and $\Psi(z)$. To obtain the overall unique wave function, the conditions of continuity of the wave function and its derivative are applied to $y(x)$ and $y'(x)$ at a 'match' point located between each pair of overlapping regions. To aid in this process, some new terminology is introduced in the next section.

2.2 Connection Matrices

In each region, the solution $y(x)$ from (2.1.21) and (2.1.22) can be expressed as

$$y(x) = a_1 u(x) \Psi_1(z) + a_2 u(x) \Psi_2(z) \qquad (2.2.22)$$

Now introduce a 'basis function' vector

$$\underline{b}(x) = (u(x) \Psi_1(z) , u(x) \Psi_2(z)) \qquad (2.2.23)$$

and a 'coefficient' vector

$$\underline{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \qquad (2.2.24)$$

Then equation (2.2.22) can be written as a dot product

$$y(x) = \underline{b}(x) \cdot \underline{a} \qquad (2.2.25)$$

Similarly, the derivative of $y(x)$ can be written

$$y'(x) = \underline{b}'(x) \cdot \underline{a} \qquad (2.2.26)$$

Now a vector $\underline{y}(x)$ can be constructed with the components $y(x)$ and $y'(x)$ to yield

$$\begin{bmatrix} y \\ y' \end{bmatrix} = \begin{bmatrix} y_1 & y_2 \\ y_1' & y_2' \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \qquad (2.2.27)$$

or

$$\underline{y} = \mathbb{B}\underline{a} \qquad (2.2.28)$$

Note that the x dependence is all in the matrix $\mathbb{B}$. The coefficient vector $\underline{a}$ is made up of constants.

A needed quantity will be $\mathbf{B}^{-1}$ = (det $\mathbf{B}$)$^{-1}$(adjoint $\mathbf{B}$). The determinant of $\mathbf{B}$ is

$$\det \mathbf{B} = y_1 y_2' - y_1' y_2 = Wr_x(y_1, y_2) \qquad (2.2.29)$$

where $Wr_x(y_1, y_2)$ is the Wronskian of the basis functions y_1, y_2 with respect to their argument x. The inverse matrix $\mathbf{B}^{-1}$ is now calculated as

$$\mathbf{B}^{-1} = \frac{1}{Wr_x(y_1, y_2)} \begin{bmatrix} y_1' & -y_2 \\ -y_1' & y_2 \end{bmatrix} \qquad (2.2.30)$$

It is clear from matrix algebra that

$$\det \mathbf{B}^{-1} = (Wr_x(y_1, y_2))^{-1} \qquad (2.2.31)$$

The relationship between the Wronskian of $y_1(x)$ and $y_2(x)$ with respect to x and the Wronskian of $\Psi_1(z)$ and $\Psi_2(z)$ with respect to z can be derived from (2.1.21)

$$\begin{aligned} Wr_x(y_1, y_2) &= u(x)\Psi_1 \frac{d}{dx}(u(x)\Psi_2(z)) - \frac{d}{dx}(u(x)\Psi_1(z))u(x)\Psi_2(z) \\ &= u\Psi_1(u'\Psi_2 + uz'\Psi_2^{\cdot}) - (u'\Psi_1 + uz'\Psi_1^{\cdot})u\Psi_2 \\ &= z'u^2(\Psi_1\Psi_2^{\cdot} - \Psi_1^{\cdot}\Psi_2) \end{aligned}$$

where $'$ refers to differentiation by x and $\cdot$ refers to differentiation by z. Thus

$$Wr_x(y_1, y_2) = z'u^2\, Wr_z(\Psi_1, \Psi_2) \qquad (2.2.32)$$

The continuity condition of the wave function and its derivative at some match point between regions n and n+1 located at $x=\xi^{(n+1,n)}$ can now be expressed as

$$\underline{y}_{n+1}(\xi^{(n+1,n)}) = \underline{y}_n(\xi^{(n+1,n)}) \qquad (2.2.33)$$

In terms of the coefficient vectors $\underline{a}$

$$\underline{a}_{n+1} = \mathbf{M}_{n+1,n}\underline{a}_n \qquad (2.2.34)$$

where

$$\mathbf{M}_{n+1,n} = \mathbf{B}_{n+1}^{-1}(\xi^{(n+1,n)})\,\mathbf{B}_n(\xi^{(n+1,n)}) \qquad (2.2.35)$$

is the 'match matrix' evaluated at the match point $x=\xi^{(n+1,n)}$. In explicit terms, the match matrix is

$$M_{n+1,n} = (1/\overline{Wr}) \begin{bmatrix} y_1\bar{y}_2' - y_1'\bar{y}_2 & \bar{y}_2'y_2 - \bar{y}_2y_2' \\ \bar{y}_1y_1' - \bar{y}_1'y_1 & \bar{y}_1y_2' - \bar{y}_1'y_2 \end{bmatrix} \qquad (2.2.36)$$

The bar on top of a symbol signifies that it refers to region n+1 while unbarred symbols refer to region n.

The determinant of the match matrix $M_{n+1,n}$ is

$$\det M_{n+1,n} = \det B_{n+1}^{-1} \det B_n$$

$$= \frac{z'u^2\, Wr_z(\Psi_1(z),\Psi_2(z))}{\bar{z}'\bar{u}^2\, Wr_{\bar{z}}(\Psi_1(\bar{z}),\Psi_2(\bar{z}))} \qquad (2.2.37)$$

Since $z'u^2=1$ from (2.1.12) then $\det M_{n+1,n}$ reduces to a ratio of the Wronskians of the neighbouring regions. If the two regions have the same model equation, as they are assumed to have in most of this work, the Wronskians are equal so

$$|\det M_{n+1,n}| = 1 \qquad (2.2.38)$$

In this work regional and global coordinates are used. While there is only one global coordinate x, each individual region n has its own regional coordinates x_n and z_n. Hence the match point is represented by the global coordinate $x=\xi^{(n+1,n)}$ and simultaneously by the two regional coordinates $x_n=\xi^{(n+1,n)}$ and $x_{n+1}=\xi^{(n+1,n)}$.

2.3 Eigenvalue Condition

Suppose N regions cover the x-axis. Then the relation between the coefficients of the end regions is

$$\underline{a}_N = M_{N,1}\, \underline{a}_1 \qquad (2.3.39)$$

where

$$M_{N,1} = M_{N,N-1}M_{N-1,N-2}\cdots M_{3,2}M_{2,1} \qquad (2.3.40)$$

In order to obtain a bound state energy eigenvalue, the wave function must vanish as $x \to \pm\infty$. At these extreme values of x, the wave function is in a classically forbidden region. Recall from section 2.1 that the solution $y_1(x)$ is chosen so that it exponentially decreases in classically forbidden regions. Thus the wave function in the limit as $x \to \pm\infty$ must be proportional to $y_1(x)$.

Without loss of generality, $y(x)$ can be chosen in region 1 as

$$y(x) = y_1(x) \qquad (2.3.41)$$

so

$$\underline{a}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad (2.3.42)$$

In region N, the wave function $y(x)$ must be chosen proportional to $y_1(x)$. If R is the constant of proportionality then $y(x)$ in this region becomes

$$y(x) = Ry_1(x) \qquad (2.3.43)$$

so

$$\underline{a}_N = \begin{bmatrix} R \\ 0 \end{bmatrix} \qquad (2.3.44)$$

Thus from equation (2.3.39)

$$\begin{bmatrix} R \\ 0 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad (2.3.45)$$

where the m_{ij} are the elements of the matrix $M_{N,1}$. Expanding (2.3.45) gives

$$m_{11} = R$$

$$m_{21} = 0 \qquad (2.3.46)$$

Since nothing is known about R except $R \neq 0$, the energy eigenvalue condition becomes

$$m_{21} = 0 \qquad (2.3.47)$$

Chapter 3

The Choice of the Model Equation

Although the form of the model equation has been specified in (2.1.3), exactly what equation it is depends on the choice of the function g(z). With this choice, the solutions Ψ_1, Ψ_2 of the model equation will be determined as well as the transformation equation (2.1.18). Even though simplicity in the solutions Ψ_1, Ψ_2 is desirable, this constraint is not as important as how the choice of g(z) will affect the solution z(x) of the transformation equation.

3.1 The Case of g(z)=0

The simplest choice is g(z)=0. Now the model equation becomes

$$\Psi''(z) = 0 \tag{3.1.1}$$

which has solutions

$$\Psi_1 = 1 \tag{3.1.2}$$
$$\Psi_2 = z$$

which certainly qualify as simple solutions. The two independent solutions become

$$y_1 = u(x) \tag{3.1.3}$$
$$y_2 = u(x) \int_{x_0}^{x} \frac{dx}{u^2}$$

since

$$z(x) = \int_{x_0}^{x} \frac{dx}{u^2} \tag{3.1.4}$$

When the associated transformation equation is obtained from

(1.2.20) it becomes

$$W(x) - u''/u = 0 \qquad (3.1.5)$$

or

$$u'' = W(x)u \qquad (3.1.6)$$

A comparison of this equation with (2.1.2) shows that it is identical to the Schroedinger equation. Since the aim of this method was the simplification of the solution of the Schroedinger equation by its transformation clearly nothing has been achieved.

3.2 Derivation of Approximation Methods

A non-zero $g(z)$ means that the transformation equation (2.1.18)

$$W(x) + \tfrac{1}{2}(z,x) - g(z)(z')^2 = 0$$

contains three terms and is no longer identical to the Schroedinger equation.

To gain a better understanding of the effect of the choice of $g(z)$ the transformation equation is examined when x is far from the turning point. Nothing can be said about the $W(x)$ term since no initial assumptions about the potential were made. Nothing can yet be said about the $g(z)z'^2$ term since nothing is yet known about $g(z)$. However, the $\tfrac{1}{2}(z,x)$ term does give some information.

A restriction on the transformation function $z(x)$ that $z'(x) \neq 0$ was introduced in the derivation of (2.1.7). As a result within the region where the transformation is valid, $z(x)$ must be a monotonicly increasing or decreasing function of x. This implies that $z(x)$ is a 'univalent' function[7]. The Schwarzian[7] (z,x) goes as $|x|^{-4}$ for large $|x|$. Thus it will be small compared

with the other terms and a good approximation for the transformation equation will become

$$W(x) - g(z)z'^2 = 0 \qquad (3.2.7)$$

for x far from the turning point.

For the choice g(z)=1, the model equation is

$$\psi''(z) = \psi(z) \qquad (3.2.8)$$

which has independent solutions

$$\psi_1(z) = e^z$$
$$\psi_2(z) = e^{-z} \qquad (3.2.9)$$

From (3.2.7) the transformation equation for x far from the turning point is

$$W(x) = z'(x)^2 \qquad (3.2.10)$$

Solving this equation yields

$$z(x) = \int_{x_0}^{x} W(x)^{1/2}\, dx \qquad (3.2.11)$$

where x_0 is the turning point.The other transformation function u(x) is

$$u(x) = z'(x)^{-1/2} = W(x)^{-1/4} \qquad (3.2.12)$$

Thus the general solution y(x) for x far from the turning point is

$$y(x) = a_1 W(x)^{-1/4} \exp\left[\int_{x_0}^{x} W(x)^{1/2}\, dx\right] + a_2 W(x)^{-1/4} \exp\left[-\int_{x_0}^{x} W(x)^{1/2}\, dx\right] \qquad (3.2.13)$$

This is the well known JWKB approximate solution[8] to the Schroedinger equation in the forbidden region. Similar analysis with g(z)=-1 yields the sine and cosine solutions of the JWKB method in the classical region. It is clear that these solutions

y(x) have a singularity at the turning point since there W(x) has the value 0.

For the choice g(z)=z, the model equation becomes

$$\psi''(z) = z\psi(z) \qquad (3.2.14)$$

which has the independent solutions

$$\psi_1(z) = Ai(z)$$
$$\psi_2(z) = Bi(z) \qquad (3.2.15)$$

where Ai,Bi are the linearly independent Airy functions. The transformation equation for x far from the turning point from (3.2.7) is

$$\tilde{W}(x) = z(x)z'(x)^2 \qquad (3.2.16)$$

so

$$z'(x) = \left[\frac{W(x)}{z(x)} \right] \frac{1}{2} \qquad (3.2.17)$$

Solving for z(x) yields

$$z(x) = \left[\frac{3}{2} \int_{x_0}^{x} W(x)^{1/2}\, dx \right] \frac{2}{3} \qquad (3.2.18)$$

where x_0 is a turning point. Note that $z(x_0)=0$ so that the quantity in brackets in (3.2.17) is always positive. It's value at x_0 will be evaluated later. Thus the other transformation function u(x) is

$$u(x) = z'(x)^{-1/2} = W(x)^{-1/4} \left[\frac{3}{2} \int_{x_0}^{x} W(x)^{1/2}\, dx \right] \frac{1}{6} \qquad (3.2.19)$$

The general solution for x far from the turning point is thus

$$y(x) = W(x)^{-1/4} \left[\frac{3}{2} \int_{x_0}^{x} W(x)^{1/2}\, dx \right] \frac{1}{6} \left[a_1 Ai(z) + a_2 Bi(z) \right] \qquad (3.2.20)$$

This is the solution obtained by the uniform[9] JWKB method of solution to the Schroedinger equation, a modification of the Langer method[10]. The Airy functions are regular so singularities can only occur if $u(x)\to\infty$ so $z'=0$. Since $z'(x)=(W(x)/z(x))^{1/2}$ then $z'(x)$ can only be zero at $x=x_0$ where $W(x_0)=0$. However, $z(x_0)=0$ so L'Hospital's rule must be used to evaluate $z'(x_0)$ or equivalently $[z'(x_0)]^2$

$$\lim_{x\to x_0} [z'(x)]^2 = \lim_{x\to x_0} \frac{W(x)}{z(x)} = \lim_{x\to x_0} \frac{W'(x)}{z'(x)} = W'(x_0)/z'(x_0) \qquad (3.2.21)$$

so

$$z'(x_0) = [W'(x_0)]^{1/3} \qquad (3.2.22)$$

Thus no singularities exist in the uniform JWKB solution if $W(x)$ has a leading linear term.

For the general case $g(z)=z^p$ for integer $p\geq 0$ the model equation is

$$\psi''(z) = z^p\psi(z) \qquad (3.2.23)$$

which has the independent solutions

$$\psi_1(z) = z^{1/2}J_\alpha(\xi)$$
$$\psi_2(z) = z^{1/2}J_{-\alpha}(\xi) \qquad (3.2.24)$$

where J's are the Bessel functions of the first kind, $\xi=2\alpha z^{1/2\alpha}$, and $\alpha=(p+2)^{-1}$. The transformation for x far from the turning point is

$$W(x) = z^p(x)z'(x)^2 \qquad (3.2.25)$$

which has solutions

$$z(x) = \left[\frac{p+2}{2}\int_{x_0}^{x} W(x)^{1/2}\,dx\right]^{\frac{2}{p+2}} \qquad (3.2.26)$$

so

$$u(x) = W(x)^{-\frac{1}{4}} \left[\frac{1}{2\alpha} \int_{x_0}^{x} W(x)^{\frac{1}{2}}\, dx \right]^{\frac{p\alpha}{2}} \qquad (3.2.27)$$

and the general solution for x far from the turning point is

$$y(x) = W(x)^{-\frac{1}{4}} \left[\frac{1}{2\alpha} \int_{x_0}^{x} W(x)^{\frac{1}{2}}\, dx \right]^{\frac{p\alpha+1}{2}} (a_1 J_{\alpha}(\xi) + a_2 J_{-\alpha}(\xi)) \qquad (3.2.28)$$

From analysis similar to the uniform JWKB case, this approximate solution for $p>0$ will be regular everywhere, even at $x=x_0$, if $W(x)$ has a lowest term of p^{th} order in its Taylor expansion about x_0.

3.3 The Model Equation Used in This Work

For this work $g(z)=z$ was chosen so the model equation is the Airy equation with the Airy functions as solutions. The choice $g(z)=\pm 1$ also seems suitable but no calculation involving these functions have yet been done. Such calculations are planned for the future.

Chapter 4

Methods of Solving the Transformation Equation

Almost all of the effort that it takes to solve the Schroedinger equation by this method goes into the solution of the transformation equation. The model equation is, in effect, already solved since g(z) is chosen to make Ψ_1, Ψ_2 well known functions. Once these functions have been expanded, no additional work need be done on other problems which share the same model equation.

The transformation equation (2.1.18) is a third order non-linear differential equation. It may seem that the Schroedinger equation, a second order differential equation would be easier to solve than a third order equation but this need not be so in practice. The solution of the Schroedinger equation is often highly oscillatory, a situation that can cause significant errors to occur in numerical methods of solution.

In this method, the oscillatory part of the wave function is largely contained in the known functions which are solutions to the model equation. The transformation equation solution z(x) is a simple monotonicly increasing or decreasing function, a situation usually well handled by numerical methods.

Three methods have been used to solve the transformation equation; the iteration method, the power series method, and the coupled differential equation method. They will be individually examined in the next three sections. The transformation equation used in these sections will have g(z)=z as discussed in section 3.3 but all the methods can be modified for use with any g(z).

In addition, all methods assume $z'(x) > 0$ so $z(x)$ is monotonicly increasing. This causes no problem since if $z'(x) < 0$ the equations are solved by reversing the orientation through the use of regional coordinates. The relation between the global and regional coordinates is

$$x_{global} = A\, x_{regional} + B \qquad (4.0.1)$$

so

$$\frac{d}{dx}_{global} = A \frac{d}{dx}_{regional} \qquad (4.0.2)$$

where $A=\pm 1$ is the orientation and B is some offset value. Often B is chosen so that $x_{regional}$ is zero at a turning point. After the equations are solved using regional coordinates, the transformation functions can be easily converted back to global coordinates.

4.1 The Iteration Method[11]

Using the terminology in the above reference, the transformation equation can be written

$$W(x) = U(x) + \Delta[U,x] \qquad (4.1.3)$$

where

$$U(x) = z(x)z'(x)^2 \qquad (4.1.4)$$

$$\Delta[U,x] = -\mathcal{S}(z,x) \qquad (4.1.5)$$

The term $U(x)$ is called the substitute potential. This name comes from the fact that $U(x)$ replaces $W(x)$ in the modified uniform JWKB approximation discussed in section 3.2 and makes the solution exact instead of an approximation.

The iteration starts with the substitute potential $U(x)$ equal to the potential $W(x)$ as in the uniform JWKB approximation

$$U_0(x) = W(x) \qquad (4.1.6)$$

and from (4.1.3)

$$\Delta[U_0,x] = 0 \qquad (4.1.7)$$

where the subscript refers to the iteration step number. The 'potential defect' P(x) is defined as

$$P_0(x) = W(x) - (U_0(x) + \Delta[U_0,x]) \qquad (4.1.8)$$

For the first step, the substitute potential becomes

$$U_1(x) = U_0(x) + \lambda P_0(x) \qquad (4.1.9)$$

where λ is an arbitrary constant chosen to enhance the convergence. Typically λ=.01 .

The iteration continues with the recursion relations for the n^{th} step

$$\begin{aligned} P_n(x) &= W(x) - (U_n(x) + \Delta[U_n,x]) \\ U_{n+1}(x) &= U_n(x) + \lambda P_n(x) \end{aligned} \qquad (4.1.10)$$

After m such steps

$$W(x) = U_m(x) + \Delta[U_m,x] + P_m(x) \qquad (4.1.11)$$

The iteration terminates when $P_m(x) < \epsilon$ where ϵ is the desired accuracy of the solutions. At this point the transformation function z(x) can be calculated knowing

$$z'(x) = U_m(x)/z(x) \qquad (4.1.12)$$

so

$$z(x) = \left[\frac{3}{2} \int_{x_0}^{x} U_m^{1/2} \, dx \right]^{\frac{2}{3}} \qquad (4.1.13)$$

where x_0 is the turning point. The solution in the region is

$$y(x) = z'^{-1/2} (a_1 Ai(z) + a_2 Bi(z)) \qquad (4.1.14)$$

This method was successfully applied to the simple harmonic oscillator potiential and the N-fold barrier potential (see section 5.4). Wave functions and eigenvalues were obtained.

The main problem with the iteration method of solution of the transformation equation is speed. The search for the energy level was so slow that even after a large amount of computer time (30 minutes on an IBM 3031 mainframe) the accuracy rarely exceeded 1%. This prompted the development of the faster methods of solution discussed in the next two sections

4.2 Power Series Method

An often used method for solving differential equations involves the expansion of the solution in terms of a power series around some point $x=x_s$. After writing out the terms of the Schwarzian, the transformation equation becomes

$$W(x)z'(x)^2 + (1/2)z'(x)z'''(x) - (3/4)z''(x)^2 - z(x)z'(x)^4 = 0 \qquad (4.2.15)$$

The expansions for the functions $W(x)$ and $z(x)$ around $x=x_s$ are

$$W(x) = \sum_{\alpha=0}^{\infty} w_\alpha (x-x_s)^\alpha \qquad (4.2.16)$$

$$z(x) = \sum_{\alpha=0}^{\infty} z_\alpha (x-x_s)^\alpha \qquad (4.2.17)$$

so

$$z'(x) = \sum_{\alpha=1}^{\infty} \alpha z_\alpha (x-x_s)^{\alpha-1} \qquad (4.2.18)$$

$$z''(x) = \sum_{\alpha=2}^{\infty} \alpha(\alpha-1) z_\alpha (x-x_s)^{\alpha-2} \qquad (4.2.19)$$

$$z'''(x) = \sum_{\alpha=3}^{\infty} \alpha(\alpha-1)(\alpha-2) z_\alpha (x-x_s)^{\alpha-3} \qquad (4.2.20)$$

In terms of these coefficients, the starting conditions at the

point. $x=x_s$ become $z(x_s)=z_0$, $z'(x_s)=z_1$, and $z''(x_s)=2z_2$ .The transformation equation with the above expansions substituted thus becomes

$$\sum_{\substack{\alpha=1\\ \beta=1\\ \gamma=0}}^{\infty} \alpha z_\alpha^\beta z_\beta w_\gamma (x-x_s)^{\alpha+\beta+\gamma-2} + \tfrac{1}{2} \sum_{\substack{\alpha=3\\ \beta=1}}^{\infty} \alpha(\alpha-1)(\alpha-2) z_\alpha^\beta z_\beta (x-x_s)^{\alpha+\beta-4}$$

$$-(3/4) \sum_{\substack{\alpha=2\\ \beta=2}}^{\infty} \alpha(\alpha-1) z_\alpha^{\beta(\beta-1)} z_\beta (x-x_s)^{\alpha+\beta-4}$$

$$- \sum_{\substack{\alpha,\beta,\gamma,\delta=1\\ \varepsilon=0}}^{\infty} \alpha z_\alpha^\beta z_\beta^\gamma z_\gamma^\delta z_\delta z_\varepsilon (x-x_s)^{\alpha+\beta+\gamma+\delta+\varepsilon-4} = 0 \qquad (4.2.21)$$

Since the $(x-x_s)^n$ term must vanish

$$\sum_{\alpha,\beta=1}^{n+1} \alpha\beta z_\alpha z_\beta w_{n-\alpha-\beta+2} + \tfrac{1}{2} \sum_{\alpha=3}^{n+4} \alpha(\alpha-1)(\alpha-2)(n-\alpha+4) z_\alpha z_{n-\alpha+4}$$

$$-(3/4) \sum_{\alpha=2}^{n+4} \alpha(\alpha-1)(n-\alpha+4)(n-\alpha+3) z_\alpha z_{n-\alpha+4}$$

$$- \sum_{\alpha,\beta,\gamma,\delta=1}^{n+1} \alpha\beta\gamma\delta z_\alpha z_\beta z_\gamma z_\delta z_{n-\alpha-\beta-\gamma-\delta+4} = 0 \qquad (4.2.22)$$

and combining the 2^{nd} and 3^{rd} sums

$$\sum_{\alpha,\beta=1}^{n+1} \alpha\beta z_\alpha z_\beta w_{n-\alpha-\beta+2} + \tfrac{1}{2} \sum_{\alpha=2}^{n+4} \alpha(\alpha-1)(n-\alpha+4)(5\alpha-3n-13) z_\alpha z_{n-\alpha+4}$$

$$- \sum_{\alpha,\beta,\gamma,\delta=1}^{n+1} \alpha\beta\gamma\delta z_\alpha z_\beta z_\gamma z_\delta z_{n-\alpha-\beta-\gamma-\delta+4} = 0 \qquad (4.2.23)$$

To obtain a general term involving n for use in a recursion formula, examine the upper limit of each sum. This general term must be non-vanishing.

For the first sum choose $\alpha=1$, $\beta=n+1$ (or vice versa) so the general term would be $(n+1)z_1 z_{n+1} w_0$. If $x_s=x_0$ then $w_0=W(x_0)=0$

and the term vanishes.

For the third sum choose $\alpha=\beta=\gamma=1$, $\delta=n+1$ so the general term would be $(n+1)z_1^3 z_{n+1} z_0$. It could happen that $z_0=z(x_s)=0$ so the general term would vanish.

For the second sum the choice $\alpha=n+4$ yields zero but $\alpha=n+3$ yields $\frac{1}{6}(n+3)(n+2)(n+1)z_{n+3}z_1$. Since $z_1=z'(x_s)\neq 0$ in the transformation range then this term will not vanish. Using this term as the general term

$$\begin{aligned}\tfrac{1}{6}(n+3)(n+2)(n+1)z_{n+3}z_1 &= -\sum_{\alpha,\beta=1}^{n+1}\alpha\beta z_\alpha z_\beta w_{n-\alpha-\beta+2} \\ &\quad -\tfrac{1}{4}\sum_{\alpha=2}^{n+2}\alpha(\alpha-1)(n-\alpha+4)(5\alpha-3n-13)z_\alpha z_{n-\alpha+4} \\ &\quad +\sum_{\alpha,\beta,\gamma,\delta=1}^{n+1}\alpha\beta\gamma\delta z_\alpha z_\beta z_\gamma z_\delta z_{n-\alpha-\beta-\gamma-\delta+4} \qquad (4.2.24)\end{aligned}$$

Using z_0, z_1, z_2 from the starting conditions all further z_n's can be recursively obtained from this equation. For example $n=0$ gives z_3.

$$\tfrac{1}{6}(3)(2)(1)z_3 z_1 = -z_1^2 w_0 + 3z_2^2 + z_0 z_1^4 \qquad (4.2.25)$$

Simplifying gives

$$z_3 = (3z_2^2 - z_1^2 w_0 + z_0 z_1^4)/(3z_1) \qquad (4.2.26)$$

Historically, this method was the successor to the iteration method. It proved to be very fast (about 1 second per integration) but the following problems occurred.

Using the non-linear simple potential (see section 5.1), an expansion around the turning point was obtained for the transformation function $z(x)$. A comparison of the calculated

to the analytic solution showed that the results were highly accurate at $x_s=x_0$ but away from this point the accuracy rapidly declined. By the time $|x-x_0|=1$, the results were completely wrong. Since this small radius of convergence was not well understood at the time, this method was superceded by the coupled differential equation method discussed in the next section.

The success of the coupled differential equation method caused the power series method to be nearly forgotten. At this time, it is not clear whether the problems encountered were related to the choice of $x_s=x_0$ the turning point or the choice of the potential or some inherent limitation in this method. Since the power series method is potentially more than ten times faster than the coupled differential equation method, a second look at it is planned for the future.

4.3 Coupled Differential Equation Method

The two equations involved in the transformation (2.1.12) and (2.1.20) can be written in the form of a coupled system of 3 first order ordinary differential equations. Each equation has the form

$$\frac{d}{dx} S_i(x) = f_i(x,S_1,S_2,\ldots,S_n) \qquad i=1,n \qquad (4.3.27)$$

where the S_i's are the functions to be obtained. In this case, the functions $S_1(x),S_2(x),S_3(x)$ are chosen to be $z(x),u(x),u'(x)$ and the coupled set of equations are

$$\frac{d}{dx} z(x) = u(x)^{-2} \qquad (4.3.28)$$

$$\frac{d}{dx} u(x) = u'(x) \qquad (4.3.29)$$

$$\frac{d}{dx} u'(x) = u(x)W(x) - z(x)u(x)^{-3} \qquad (4.3.30)$$

Although equation (4.3.29) looks artificial, it is needed to express the relationship between $S_2'(x)$ and $S_3(x)$.

These equations are solved by the subprogram DDFSYS[12]. The values of $z(x)$, $u(x)$, $u'(x)$ at the point x_n and the distance h to the next point are required as inputs to calculate $z(x_n+h)$, $u(x_n+h)$, and $u'(x_n+h)$.

At first some difficulties occasionally occurred when transformation functions grew very large, changed sign, and rapidly lost accuracy. This behavior can be easily corrected by starting a new region as discussed in section 4.4. Some other ideas for dealing with this are discussed in section 4.5.

This method has proven to be very successful in the solution of the transformation equation. The full program to solve the Eckart potential is presented in Appendix B. Often the accuracy of the eigenvalues and/or wavefunctions is 8 significant figures. Since the Airy function expansions used are accurate to about 8 significant figures, it is clear in these cases that the accuracy of the transformatin function $z(x)$ solved by this method must be at least that.

The search for a typical energy level and its associated wave function to an accuracy of about 7 significant figures takes about 1 minute on an IBM 3031 mainframe computer. To calculate the value of the eigenvalue condition m_{21} for a particular energy

takes about 4 seconds.

4.4 Starting Conditions for the Transformation Equation

The starting condition for the solution of the transformation equation are not related to the eigenvalue conditions discussed in section 2.3. The transformation of the Schroedinger equation to the model equation is apparently free from constraints. As a result, the starting conditions of the transformation equation are completely arbitrary, and thus can be chosen at will.

In practice, however, the choice of the starting conditions seems to affect the size of the region over which the transformation is valid. Thus a "poor" choice of the starting conditions can seriously hamper the numerical solution of the transformation equation. Although the question of choosing the "best" starting conditions is currently unsolved, the remedies presented here and in section 4.5 effectively eliminate the problem.

Because problems can occur if the choice of the starting conditions is completely random, it makes sense to choose conditions which are closely related to the solution. For x far from the turning point x_0, the approximate solutions discussed in section 3.2 give a good estimate for the solution. Using the uniform JWKB approximation, the starting conditions at $x=x_s$ become

$$z(x_s) = \left[\frac{3}{2} \int_{x_0}^{x_s} W(x)^{\frac{1}{2}} dx \right]^{\frac{2}{3}} \qquad (4.4.31)$$

$$z'(x_s) = \left[\frac{W(x_s)}{z(x_s)} \right]^{\frac{1}{2}} \qquad (4.4.32)$$

$$z''(x_s) = \frac{1}{2} z'(x_s) \left[\frac{W'(x_s)}{W(x_s)} - \frac{z'(x_s)}{z(x_s)} \right] \qquad (4.4.33)$$

These starting conditions can be easily calculated everywhere except at $x=x_0$.

At the turning point $x=x_0$, a closely related set of conditions are used.

$$W(x_0) = U(x_0) \qquad (4.4.34)$$

$$W'(x_0) = U'(x_0) \qquad (4.4.35)$$

$$W''(x_0) \neq U''(x_0) \qquad (4.4.36)$$

where $U(x)=z(x)z'(x)^2$ is the substitute potential discussed in section 4.1. The derivatives of $U(x)$ are

$$U'(x) = z'(x)^3 + 2z(x)z'(x)z''(x) \qquad (4.4.37)$$

$$U''(x) = 3z'(x)^2z''(x) + 2z'(x)^2z''(x) + 2z(x)z''(x)^2 + 2z(x)z'(x)z'''(x) \qquad (4.4.38)$$

At the turning point $W(x_0)=0$ and since $z'(x)\neq 0$ for any x then from (4.4.34) $z(x_0)=0$. Substituting into (4.4.37) and (4.4.38) and using (4.4.35), (4.4.36) yields

$$W'(x_0) = z'(x_0)^3 \qquad (4.4.39)$$

$$W''(x_0) = 5z'(x_0)^2z''(x_0) \qquad (4.4.40)$$

Thus the starting conditions for $x=x_0$

$$z(x_0) = 0 \qquad (4.4.41)$$

$$z'(x_0) = W'(x_0)^{1/3} \qquad (4.4.42)$$

$$z''(x_0) = (1/5)W''(x_0)W'(x_0)^{-2/3} \qquad (4.4.43)$$

It is clear that other starting conditions based on other approximations of the Schroedinger equation could have been used.

4.5 Other Conditions

Although the relationship between the starting conditions and the the range of validity of the transformation equation is only qualitatively understood the problem of where to start a new region has effectively been solved in the following way.

The derivative of the transformation function $z'(x)$ and the term of u''/u in the transformation equation are closely monitored. As the range of validiy of a region is ending, it seems that $z'(x) \to 0$. As this happens the quantity u''/u increases enormously in magnitude, tending to ∞. A new transformation region begins when $z'(x)$ changes signs. At this point u''/u abruptly changes signs and its magnitude decreases to a reasonable value.

A logical condition to start a new region is thus a large change in the quantity u''/u. The program in Appendix B starts a new region if u''/u changes by a factor of 2 over one integration step. Normally u''/u changes slowly so new regions are not frequently invoked.

When the importance of the term $u''/u = -\frac{1}{2}(z,x)$ was understood, it seemed that the maximum range of a region might be obtained if the integral of $(u''/u)^2$ could be minimized. It is clear a minimum exists since $u''/u \neq 0$ for all x (unless $W(x)$ is chosen as in section 5.1) and $u''/u = -\frac{1}{2}(z,x)$ tends to zero far from the turning point. Using the calculus of variations[13], the equations for the integral to be a minimum were derived (see Appendix A). This introduced additional unknown functions from the Lagrange multipliers which could not be solved unless the transformation

functions were already known. As a result this idea was not pursued further.

Recently several other attempts have been made to better understand the effect of the starting conditions on the range of the transformation equation. One idea involves stability analysis of non-linear systems[14],[15]. As of yet no results have been obtained since most of the available work deals with 2^{nd} order differential equations and the generalization to 3^{rd} order was not clear. The other idea involves the understanding of the transformation of the Schroedinger equation as a conformal mapping[16]. So far this idea seems to confirm that multiple regions are needed to cover the real axis.

Chapter 5

Examples

Some examples involving a variety of potentials follow. Most of these potentials were chosen because their exact solutions are available for comparison with results from this method.

5.1 Potentials with $\chi(z,x)=0$

The simplest transformation to solve with this method has the term $\chi(z,x)=0$ for all x. Potentials which fit this condition can be solved for directly using analytical methods. The transformation equation becomes

$$W(x) = z(x)z'(x)^2 \qquad (5.1.1)$$

and since $u''/u=-\chi(z,x)=0$

$$u''(x) = 0 \qquad (5.1.2)$$

This very simple diferential equation can be solved yielding

$$u(x) = C_1x + C_2 \qquad (5.1.3)$$

for arbitrary constants C_1, C_2.

If $C_1=0$ then $u=C_2$ so $z'=u^{-2}=C_2^{-2}=A$ and z can be written as

$$z(x) = A(x-x_0) \qquad (5.1.4)$$

for some constants A,x_0. Thus the corresponding potential W(x) can be calculated as

$$W(x) = z(x)z'(x)^2 = A^3(x-x_0) \qquad (5.1.5)$$

the exactly linear potential. It thus has the general solution

$$y(x) = a_1Ai(A(x-x_0)) + a_2Bi(A(x-x_0)) \qquad (5.1.6)$$

If $C_1\neq0$ then the potential is more interesting. Now

$$z'(x) = A(x-x_p)^{-2} \qquad (5.1.7)$$

and

$$z(x) = A\left[\frac{1}{(x-x_p)} - \frac{1}{(x_o-x_p)}\right] \qquad (5.1.8)$$

where A, x_o, x_p are arbitrary constants. Thus the potential $W(x)$ becomes

$$W(x) = \frac{A^3}{(x-x_p)^4}\left[\frac{1}{(x-x_p)} - \frac{1}{(x_o-x_p)}\right] \qquad (5.1.9)$$

An examination of the constants shows that $x=x_o$ is the classical turning point since $W(x_o)=0$. In addition, a pole in the potential occurs at $x=x_p$.

For physically interesting potentials, $W(x)$ will have $x_p \leq x_o$ and preferably $x_p \leq 0$. The potential $W(x)$ now has a minimum for $A<0$ (or a maximum for $A>0$) at x_m given by

$$x_m - x_p = (5/4)(x_o-x_p) \qquad (5.1.10)$$

At this point the value of the potential is

$$W(x_m) = 4^4 A^3 [5(x_o-x_p)]^{-5} \qquad (5.1.11)$$

The general solution is

$$y(x) = (x-x_p)(a_1 Ai(z) + a_2 Bi(z)) \qquad (5.1.12)$$

where z is given by equation (5.1.8). The specific solution depends on the boundary condtions.

To understand the problem just solved look more closely at the potential. In the limit as $x\to\infty$ the potential $W(x)\to 0$. Since $W(x)$ is energy adjusted, the solution (5.1.12) will be valid for the energy equal to the 'dissociation limit' of the potential. With the added requirement that the solution be an eigenstate, the boundary conditions applied to (5.1.12) become

$$\begin{aligned} y(x_p) &= 0 \\ y(\infty) &= 0 \end{aligned} \qquad (5.1.13)$$

Recall that in the limit as $z\to\infty$ the Airy function $Ai\to 0$ while

Bi→∞. Thus the condition $y(x_p)=0$ means that a_2 in equation (5.1.12) must be zero. Also since

$$\lim_{x\to\infty} z(x) = -\frac{A}{x_o-x_p} = \text{constant} \qquad (5.1.14)$$

then the only way to make $y(z(\infty))=0$ is to have $Ai(z(\infty))=0$. Thus $z(\infty)$ is a zero of the Airy function $Ai(z)$. If $z(\infty)$ is the n^{th} zero of the Airy function Ai and thus of the wave function then the chosen potential will support n-1 eigenstates of less energy.

In figure 1 is an example of this potential for the dissociation limit corresponding to the 5^{th} eigenenergy. The associated wave function and the transformation function z(x) are shown. The arbitrary constants are chosen as $A=-16$, $x_p=0$. The 5^{th} zero[17] of Ai(z) occurs at $z=-7.94413359$ so $x_o=A/z=2.014064821$. This type of potential where $(z,x)=0$ for all x can be obtained from any model equation of the form $\psi''(z)=z^n\psi(z)$. Then the potential becomes

$$W(x) = \frac{A^{n+2}}{(x-x_p)^4}\left[\frac{1}{(x-x_p)} - \frac{1}{(x_o-x_p)}\right]^n \qquad (5.1.15)$$

with an extremum point at x_m where

$$x_m - x_p = (4+n)(x_o-x_p)/4 \qquad (5.1.16)$$

and extremum value

$$W(x_m) = 4^4 A^{n+2} n^n ((4+n)(x_o-x_p))^{4+n} \qquad (5.1.17)$$

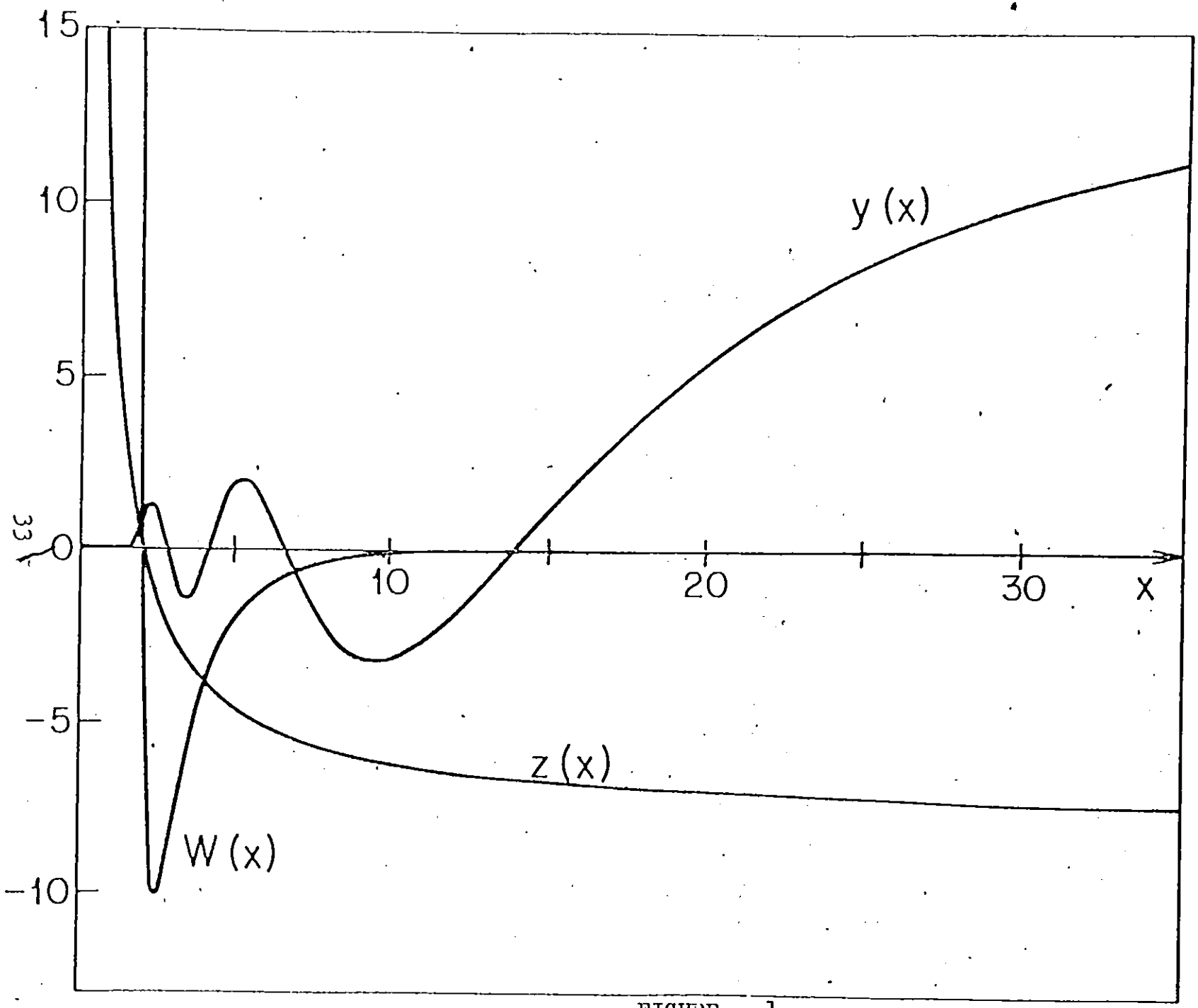


FIGURE 1

5.2 Harmonic Oscillator

The next logical choice for a test of the method is the harmonic oscillator potential $V(x)=ax^2$. It has energy levels[18]

$$E_n = (\hbar^2/2m)^{1/2}(2n+1)a^{1/2} \qquad (5.2.18)$$

with eigenfunctions

$$y_n(x) = \left[\frac{C}{\pi^{1/2}2^n n!} \right]^{1/2} e^{-\frac{1}{2}C^2x^2} H_n(Cx) \qquad (5.2.19)$$

where

$$C^2 = \left[\frac{2ma}{\hbar^2} \right]^{1/2} \qquad (5.2.20)$$

and H_n are the Hermite polynomials of degree n. For simplicity choose $2m/\hbar^2=1$ and $a=1$ so the energy adjusted potential is

$$W(x) = (2m/\hbar^2)(V(x) - E) = x^2 - E \qquad (5.2.21)$$

and the energy levels become

$$E_n = 2n+1 \qquad (5.2.22)$$

In this case, the potential is arbitrarily broken up into 2 regions with the cut occuring at the minimum point. Because of reflection symmetry around this point, only one of the regions need be solved for the transformation functions. The transformation functions in the other regions can be obtained by realizing that the effect of the reflection is to change the sign of the derivative so

$$\frac{d}{dx}_{right} = - \frac{d}{dx}_{left} \qquad (5.2.23)$$

where x_{right} and x_{left} are regional coordinates in the respective regions. Thus evaluated at the match point $\xi^{(right,left)}$

Energy Level	
Calculated	Exact
1.0000003123	1.0000000000

	Wave Function	
x	Calculated	Exact
5.0000	8.341367 E-03	8.344251 E-03
4.8000	1.490185 E-02	1.490315 E-02
4.6000	2.557382 E-02	2.557391 E-02
4.4000	4.216534 E-02	4.216425 E-02
4.2000	6.679377 E-02	6.679129 E-02
4.0000	1.016580 E-01	1.016537 E-01
3.8000	1.486527 E-01	1.486467 E-01
3.6000	2.088502 E-01	2.088409 E-01
3.4000	2.819184 E-01	2.819057 E-01
3.2000	3.656286 E-01	3.656120 E-01
3.0000	4.556014 E-01	4.555806 E-01
2.8000	5.454540 E-01	5.454290 E-01
2.6000	6.274215 E-01	6.273927 E-01
2.4000	6.934080 E-01	6.933762 E-01
2.2000	7.362860 E-01	7.362522 E-01
2.0000	7.511600 E-01	7.511255 E-01
1.8000	7.362860 E-01	7.362522 E-01
1.6000	6.934081 E-01	6.933762 E-01
1.4000	6.274215 E-01	6.273927 E-01
1.2000	5.454540 E-01	5.454290 E-01
1.0000	4.556015 E-01	4.555806 E-01
0.8000	3.656287 E-01	3.656120 E-01
0.6000	2.819185 E-01	2.819057 E-01
0.4000	2.088502 E-01	2.088409 E-01
0.2000	1.486527 E-01	1.486467 E-01
0.0000	1.016580 E-01	1.016537 E-01
-0.2000	6.679377 E-02	6.679129 E-02
-0.4000	4.216534 E-02	4.216425 E-02
-0.6000	2.557382 E-02	2.557391 E-02
-0.8000	1.490185 E-02	1.490315 E-02
-1.0000	8.341368 E-03	8.344251 E-03

Table 1:

Harmonic oscillator potential: Comparison of the exact and calculated energy level and wave function for the ground state n=0

Energy Level	
Calculated	Exact
3.0000045499	3.0000000000

	Wave Function	
x	Calculated	Exact
5.0000	1.771972 E-02	1.770082 E-02
4.8000	2.954491 E-02	2.950673 E-02
4.6000	4.708187 E-02	4.701706 E-02
4.4000	7.165601 E-02	7.155511 E-02
4.2000	1.040512 E-01	1.039028 E-01
4.0000	1.439665 E-01	1.437601 E-01
3.8000	1.894688 E-01	1.891964 E-01
3.6000	2.366172 E-01	2.362765 E-01
3.4000	2.794752 E-01	2.790724 E-01
3.2000	3.106801 E-01	3.102321 E-01
3.0000	3.226095 E-01	3.221441 E-01
2.8000	3.089871 E-01	3.085412 E-01
2.6000	2.665649 E-01	2.661802 E-01
2.4000	1.963999 E-01	1.961164 E-01
2.2000	1.042723 E-01	1.041217 E-01
2.0000	-2.731986 E-10	0.000000 E-01
1.8000	-1.042723 E-01	-1.041217 E-01
1.6000	-1.963999 E-01	-1.961164 E-01
1.4000	-2.665649 E-01	-2.661802 E-01
1.2000	-3.089871 E-01	-3.085412 E-01
1.0000	-3.226095 E-01	-3.221441 E-01
0.8000	-3.106801 E-01	-3.102321 E-01
0.6000	-2.794752 E-01	-2.790724 E-01
0.4000	-2.366171 E-01	-2.362765 E-01
0.2000	-1.894688 E-01	-1.891964 E-01
0.0000	-1.439665 E-01	-1.437601 E-01
-0.2000	-1.040512 E-01	-1.039028 E-01
-0.4000	-7.165601 E-02	-7.155511 E-02
-0.6000	-4.708187 E-02	-4.701706 E-02
-0.8000	-2.954491 E-02	-2.950673 E-02
-1.0000	-1.771972 E-02	-1.770082 E-02

Table 1:

Harmonic oscillator potential: Comparison of the exact and calculated energy level and wave function for n=1.

$$z_{right} = z_{left}$$
$$z'_{right} = -z'_{left} \qquad (5.2.24)$$
$$z''_{right} = z''_{left}$$

The results of the calculations for n=0 are shown in Table 1 and for n=1 are shown in Table 2.

5.3 Eckart Potential

The non-symmetric Eckart potential[19] is next chosen. It has the form

$$V(x) = \frac{A}{1+e^{-x}} + \frac{B}{(1+e^{-x})^2} \qquad (5.3.25)$$

The corresponding Schroedinger equation has the solution[20]

$$\Psi(x) = s^{-i\alpha}(1-s)^{-i\beta}F(a,b;c;s) \qquad (5.3.26)$$

where

$$\alpha/C = -i(-E_n)^{1/2} \qquad (5.3.27)$$
$$\beta/C = i(A - E_n)^{1/2} \qquad (5.3.28)$$

and F(a,b;c;s) is the hypergeometric function with the parameters a, b, and c given by

$$a/C = (-E_n)^{1/2} + (A - E_n)^{1/2} + \tfrac{1}{2}[1 - (1-4B)^{1/2}] \qquad (5.3.29)$$
$$b/C = (-E_n)^{1/2} + (A - E_n)^{1/2} + \tfrac{1}{2}[1 + (1-4B)^{1/2}] \qquad (5.3.30)$$
$$c/C = 1 + 2(A - E_n)^{1/2} \qquad (5.3.31)$$

with $C^2=2m/\hbar^2$. Setting C=1 for convenience, the values chosen for the calculations are A=-1.922 and B=-11.2 for direct comparison with previous work[21].

The energy levels E_n of the bound states are characterized by wave functions which vanish at infinity. This forces the 'a' parameter of F(a,b;c;s) to be a negative integer and yields the energy levels

$$E_n = -\tfrac{1}{2}T^{-1}(T-A)^2 \qquad (5.3.32)$$

with

$$T = \tfrac{1}{2}[(1-4B)^{\frac{1}{2}} - (2n+1)]^2 \qquad (5.3.33)$$

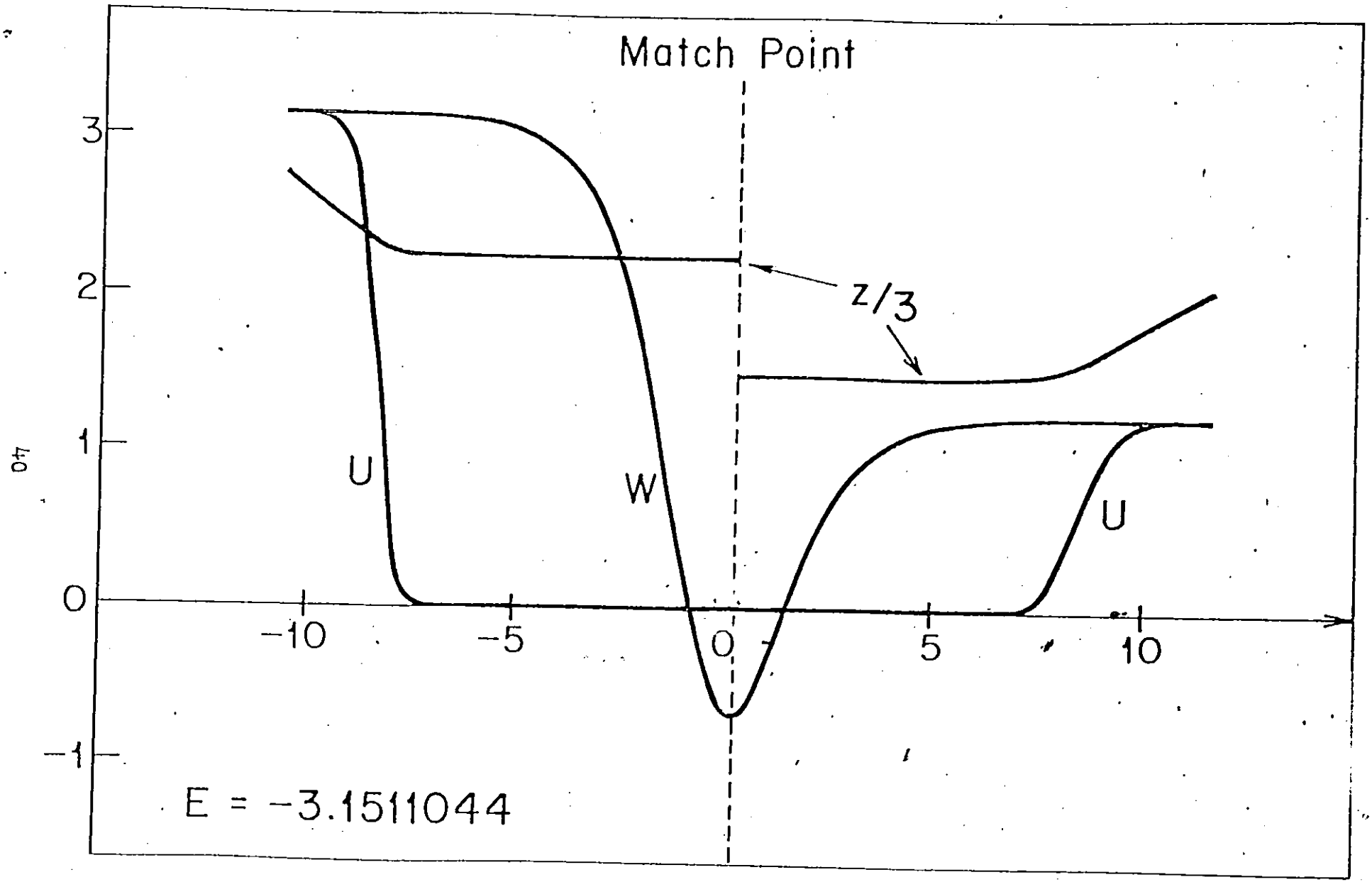
The results of calculations for n=0 and n=1 are shown in tables 3 and 4 respectively. Figures 2 and 3 show the potential W(x) and the transformation functions for n=0 and n=1 respectively. In both cases, the potential was divided into two regions by a cut near the minimum. No additional regions needed to be invoked as described in section 4.5.

Energy Level	
Calculated	Exact
-3.1511044226	-3.1511044226

	Wave Function	
x	Calculated	Exact
9.9268	7.566837 E-05	7.566830 E-05
9.3632	1.413181 E-04	1.413187 E-04
8.7997	2.639051 E-04	2.639067 E-04
8.2362	4.927618 E-04	4.927648 E-04
7.6726	9.198526 E-04	9.198582 E-04
7.1091	1.716364 E-03	1.716374 E-03
6.5456	3.200128 E-03	3.200147 E-03
5.9820	5.958545 E-03	5.958583 E-03
5.4185	1.106849 E-02	1.106856 E-02
4.8549	2.047591 E-02	2.047603 E-02
4.2914	3.760708 E-02	3.760731 E-02
3.7279	6.821328 E-02	6.821370 E-02
3.1643	1.210987 E-01	1.210995 E-01
2.6008	2.073145 E-01	2.073158 E-01
2.0373	3.342930 E-01	3.342951 E-01
1.4737	4.904924 E-01	4.904954 E-01
0.9102	6.261718 E-01	6.261757 E-01
0.3466	6.635370 E-01	6.635411 E-01
-0.2045	5.679188 E-01	5.679225 E-01
-0.7556	3.923959 E-01	3.923984 E-01
-1.3067	2.244269 E-01	2.244283 E-01
-1.8579	1.108331 E-01	1.108338 E-01
-2.4090	4.937743 E-02	4.937774 E-02
-2.9601	2.056849 E-02	2.056862 E-02
-3.5113	8.217446 E-03	8.217498 E-03
-4.0624	3.201369 E-03	3.201389 E-03
-4.6135	1.228748 E-03	1.228756 E-03
-5.1646	4.675227 E-04	4.675255 E-04
-5.7158	1.769869 E-04	1.769875 E-04
-6.2669	6.680529 E-05	6.680421 E-05
-6.8180	2.517637 E-05	2.517257 E-05

Table 3:

Eckart potential: Comparison of the exact and calculated energy level and wave function for the lower (ground) level.


Match Point
3
2
1
0
−1
−10
−5
0
5
10
z/3
U
W
U
E = −3.1511044

Energy Level	
Calculated	Exact
-2.1084074538	-2.1084074165

	Wave Function	
x	Calculated	Exact
11.1585	1.825204 E-02	1.824907 E-02
10.4827	2.443178 E-02	2.442962 E-02
9.8070	3.270276 E-02	3.270122 E-02
9.1313	4.376885 E-02	4.376780 E-02
8.4555	5.856516 E-02	5.856450 E-02
7.7798	7.832449 E-02	7.832419 E-02
7.1041	1.046474 E-01	1.046474 E-01
6.4283	1.395459 E-01	1.395462 E-01
5.7526	1.853736 E-01	1.853742 E-01
5.0768	2.444083 E-01	2.444093 E-01
4.4011	3.175089 E-01	3.175104 E-01
3.7253	4.005864 E-01	4.005882 E-01
3.0496	4.768317 E-01	4.768341 E-01
2.3739	5.042917 E-01	5.042942 E-01
1.6981	4.117384 E-01	4.117405 E-01
1.0224	1.474158 E-01	1.474166 E-01
0.8534	6.040970 E-02	6.041004 E-02
0.7312	-4.426915 E-03	-4.426918 E-03
0.3466	-2.000887 E-01	-2.000898 E-01
-0.0425	-3.472107 E-01	-3.472124 E-01
-0.4316	-4.117619 E-01	-4.117640 E-01
-0.8207	-3.962737 E-01	-3.962758 E-01
-1.2098	-3.297778 E-01	-3.297796 E-01
-1.5989	-2.465104 E-01	-2.465117 E-01
-1.9881	-1.701896 E-01	-1.701905 E-01
-2.3772	-1.108754 E-01	-1.108760 E-01
-2.7663	-6.928831 E-02	-6.928867 E-02
-3.1554	-4.204447 E-02	-4.204470 E-02
-3.5445	-2.499408 E-02	-2.499422 E-02
-3.9336	-1.464845 E-02	-1.464854 E-02
-4.3228	-8.501619 E-03	-8.501678 E-03
-4.7119	-4.901226 E-03	-4.901278 E-03
-5.1010	-2.812697 E-03	-2.812753 E-03
-5.4901	-1.609098 E-03	-1.609178 E-03
-5.8792	-9.185397 E-04	-9.186689 E-04

Table 4:

Eckart potential: Comparison of the exact and calculated energy level and wave function for the upper (excited) level.

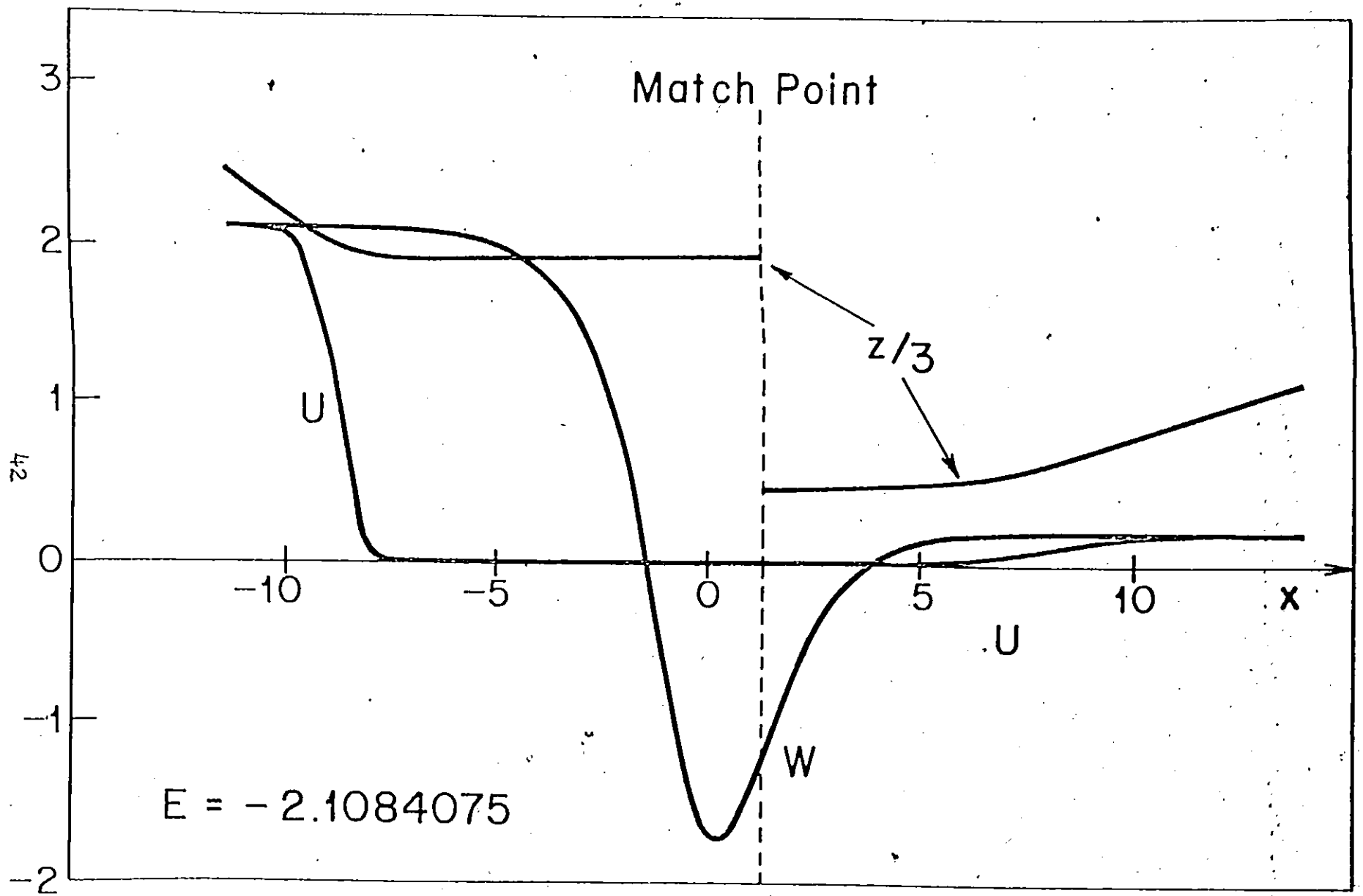


Fig 3

5.4 N-fold Periodic Ring Potential

In this case, the basic sector potential

$$V(x) = 1 - \cos x \qquad (5.3.34)$$

for $0 \leq x \leq 2\pi$ is repeated N times and then closes in on itself (see figure 4). As a result, the periodic boundary condition

$$\begin{aligned} y(x) &= y(x+2\pi N) \\ y'(x) &= y'(x+2\pi N) \end{aligned} \qquad (5.3.35)$$

must be used instead of that discussed in section 2.3.

The highly symmetric nature of the potential suggests that a wise selection of cuts could effectively reduce the number of regions to be solved. If a cut occurs at each maximum or mimimum point then 2N nearly identical regions will be formed. Neighbouring regions n,n+1 will have opposite orientation. Thus for the regional coordinates x_n, x_{n+1}

$$\frac{d}{dx_{n+1}} = - \frac{d}{dx_n} \qquad (5.4.36)$$

so at the match point $\xi^{(n+1,n)}$

$$\begin{aligned} z_{n+1} &= z_n \\ z'_{n+1} &= - z'_n \\ z''_{n+1} &= z''_n \end{aligned} \qquad (5.4.37)$$

The relation between the $\mathbb{B}$ matrices of neighbouring regions comes about since

$$\begin{aligned} \underline{b}_{n+1} &= \underline{b}_n \\ \underline{b}'_{n+1} &= - \underline{b}'_n \end{aligned} \qquad (5.4.38)$$

so

$$\mathbb{B}_{n+1}(\xi^{(n+1,n)}) = \mathbb{E}\, \mathbb{B}_n(\xi^{(n+1,n)}) \qquad (5.4.39)$$

where

Fig. 4.

$$E = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad (5.4.40)$$

and note that $E^{-1}=E$.

If regions of positive orientation are identified with '+' and regions of negative orientation with '-' then the B matrices at a given match point $x=\xi$ are

$$B_+ = E\,B_- \qquad B_+^{-1} = B_-^{-1}\,E$$
$$B_- = E\,B_+ \qquad B_-^{-1} = B_+^{-1}\,E \qquad (5.4.41)$$

The match matrix for the first sector is

$$M_{31} = M_{32}\,M_{21} \qquad (5.4.42)$$
$$= B_3^{-1}(\xi^{(32)})\;B_2(\xi^{(32)})\;B_2^{-1}(\xi^{(21)})\;B_1(\xi^{(21)})$$

or using (5.4.41)

$$M_{31} = B_+^{-1}(\xi^{(32)})\;E\;B_+(\xi^{(32)})\;B_+^{-1}(\xi^{(21)})\;E\;B_+(\xi^{(21)}) \qquad (5.4.43)$$

and the equation will remain true if B_+ is replaced by B_-.

The connection matrices of the other sectors will be equal to M_{31} for any sector potential V(x). Thus if M_{31} is defined as M the connection matrix around the ring (that is for N sectors) is M^N.

The boundary conditions (5.4.35) imply that after going around the ring, the wave function will repeat itself exactly. Thus

$$\underline{a} = M^N\,\underline{a} \qquad (5.4.44)$$

so

$$M^N = 1 \qquad (5.4.45)$$

As a result if λ is an eigenvalue of M then

$$\lambda^N = 1 \qquad (5.4.46)$$

so

$$\lambda_k = e^{2\pi ik/N} \qquad (5.4.47)$$

for $k=0,1,\ldots,N-1$ or alternately for integer k when $-N/2 \leq k \leq (N/2+1)$. All eigenvalues occur in complex conjugate pairs on the unit circle except for $\lambda_0=1$ and $\lambda_{N/2}=-1$ for even N which are real.

If the eigenvectors of M are a_k then

$$M\, a_k = \lambda_k\, a_k \qquad (5.4.48)$$

This gives the coefficient vector in the first sector. The value of the coefficient vector in a given sector S is obtained by applying M a total of S times so

$$a_{\text{sector } S} = M^S\, a_k \qquad (5.4.49)$$

A matrix theorem[22] states

$$\text{trace } M = \lambda_k + \lambda_{-k} \qquad (5.4.50)$$

so

$$\text{trace } M = 2\cos(2\pi k/N) \qquad (5.4.51)$$

is the condition for bound states. The quantity k^* is written for k and k is redefined as

$$k = 2\pi k^*/N$$

Since trace M is a function of energy then call P(E)=trace M so the bound state condition becomes

$$P(E) = 2\cos k \qquad (5.4.52)$$

Now it can be seen that all energy levels of the same ratio k^*/N are the same so that all levels are doubly degenerate except for $k=0$ and $k=\pm\pi$.

At this point using the bound state condition, energy levels and wave functions for specific N and k values can be solved for.

N	level	This work	exact[23],[24] Mathieu Method
1	A	0.6215109	0.62151
	A	1.9180610	1.91806
3	A	0.6215109	0.62151
	E	0.6441334	0.64413
	E	1.6492793	1.64928
	A	1.9180610	1.91806
4	A	0.6215109	0.62151
	E	0.6362859	0.63629
	A	0.6523310	0.65233
	A	1.5947995	1.59480
	E	1.7130943	1.71309
	A	1.9180610	1.91806

A non-degenerate level
E doubly degenerate level

Table 5:
Energy levels for a ring potential consisting of N repeating potential wells of the form $V(x) = 1 - \cos(x)$. Comparison between proposed method and exact method (based on Mathieu functions). For N = 2 take the A levels of the N = 4 case. While the Mathieu function method is restricted to sinusoidal potentials, non-symmetric and non-repetitive multiple potential wells can now be treated under the proposed method.

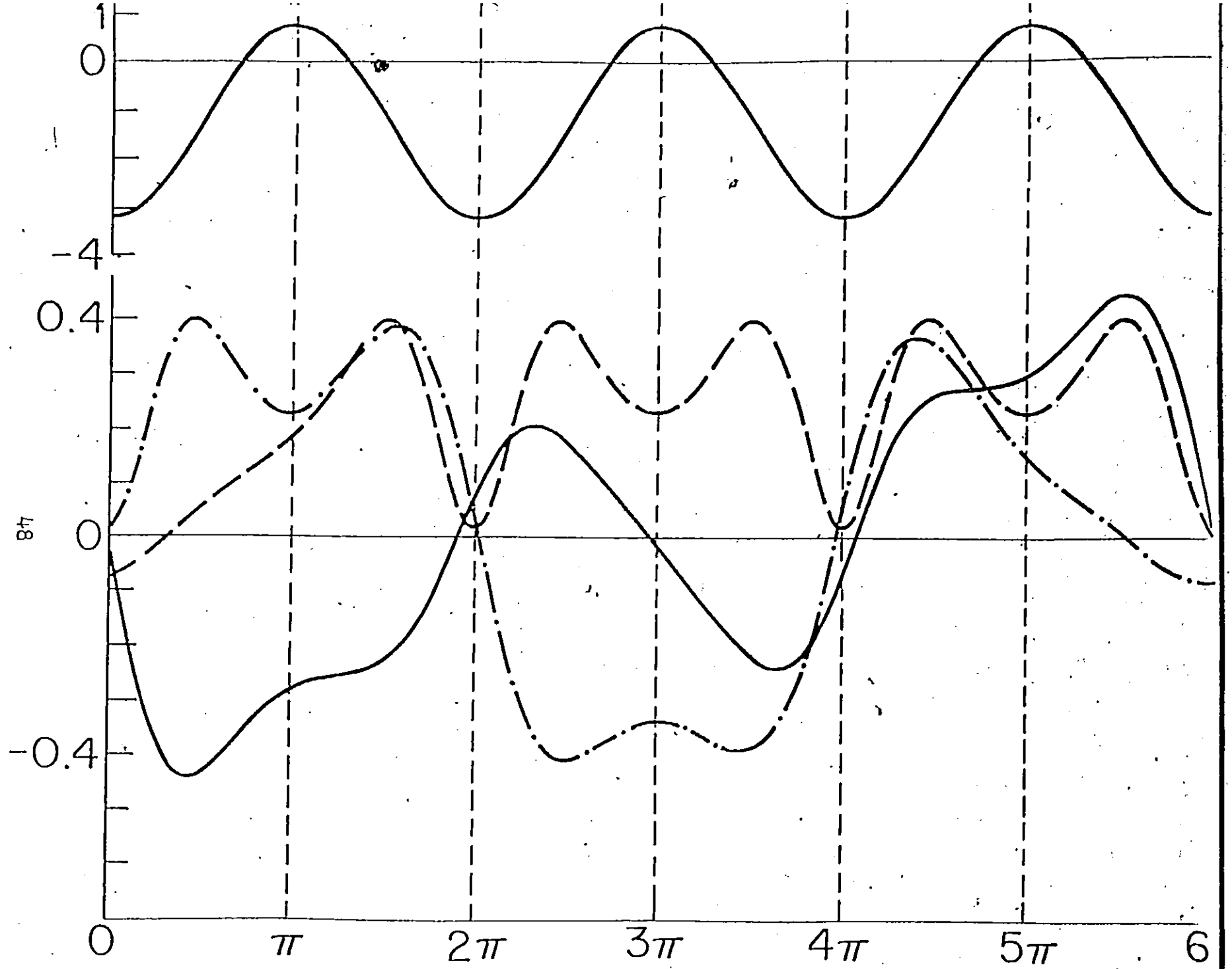


Fig. 5

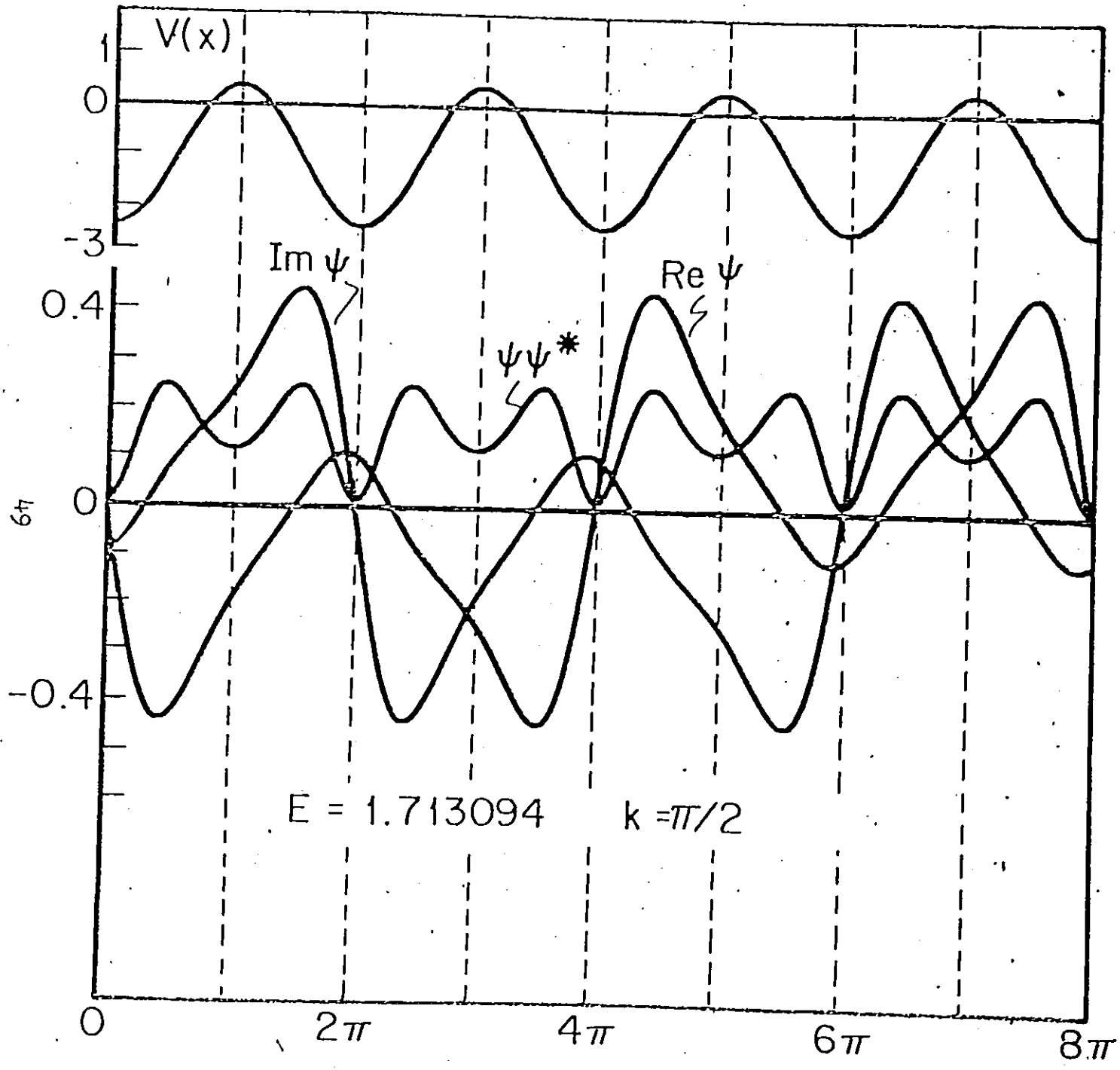


Fig. 6

Table 5 shows the results for N=1 to N=4 for all possible energy levels compared to exact results[23],[24] obtained by directly solving the Schroedinger equation. Figure 5 gives the wave function for $k=\pi/2$ corresponding to an energy of E=1.713094 and figure 6 gives the wave function for $k=2\pi/3$ corresponding to an energy E=1.6492793.

It is clear, however, that equation (5.4.52) contains more information. The solid state limit of $N\to\infty$ can be obtained by considering k a continuous variable. Thus a Brillouin zone plot of energy E as a function of k can be obtained here (see figure 7).

Energy bands will be formed whenever there are real valued solutions k to equation (5.4.52) (see figure 8). The order of the levels in each band alternate from band to band, so that k=0 is the lowest level in the lowest band, but the highest level in the next band and so on. Each band is made up of doubly degenerate levels, except for the levels at the band edge where $k=0,\pi$. Thus the case N=1 yields the number of bands supported by the potential and with the addition of the N=2 case establishes the band edges. Higher N's fill in the band with doubly degenerate levels (see figure 9).

By obtaining the function P(E) to a high accuracy in the region covered by the energy bands, the case of any integer N can be solved easily by high accuracy interpolation methods. The case of N=60, chosen for its large number of level subsets, is shown in table 6.

The bound state condition (5.4.52) is valid for any given sector potential repeated N times. Thus even non-symmetric sector

potentials can be solved in this way.

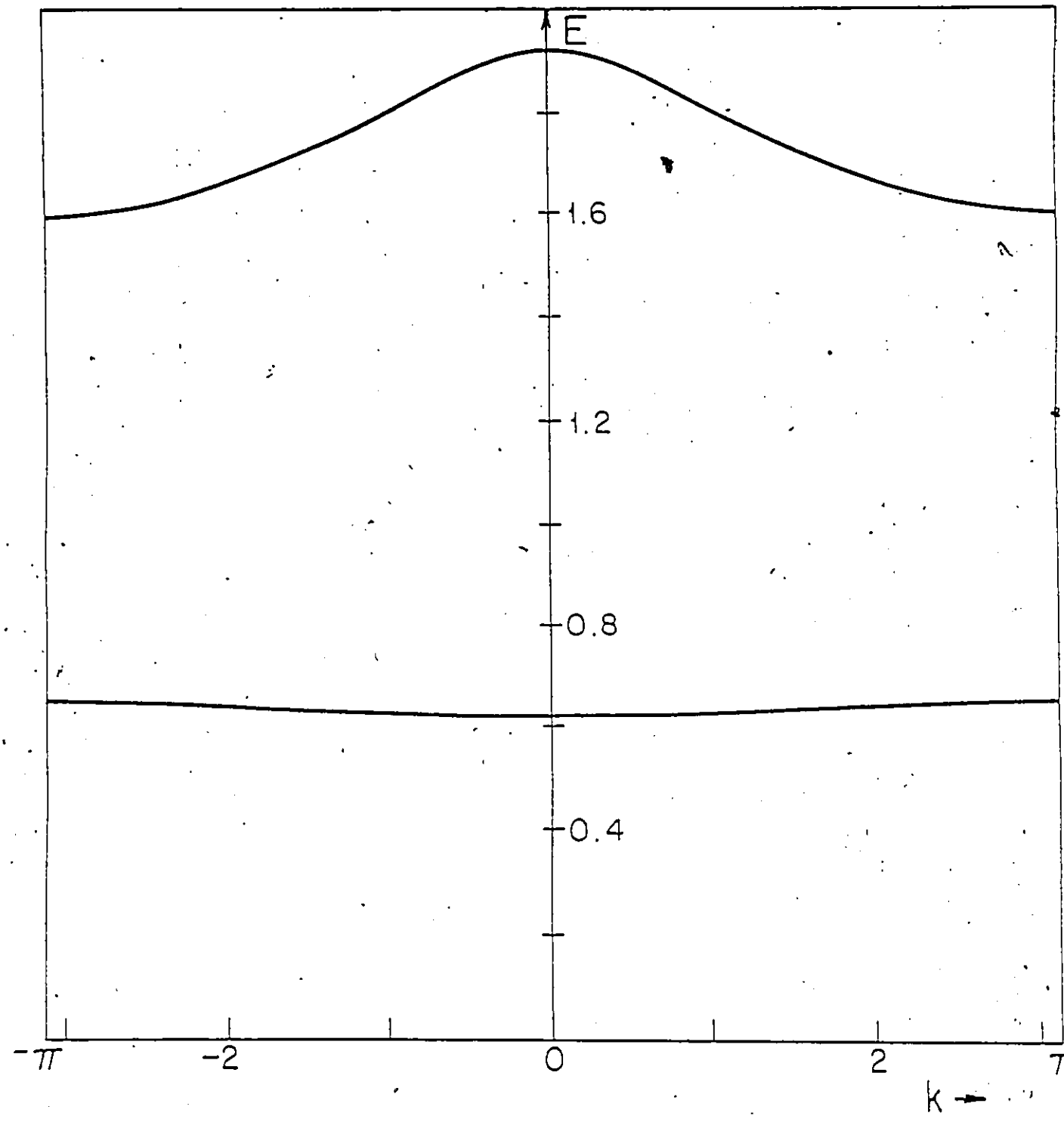


Fig. 7

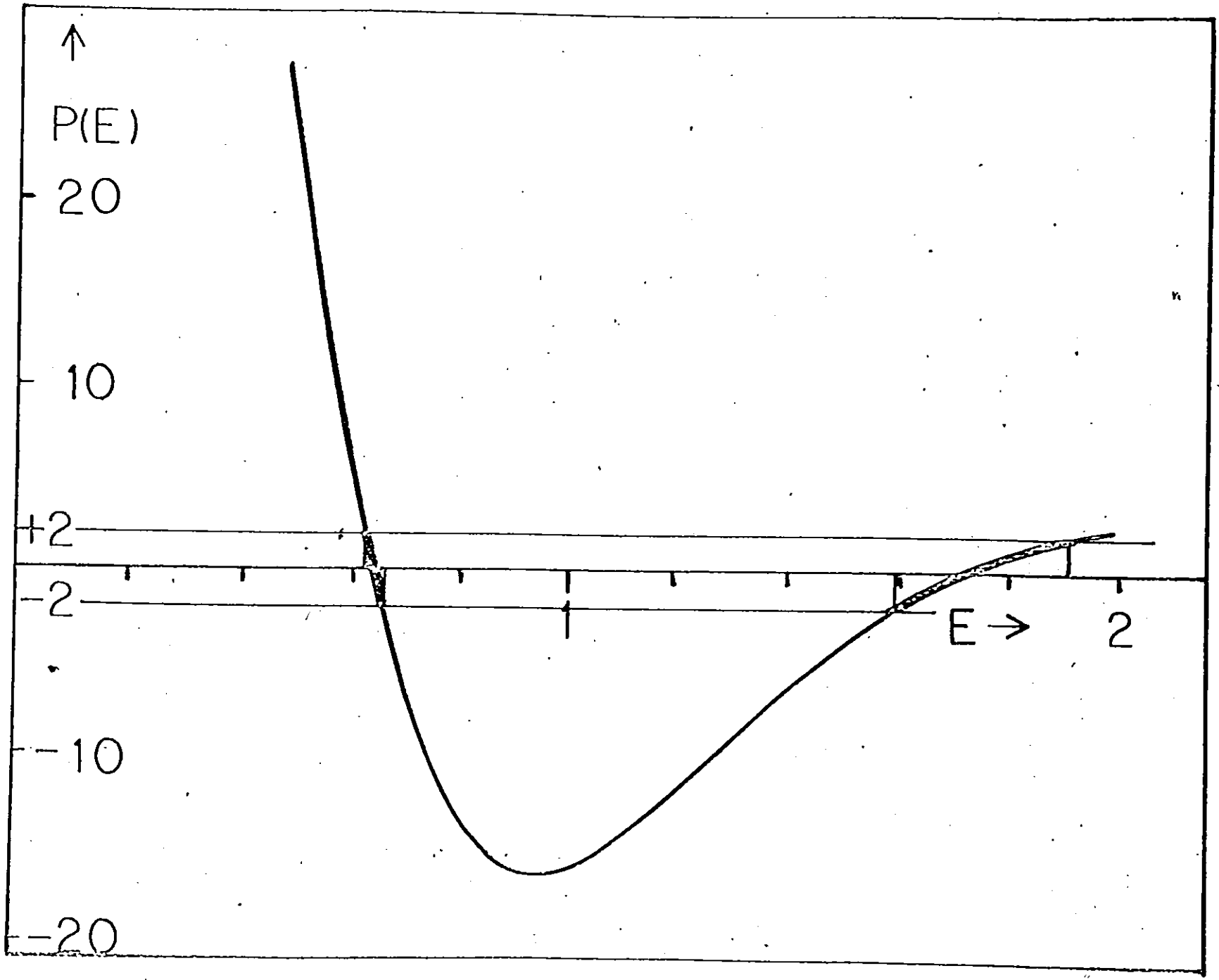


Fig. 8

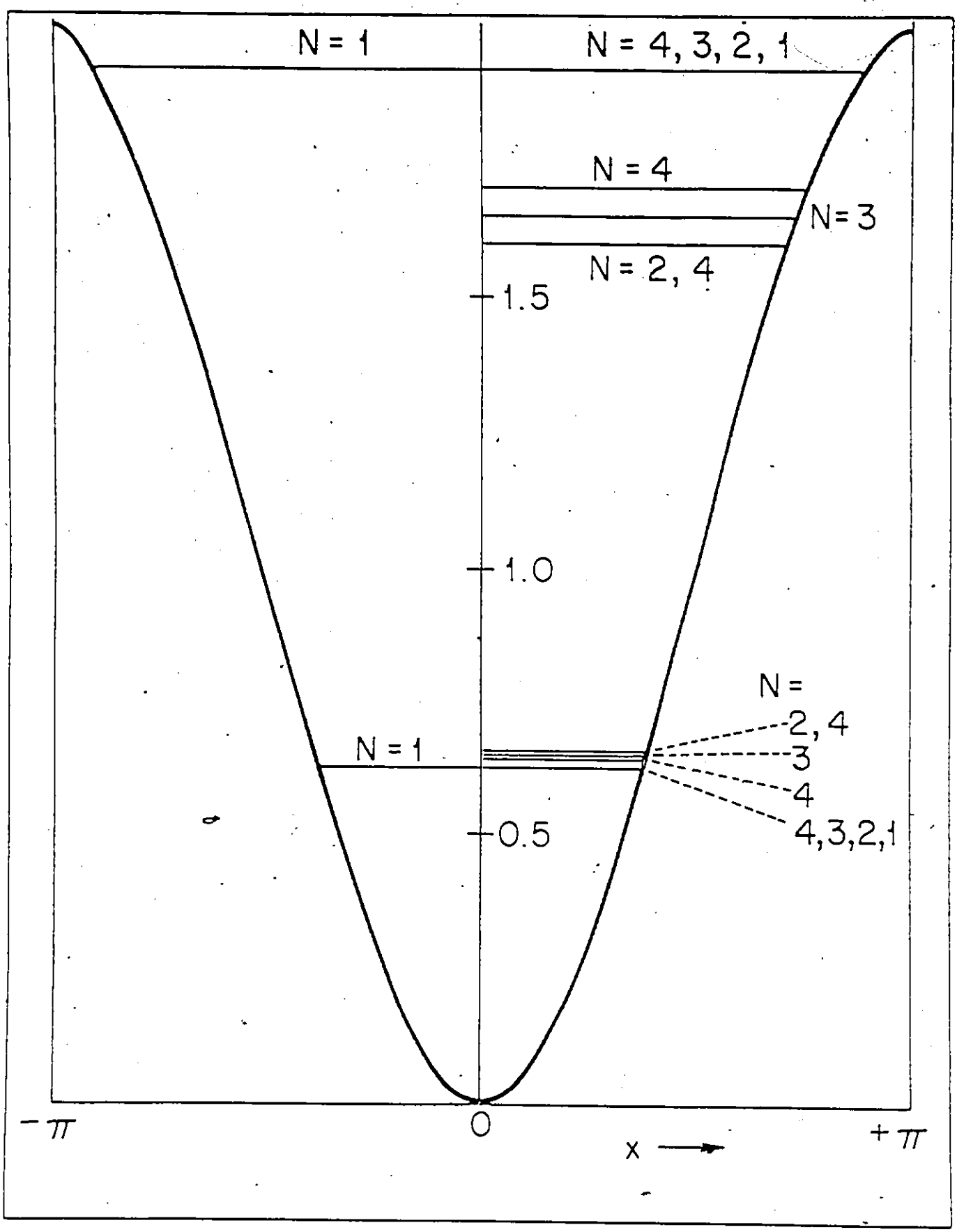


Fig. 9

$\pm k^*$	Lower Band	Upper Band
0	.62151081	1.9180582
1	.62158871	1.9161050
2	.62182174	1.9104128
3	.62220790	1.9014304
4	.62274387	1.8897618
5	.62342502	1.8760388
6	.62424537	1.8608387
7	.62519767	1.8446484
8	.62627337	1.8278616
9	.62746266	1.8107895
10	.62875450	1.7936772
11	.63013666	1.7767181
12	.63159578	1.7600663
13	.63311742	1.7438464
14	.63468617	1.7281606
15	.63628576	1.7130943
16	.63789916	1.6987199
17	.63950874	1.6850997
18	.64109648	1.6722884
19	.64264409	1.6603339
20	.64413334	1.6492792
21	.64554622	1.6391625
22	.64686524	1.6300182
23	.64807371	1.6218768
24	.64915602	1.6147654
25	.65009792	1.6087075
26	.65088680	1.6037232
27	.65151196	1.5998291
28	.65196480	1.5970383
29	.65223906	1.5953600
30	.6523309-1	1.5948000

Table 6:

Energy levels of an N-fold periodic sinusoidal ring potential $V(x) = 2m/\hbar^2\ (1 - \cos x)$ for $N = 60$ and $2m/\hbar^2 = 1$. All levels are doubly degenerate, except the levels for $k^* = 0$ and $k^* = 30$ which are the ones at the band limit. Certain values of $N < 60$ yield appropiate subsets of the above list. Starting with $k^* = 0$, increment k^* by $\Delta k^* = 60/N$ until $k^* = 30$ is surpassed. Thus, this table is valid for N = 1, 2, 3, 4, 5, 6, 10, 12, 15, 20 and 30.

Appendix A

Minimization of u"/u

The calculus of variations can be used to minimize[13] the square of the quantity $u''/u = W - zz'^2$ over the interval (x_1, x_2). Define the integral I as

$$I = \int_{x_1}^{x_2} (W(x) - z(x)z'(x)^2)^2 \, dx \qquad (A.1)$$

subject to the conditions put on by the solution of the transformation equation

$$u''/u - (W - zz'^2) = 0 \qquad (A.2)$$

$$u - (z')^{-1/2} = 0 \qquad (A.3)$$

The overall function to minimize $\Phi(x, z, z', u, u', u'')$ thus becomes

$$\Phi = (W - zz'^2)^2 + \lambda_1(u''/u - (W - zz'^2)) + \lambda_2(u - (z')^{-1/2}) \qquad (A.4)$$

for the Lagrange multipliers λ_1, λ_2.

The Euler equations to solve are

$$\Phi_z - (d/dx)\Phi_{z'} = 0 \qquad (A.5)$$

$$\Phi_u - (d/dx)\Phi_{u'} + (d/dx)^2\Phi_{u''} = 0 \qquad (A.6)$$

The needed partial derivatives of Φ are

$$\Phi_z = 2(W - zz'^2)(-z'^2) + \lambda_1(z'^2) \qquad (A.7)$$

$$\Phi_{z'} = 2(W - zz'^2)(-2zz') + \lambda_1(2zz') + \tfrac{1}{2}\lambda_2(z')^{-3/2} \qquad (A.8)$$

$$\Phi_u = -\lambda_1 u'' u^{-2} + \lambda_2 \qquad (A.9)$$

$$\Phi_{u'} = 0 \qquad (A.10)$$

$$\Phi_{u''} = \lambda_1 u^{-1} \qquad (A.11)$$

In this case, use the so called natural boundary conditions

$$\Phi_{z'}(x_{end}) = \Phi_{u''}(x_{end}) = 0 \qquad (A.12)$$

for $x_{end} = x_1$ or $x_{end} = x_2$. The Lagrange multiplier functions thus have the value

$$\lambda_1 = 0 \quad (\text{if } u \neq 0) \tag{A.13}$$

$$\lambda_2 = 8(W - zz'^2)z(z')^{-5/2} \tag{A.14}$$

at the endpoints x_1 and x_2.

Substituting (A.7) to (A.10) into (A.5) and (A.6) gives the differential equations

$$-2(W - zz'^2)z'^2 + \lambda_1 z'^2 \tag{A.15}$$

$$-(d/dx)[-4zz'(W - zz'^2) + 2\lambda_1 zz' + \tfrac{1}{2}\lambda_2(z')^{-3/2}] = 0$$

$$-\lambda_1 u'' u^{-2} + \lambda_2 + (d/dx)^2[\lambda_1 u^{-1}] = 0 \tag{A.16}$$

Although the λ's are in theory now specified, in practise their solution is extremely difficult. In order to solve (A.15) and (A.16) the transformation functions z, z', u, u', u'' must be known not only as functions of x but as functions of the starting conditions. This is clearly enormously more difficult than the current problem of determining the transformation functions for one given starting condition. When this was realized, the idea was not pursued further.

Appendix B

Computer Program using the Transformation Method

The following program solves for the energy level and the wave function of the ground state (n=0) of the Eckart potential using the coupled differential method discussed in section 4.3. The model equation in all regions is the Airy equation thus solutions are in terms of the Airy function Ai,Bi. A brief explanation of the program follows. Modifications needed for use with a different potential are also discussed.

The 'MAIN' program searches for a change in sign of the value of m_{21} as given by the function COND. When an interval where the sign changes is identified, the routine CONVER uses a modified Newton method to converge on the energy which makes $m_{21}=0$. This energy value is used by the program WAVE to calculate the final wave function.

The function COND consists of two sections. The first half of the routine determines the regions and then for each region calls the routine SUBPOT to solve the coupled differential equations to obtain the transformation functions. The last half of the routine uses the endpoints of the regions to form the appropriate match matrix, taking into account regional coordinates. These matrices are multiplied together to get the overall match matrix and hence the value of m_{21}.

Most of the remaining routines are self explanatory. The Airy function routine is based on an expansion of Chebyshev polynomials. Regular polynomial expansions can be found in references (1) and (2).

If the solution of a different potential is desired, certain

changes have to be made to the program. First of all, the potential and its derivatives must be available from V,VP,VPP. Major changes are needed in the COND routine to tell the program where the new arbitrarily defined regions are to be. Also changes often have to be made in MAIN to change the upper and lower limits of the search. Other minor changes may need to be done in TITLE, WAVE, and the routines used to calculate the exact value of the wave function if that is desired.

```
//ECKART  JOB  (XXXXXXXXXX),'ED BREWER'
//  EXEC  WATFIV
//GO.SYSIN  DD  *
$JOB   WATFIV  XXXXXXXXX BREWER
C      MAIN
       IMPLICIT REAL*8 (A-H,O-Z)
       EXTERNAL COND
       COMMON /POTENT/ ACOEFF,BCOEFF,EEXACT,NLEVEL
       REAL*8 ACOEFF/-1.922D0/,BCOEFF/-11.20D0/
       REAL*8 PI/3.14159265358979/,ACCUR/1.D-8/
C
       WMAX=ACOEFF-ACCUR
       WMIN=.25D0*(ACOEFF+BCOEFF)**2/BCOEFF+ACCUR
       NLEVEL=1
       CALL INIT
C
       WMESH=DABS(WMIN-WMAX)/20.D0
       WNEW=WMIN
       WOLD=0
       COLD=0
C
C      FIND AN AREA WHERE THE CONDITION CHANGES SIGN
C
       WHILE (DABS(WMAX-WOLD).GT.1.D-14) DO
       CNEW=COND(WNEW)
C       CALL WAVE(WNEW)
       PRINT,'W=',WNEW,'COND=',CNEW
C
C      CONVERGE TO THE ZERO POINT IN THE SIGN CHANGE REGION
C
       IF (COLD/CNEW.LT.0) THEN DO
             CALL CONVER(COND,CNEW,WNEW,COLD,WOLD,0.D0,ACCUR,CBEST,WBEST)
             CALL WAVE(WBEST)
             STOP
             END IF
       WOLD=WNEW
       COLD=CNEW
       WNEW=WNEW+DMIN1(DABS(WMAX-WNEW),WMESH)
       END WHILE
       STOP
       END
C
       FUNCTION COND(ENER)
C
C      CALCULATE THE VALUE OF M(2,1) SINCE THE EIGENVALUE CONDTION
C             IS M(2,1)=0
C
C
C      NOTE THAT THE ZEDVAL COMMON AREA CONTAINS INFORMATION
C             ABOUT ALL OF CALCULATED REGIONS:
C
C      ENERGY     --- ENERGY AT WHICH Z,U,UP VALUES ARE CALCULATED
C      MAP(N)     --- TYPE OF REGION N
C      TRANSF(N,M)-- TRANSFORMATION INFO FOR REGION N
```

```
C                    GLOBAL X=TRANSF(N,1)+ TRANSF(N,2)*(LOCAL X)
C     MATCH(N,I,J)  -- MATCH MATRIX FOR REGION N
C     MAXREG ------ MAXIMUM NUMBER OF REGIONS
C
C          EACH INTEGRATION USING SUBPOT WILL CALCULATE VALUES OF
C      Z,U,UP  WHICH DEPEND ONLY ON THE ENERGY ADJUSTED  POTENTIAL
C      W(X)  (WHERE  X IS A LOCAL X).  DIFFERENT  POTENTIALS  W(X)
C      DEFINE DIFFERENT "TYPES" OF REGIONS.OFTEN DIFFERENT REGIONS
C      WILL HAVE THE SAME W(X) SO, INSTEAD OF RECALCULATION, THOSE
C      REGIONS ARE ASSIGNED (IN THE VECTOR "MAP") THE SAME TYPE AS
C      THE FIRST.
C
C     ZED(I,J,K)  -- I= REGION TYPE
C                 --- J=VALUES OF X,Z,U,UP
C                          J=1 IS LOCAL X
C                          J=2 IS Z
C                          J=3 IS U
C                          J=4 IS UP
C                --- K=POINT #
C
C     NP(I)-------- NUMBER OF POINTS IN ZED(I,J,K) IN REGION TYPE I
C     ORDER(I)----- ORDERING OF POINTS IN ZED(I,J,K) IN REGION TYPE I
C                     ORDER(I)=+1 IF POINTS ORDERED SMALLEST TO LARGEST
C                     ORDER(I)=-1 IF POINTS ORDERED LARGEST TO SMALLEST
C     MAXTYP ------ MAXIMUM NUMBER OF REGION TYPES
C     MAXPT ------- MAXIMUM NUMBER OF POINTS IN A REGION
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /ERROR/ IERROR
      COMMON /POTENT/ ACOEFF,BCOEFF,EEXACT,NLEVEL
      COMMON /ZEDVAL/ENERGY,ZED,TRANSF,MATCH,
     X                MAP,NP,ORDER,REGION,TYPE,MAXREG,MAXTYP,MAXPT
      INTEGER REGION,TYPE,MAP(20),NP(20),ORDER(20)
      REAL*8 ZED(20,4,20),TRANSF(20,2),MATCH(20,2,2)
      INTEGER MAXTYP/20/,MAXREG/20/,MAXPT/20/
      REAL*8 B(2,2),BI(2,2),M(2,2)
      REAL*8 ONE(2,2)/1.D0,0.D0,0.D0,1.D0/
C
C     SET UP FOR ERROR RETURN
      IERROR=1
      COND=0
C
C     PREPARE FOR MULTIPLE REGIONS
C
      ENERGY=ENER
      TYPE=REGION=1
      DO 10 I=1,MAXTYP
10    NP(I)=0
C
      CALL ENDPT(ENERGY,XRIGHT,XLEFT,XMIN,IER)
      IF (IER.NE.0) RETURN
C
C     SET UP CUT
C
      XCUT=XMIN
```

```
C
C     CALCULATE Z,U,UP FOR RIGHT REGION
C
      XSTART=6.D0
      XEND=XCUT-XRIGHT
      CALL INTER(XSTART,XEND,XRIGHT,1.D0)
C
C     CALCULATE Z,U,UP FOR LEFT REGION
C
      XSTART=4.D0
      XEND=XLEFT-XCUT
      CALL INTER(XSTART,XEND,XLEFT,-1.D0)
C
C     CALCULATE THE OVERALL MATCH MATRIX AND ASSIGN REGION TYPES
C
      LASREG=REGION-1
C
C     CALCULATE MATRIX IN FIRST REGION
C
      REGION=1
      CALL ENDMAT(B,BI,+1,IER)
      IF (IER.NE.0) RETURN
      CALL MATMUL(M,B,ONE)
      REGION=REGION+1
C
C     MULTIPLY BY B*(B INVERSE) UNTIL LAST REGION
C
      WHILE (REGION.LT.LASREG) DO
      CALL ENDMAT(B,BI,+1,IER)
      IF (IER.NE.0) RETURN
      CALL MATMUL(M,BI,M)
      EXECUTE SMATCH
      CALL MATMUL(M,B ,M)
      REGION=REGION+1
      END WHILE
C
C     CALCULATE MATRIX FOR LAST REGION
C
      REGION=LASREG
      CALL ENDMAT(B,BI,+1,IER)
      IF (IER.NE.0) RETURN
      CALL MATMUL(M,BI,M)
      EXECUTE SMATCH
C
      IERROR=0
      PRINT,' '
      PRINT,'M MATRIX='
      CALL MATPRT(M)
      COND=M(2,1)
      RETURN
C
C     REMOTE BLOCKS
C
      REMOTE BLOCK SMATCH
      DO 11 IVAL=1,2
```

```
      DO 11 JVAL=1,2
11    MATCH(REGION-1,IVAL,JVAL)=M(IVAL,JVAL)
      END BLOCK
      END
C
      SUBROUTINE INTER(XFIRST,XLAST,SHIFT,SENSE)
C
C     ROUTINE TO CALCULATE Z,U,UP OVER SOME INTERVAL
C
C           NOTE: NEW REGIONS MAY BE INVOKED IF
C                 CONDITION DESCRIBED IN ULIMIT
C                 IS VIOLATED
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /ZEDVAL/ENERGY,ZED,TRANSF,MATCH,
     X               MAP,NP,ORDER,REGION,TYPE,MAXREG,MAXTYP,MAXPT
      INTEGER REGION,TYPE,MAP(20),NP(20),ORDER(20)
      REAL*8 ZED(20,4,20),TRANSF(20,2),MATCH(20,2,2)
      REAL*8 PAR(10)
      INTEGER NPOINT/10/
      EXTERNAL ZEROTH,ULIMIT
C
      STEP=DABS(XFIRST-XLAST)/NPOINT
      XSTART=XFIRST
      XEND=XLAST
      INTREG=REGION
C
      IER=-1
      WHILE (IER.NE.0) DO
      EXECUTE NEWREG
C
C     IF DELTA U CANNOT BE MADE EQUAL TO 0 AT A POINT THEN
C           START INTEGRATION AT 0 AND GO TO THAT POINT
C
      IF (IER.EQ.2) THEN DO
      MIDREG=REGION
      XSTART=0
      XEND=ZED(MAP(REGION-1),1,NP(MAP(REGION-1)))
C
C     MAKE SURE END POINT IS POSITIVE
C
      WHILE (XEND.LT.0) DO
      PRINT,'FORCED TO STEP BACK ONE POINT'
      NP(MAP(REGION-1))=NP(MAP(REGION-1))-1
      XEND=ZED(MAP(REGION-1),1,NP(MAP(REGION-1)))
      END WHILE
      PRINT,'STARTING AT ZERO'
C
C     INTEGRATE OUT TO XEND MAKING NEW REGIONS AS NECESSARY
C
      EXECUTE NEWREG
C
C     PUT REGIONS BACK IN CORRECT ORDER
C
      CALL SWITCH(MIDREG,LASREG)
```

```
C
C       CONTINUE OLD INTEGRATION FROM ZERO
C
        PRINT,'OLD INTEGRATION CONTINUES'
        XSTART=0
        XEND=XLAST
        END IF
C
        END WHILE
C
C       SWITCH ORDER OF REGIONS IF IN NEGATIVE SENSE AREA
C
        IF (SENSE.LT.0) CALL SWITCH(INTREG,LASREG)
        RETURN
C
        REMOTE BLOCK NEWREG
C
C       PROCEDURE TO CHOOSE ARGUMENTS FOR AND EXECUTE ROUTINE SUBPOT
C
C
        IER=-1
        WHILE (IER.NE.0.AND.IER.NE.2) DO
C
        TRANSF(REGION,1)=SHIFT
        TRANSF(REGION,2)=SENSE
C
        IF (IER.EQ.1) THEN DO
        MR=MAP(REGION-1)
        XSTART=ZED(MR,1,NP(MR))
        Z      =ZED(MR,2,NP(MR))
        U      =ZED(MR,3,NP(MR))
        UP     =ZED(MR,4,NP(MR))
        IOP=2
        ELSE DO
        IOP=1
        END IF
C
        CALL SUBPOT(XSTART,XEND,STEP,Z,U,UP,ZEROTH,ULIMIT,PAR,IOP,IER)
C
        IF (IER.NE.2) THEN DO
        MAP(REGION)=TYPE
        LASREG=REGION
        TYPE=TYPE+1
        REGION=REGION+1
        END IF
C
        END WHILE
        END BLOCK
        END
C
        SUBROUTINE SWITCH(IREG,LREG)
C
C       REVERSE THE ORDER OF REGIONS IREG THROUGH LREG
C
        IMPLICIT REAL*8(A-H,O-Z)
```

```
      COMMON /ZEDVAL/ENERGY,ZED,TRANSF,MATCH,
     X                MAP,NP,ORDER,REGION,TYPE,MAXREG,MAXTYP,MAXPT
      INTEGER REGION,TYPE,MAP(20),NP(20),ORDER(20)
      REAL*8 ZED(20,4,20),TRANSF(20,2),MATCH(20,2,2)
C
      IF (LREG.GT.IREG) THEN DO
      I=IREG
      L=LREG
      ELSE DO
      I=LREG
      L=IREG
      END IF
C
      WHILE (L.GT.I)
C
      ISAVE=MAP(I)
      MAP(I)=MAP(L)
      MAP(L)=ISAVE
C
      I=I+1
      L=L-1
      END WHILE
      RETURN
      END
C
      SUBROUTINE ENDMAT(B,BI,ISW,IER)
C
C     CALCULATE THE B AND B INVERSE MATRICES AT END POINTS
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 B(2,2),BI(2,2)
      COMMON /ZEDVAL/ENERGY,ZED,TRANSF,MATCH,
     X                MAP,NP,ORDER,REGION,TYPE,MAXREG,MAXTYP,MAXPT
      INTEGER REGION,TYPE,MAP(20),NP(20),ORDER(20)
      REAL*8 ZED(20,4,20),TRANSF(20,2),MATCH(20,2,2)
      INTEGER RIGHT,LEFT,PB,PBI
C
C     FIND RIGHT AND LEFT ENDPOINTS OF EACH REGION
C           USING TRANSF(REGION,2)*(POINT ORDERING OF REGION)
C
      MR=MAP(REGION)
      RIGHT=LEFT=1
      IF (TRANSF(REGION,2)*ORDER(MR).GT.0) THEN DO
      RIGHT=NP(MR)
      ELSE DO
      LEFT=NP(MR)
      END IF
C
C     IF ISW=+1 THEN CALCULATE B          AT LEFT  ENDPOINT
C                    AND  B INVERSE AT RIGHT ENDPOINT
C     IF ISW=-1 THEN CALCULATE B          AT RIGHT ENDPOINT
C                    AND  B INVERSE AT LEFT  ENDPOINT
C
      IF (ISW.EQ.1) THEN DO
      PB =LEFT
```

```
      PBI=RIGHT
      ELSE DO
      PB =RIGHT
      PBI=LEFT
      END IF
C
      CALL BMAT(B,ZED(MR,2,PB),ZED(MR,3,PB),ZED(MR,4,PB),
     X                          TRANSF(REGION,2),IER)
      IF (IER.NE.0) RETURN
      CALL BIMAT(BI,ZED(MR,2,PBI),ZED(MR,3,PBI),ZED(MR,4,PBI),
     X                          TRANSF(REGION,2),IER)
      RETURN
      END
C
C
      FUNCTION WAVEFN(X)
C
C     CALCULATE THE EXACT WAVEFUNCTION (HYPERGEOMETRIC FUNCTIONS)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /EXWAVE/ WNORM,ALPHA,BETA,AW,BW,CW,N
      Y=1.D0/(1.D0+DEXP(X))
      FACTOR=Y**BETA*(1.D0-Y)**ALPHA
      HYPER=TERM=1.D0
      IF (N.NE.0) THEN DO
      A=AW
      B=BW
      C=CW
      DO 1 I=1,N
      TERM=Y*A*B/C
      A=A+1.D0
      B=B+1.D0
      C=C+1.D0
1     HYPER=HYPER+TERM
      END IF
      WAVEFN=WNORM*FACTOR*HYPER
      RETURN
      END
C
      SUBROUTINE INIT
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /POTENT/ ACOEFF,BCOEFF,EEXACT,NLEVEL
      COMMON /EXWAVE/ WNORM,ALPHA,BETA,AW,BW,CW,N
C
      BETAFN(A,B)=DGAMMA(A)*DGAMMA(B)/DGAMMA(A+B)
C
      N=NLEVEL
      ELEVEL=2.D0*NLEVEL+1.D0
      T=.25D0*(DSQRT(1.D0-4.D0*BCOEFF)-ELEVEL)**2
      EEXACT=-.25D0*(T-ACOEFF)**2/T
C
C     CALCULATE VARIOUS CONSTANTS ASSOCIATED WITH
C           A PARTICULAR VALUE OF N
C
      ALPHA=DSQRT(-EEXACT)
```

```
      BETA=DSQRT(ACOEFF-EEXACT)
      AW=-N
      BW=ALPHA+BETA+.5D0*(1.D0+DSQRT(1.D0-4.D0*BCOEFF))
      CW=1.D0+2.D0*BETA
C
C     WNORM IS THE NORMALIZATION FACTOR
C
      WNORM=BETAFN(2.D0*BETA,2.D0*ALPHA)
      IF (N.EQ.1) THEN DO
      WNORM=WNORM-2.D0*BW*BETAFN(2.D0*BETA+1.D0,2.D0*ALPHA)/CW
      WNORM=WNORM+BW**2*BETAFN(2.D0*BETA+2.D0,2.D0*ALPHA)/CW**2
      END IF
      WNORM=1.D0/DSQRT(DABS(WNORM))
      RETURN
      END
C
C
      SUBROUTINE ENDPT(ENERGY,XRIGHT,XLEFT,XMIN,IER)
C
C     CALCULATE THE TURNING POINTS FOR A GIVEN VALUE OF ENERGY
C           ALSO XMIN=THE VALUE OF X WHERE THE MINIMUM OF
C           POTENTIAL OCCURS
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /POTENT/ ACOEFF,BCOEFF,EEXACT,NLEVEL
      AX=ACOEFF-ENERGY
      BX=2.D0*ENERGY-ACOEFF-BCOEFF
      CX=-ENERGY
      IER=1
      DISC=BX**2-4.D0*AX*CX
      IF (DISC.LT.0) RETURN
      ROOT=DSQRT(DISC)
      XE1=(-BX-ROOT)/2.D0/AX
      XE2=(-BX+ROOT)/2.D0/AX
      IF (XE1.GE.0.OR.XE2.GE.0) RETURN
      XRIGHT=DLOG(-XE1)
      XLEFT=DLOG(-XE2)
      XMIN=DLOG((BCOEFF+ACOEFF)/(BCOEFF-ACOEFF))
      IER=0
      RETURN
      END
C
      FUNCTION V(X)
C
C     CALCULATE THE VALUE OF THE POTENTIAL AT THE GLOBAL X
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /POTENT/ ACOEFF,BCOEFF,EEXACT,NLEVEL
      ZETA=-DEXP(X)
      FACTOR=ZETA/(1.D0-ZETA)
      V=-ACOEFF*FACTOR-BCOEFF*FACTOR/(1.D0-ZETA)
      RETURN
      END
C
      FUNCTION VP(X)
```

```
C
C       CALCULATE THE 1ST DERIV OF THE POTENTIAL AT THE GLOBAL X
C
        IMPLICIT REAL*8(A-H,O-Z)
        COMMON /POTENT/ ACOEFF,BCOEFF,EEXACT,NLEVEL
        ZETA=-DEXP(X)
        FACTOR=ZETA/(1.D0-ZETA)
        VP=(V(X)-BCOEFF*FACTOR**2)/(1.D0-ZETA)
        RETURN
        END
C
        FUNCTION VPP(X)
C
C       CALCULATE THE 2ND DERIV OF THE POTENTIAL AT THE GLOBAL X
C
        IMPLICIT REAL*8(A-H,O-Z)
        COMMON /POTENT/ ACOEFF,BCOEFF,EEXACT,NLEVEL
        ZETA=-DEXP(X)
        FACTOR=ZETA/(1.D0-ZETA)
        VPP=2.D0*(VP(X)-BCOEFF*FACTOR**3/ZETA**2)/(1.D0-ZETA)
        RETURN
        END
C
        SUBROUTINE WAVE(ECALC)
C
C       CALCULATE THE WAVE FUNCTION
C
        IMPLICIT REAL*8(A-H,O-Z)
        COMMON /POTENT/ ACOEFF,BCOEFF,EEXACT,NLEVEL
        COMMON /ZEDVAL/ENERGY,ZED,TRANSF,MATCH,
     X                  MAP,NP,ORDER,REGION,TYPE,MAXREG,MAXTYP,MAXPT
        INTEGER REGION,TYPE,MAP(20),NP(20),ORDER(20)
        REAL*8 ZED(20,4,20),TRANSF(20,2),MATCH(20,2,2)
        REAL*8 B(2,2),A(2)
C
        CALL TITLES(ECALC)
C
C
C       CHOOSE COEFFICIENT VALUES IN REGION 1 TO BE (1,0)
C
        A(1)=1.D0
        A(2)=0.D0
C
C       RUN OVER ALL REGIONS
C
        IMAX=REGION
        DO 1 IREG=1,IMAX
C
C       CALCULATE WAVE FUNCTION IN AT EACH POINT IN REGION
C
        NT=NP(MAP(IREG))
        MREG=MAP(IREG)
        DO 2 N=1,NT
        NN=N
        IF (TRANSF(IREG,2)*ORDER(MREG).GT.0) NN=NT-N+1
```

```
      CALL BMAT(B,ZED(MREG,2,NN),ZED(MREG,3,NN)
     X     ,ZED(MREG,4,NN),TRANSF(IREG,2),IER)
      WAVEF=A(1)*B(1,1)+A(2)*B(1,2)
      X=TRANSF(IREG,1)+ZED(MREG,1,NN)*TRANSF(IREG,2)
      CALL WAVOUT(X,WAVEF,ZED(MREG,2,NN),ZED(MREG,3,NN),ENERGY)
2     CONTINUE
C
      WRITE(6,200) A
200   FORMAT(' ',/,' A(1)=',D22.12,' A(2)=',D22.12,/)
C
C     GET COEFFICIENTS FOR NEXT REGION
C
      IF (IREG.NE.IMAX) THEN DO
      A(1)=MATCH(IREG,1,1)
      A(2)=MATCH(IREG,2,1)
      END IF
C
C     GO TO NEXT REGION
C
1     CONTINUE
C
      RETURN
      END
C
      SUBROUTINE TITLES(ECALC)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /POTENT/ ACOEFF,BCOEFF,EEXACT,NLEVEL
C
      WRITE(6,10) NLEVEL
10    FORMAT(' ',/' WAVEFUNCTIONS OF ECKART POTENTIAL FOR N=',I2)
C
C     PRINT TITLES
C
      WRITE(6,120) EEXACT
120   FORMAT(' AND FOR  EXACT ENERGY LEVEL =',F15.10)
      WRITE(6,100) ECALC
100   FORMAT(' WITH CALCULATED ENERGY LEVEL=',F15.10/)
      WRITE(6,101)
101   FORMAT(' ',7X,'X',7X,2X,'CALCULATED',3X,
     X  8X,'Z',8X,5X,'SMALL U',5X
     X ,8X,'U',8X,8X,'W',8X)
C
      RETURN
      END
C
      SUBROUTINE WAVOUT(X,WAVEF,Z,U,ENERGY)
C
C     PRINT OUT THE CALCULATED WAVE FUNCTIONS
C
      IMPLICIT REAL*8(A-H,O-Z)
      ZP=1.D0/U**2
      WRITE(6,1) X,WAVEF,Z,U,Z/U**4,V(X)-ENERGY
1     FORMAT(1X,F12.7,3X,1(1PD14.7,1X),4(D17.10))
      RETURN
      END
```

```
C
      SUBROUTINE BMAT(BM,Z,U,UP,XSIGN,IER)
C
C     CALCULATE THE B MATRIX GIVEN THE Z,U,UP VALUES
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BM(2,2)
C
C     XSIGN IS THE ORIENTATION OF THE LOCAL X RELATIVE TO
C           THE GLOBAL X (+1 FOR SAME,-1 FOR OPPOSITE)
C
      CALL AIRY(AI,BI,AIP,BIP,Z,IER)
      IF (IER.NE.0) RETURN
      BM(1,1)=U*AI
      BM(1,2)=U*BI
      BM(2,1)=XSIGN*(UP*AI+AIP/U)
      BM(2,2)=XSIGN*(UP*BI+BIP/U)
      RETURN
      END
C
      SUBROUTINE BIMAT(BIM,Z,U,UP,XSIGN,IER)
C
C     CALCULATE THE B INVERSE MATRIX GIVEN THE Z,U,UP VALUES
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BIM(2,2)
      REAL*8 PI/3.14159265358979/
C
C     XSIGN IS THE ORIENTATION OF THE LOCAL X RELATIVE TO
C           THE GLOBAL X (+1 FOR SAME,-1 FOR OPPOSITE)
C
      CALL AIRY(AI,BI,AIP,BIP,Z,IER)
      IF (IER.NE.0) RETURN
C
C     WRONSK= INVERSE OF THE WRONSKIAN OF THE AIRY FUNCTIONS (1/PI)
C
      WRONSK=XSIGN*PI
      BIM(1,1)= XSIGN*(UP*BI+BIP/U)*WRONSK
      BIM(1,2)=-(U*BI)*WRONSK
      BIM(2,1)=-XSIGN*(UP*AI+AIP/U)*WRONSK
      BIM(2,2)= (U*AI)*WRONSK
      RETURN
      END
C
      SUBROUTINE MATMUL(A,B,C)
C
C     DO A 2 BY 2 MATRIX MULTIPICATION A=B*C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(2,2),B(2,2),C(2,2),TEMP(2,2)
      DO 1 I=1,2
      DO 1 J=1,2
      TEMP(I,J)=0.D0
      DO 1 K=1,2
1     TEMP(I,J)=TEMP(I,J)+B(I,K)*C(K,J)
```

```
      DO 2 I=1,2
      DO 2 J=1,2
2     A(I,J)=TEMP(I,J)
      RETURN
      END
C
      SUBROUTINE MATPRT(A)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(2,2)
      WRITE(6,1) ((A(I,J),J=1,2),I=1,2)
1     FORMAT(2(' | ',D22.12,3X,D22.12,' |'/))
      RETURN
      END
C
      SUBROUTINE SUBPOT(XSTART,XEND,STEP,Z,U,UP,ICOND,ERCOND,PAR
     X                  ,IOP,IER)
C
C     CALCULATE THE Z VALUES FOR THE ENERGY NOW IN COMMON
C           STARTING INTEGRATION AT XSTART AND GOING TO XEND
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 EPS/1E-6/,STEP*8/10.D0/,HMAX*8
      DIMENSION Y(5),DY(5),PAR(10)
      EXTERNAL LANGER
C
C     FIND THE INITIAL CONDITIONS USING ROUTINE ICOND
C
      IF (IOP.NE.0) CALL ICOND(XSTART,XEND,Z,U,UP,PAR,IOP,IER)
      IF (IER.EQ.2) RETURN
C
C     USING ZP=1/U**2, SET UP INITIAL CONDITIONS FOR NEW
C           EQUATIONS WHERE U=Y(1),UP=Y(2),Z=Y(3)
C           AND Y(4)=INTEGRAL OF (DELTA U)**2
C
      Y(1)=U
      Y(2)=UP
      Y(3)=Z
      Y(4)=0.D0
      X=XSTART
C
      PRINT,' '
      PRINT,'AT UPPER POINT'
      PRINT,'Z=',Z
      PRINT,'U=',U,'UP=',UP
C
C     LOOP CONDITIONS
C
      H=(XEND-XSTART)/STEP
      HMAX=DABS(H)
      IER=-1
      CALL ERCOND(X,Y,PAR,IER)
C
C     N=4 IF INTEGRAL OF DELTA U SQUARED WANTED
C     N=3 IF NOT
C
```

```
      N=3
C
C     LOOP
C
      WHILE (DABS(XEND-X).GT.1.D-14.AND.IER.EQ.0) DO
      H=DSIGN(DMIN1(DABS(H),DABS(XEND-X),DABS(HMAX)),H)
C       PRINT,' '
C       PRINT,'X=',X
C       PRINT,'Z=',Y(3)
C       PRINT,'U=',Y(1),'UP=',Y(2)
      CALL SAVZED(X,Y,PAR)
      CALL DDFSYS(N,LANGER,EPS,H,X,Y,PAR)
      CALL ERCOND(X,Y,PAR,IER)
      END WHILE
C
C
C     THE VALUES OF Z,U,UP,ZP,ZPP AT THE END POINT ARE NOW KNOWN
C
      PRINT,' '
      IF (IER.NE.0) THEN DO
      PRINT,'CONDITION FORCED PREMATURE END OF REGION AT'
      ELSE DO
      PRINT,'AT LOWER POINT'
      CALL SAVZED(X,Y,PAR)
      END IF
      U=Y(1)
      UP=Y(2)
      Z=Y(3)
      PRINT,'Z=',Z
      PRINT,'U=',U,'UP=',UP
      RETURN
      END
C
      SUBROUTINE SAVZED(X,Y,PAR)
C
C     SAVE VALUES OF U,Z,ZP ETC. NEEDED LATER TO CALCULATE THE
C           WAVEFUNCTION
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /ZEDVAL/ENERGY,ZED,TRANSF,MATCH,
     X                MAP,NP,ORDER,REGION,TYPE,MAXREG,MAXTYP,MAXPT
      INTEGER REGION,TYPE,MAP(20),NP(20),ORDER(20)
      REAL*8 ZED(20,4,20),TRANSF(20,2),MATCH(20,2,2)
      DIMENSION Y(5),DY(5),PAR(10)
C
      IF (NP(TYPE).LT.MAXPT) NP(TYPE)=NP(TYPE)+1
      N=NP(TYPE)
      IF (N.EQ.2) THEN DO
            IF (X.GE.ZED(TYPE,1,1)) THEN DO
            ORDER(TYPE)=+1
            ELSE DO
            ORDER(TYPE)=-1
            END IF
      END IF
      ZED(TYPE,1,N)=X
```

```
      ZED(TYPE,2,N)=Y(3)
      ZED(TYPE,3,N)=Y(1)
      ZED(TYPE,4,N)=Y(2)
      RETURN
      END
C
C
      SUBROUTINE ULIMIT(X,Y,PAR,IER)
C
C     CHECK IF INTEGRATION SHOULD BE STOPPED AND
C           A NEW REGION STARTED
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(5),YSAVE(5),PAR(10)
      DATA RANGE/2.D0/
C
C     IF IER=-1 ON ENTRANCE THEN USE X,Y,PAR VALUES
C               TO SET LIMITS
C
      IER=0
      U=Y(1)
      IF (IER.EQ.-1) THEN DO
      UUPPER=U*RANGE
      ULOWER=U/RANGE
      ELSE DO
      IF (U.GT.UUPPER.OR.U.LT.ULOWER) IER=1
      END IF
      RETURN
      END
C
      SUBROUTINE ZEROTH(XSTART,XEND,Z,U,UP,PAR,IOP,IER)
C
C     GET THE INITIAL CONDITIONS AT XSTART USING THE ZEROTH ORDER
C          APPROXIMATION Z*ZP**2=W(X),U=1/Z**2 (DELTA U=0)
C
C
C     ON ENTRANCE IOP=1  -> INTEGRATE FROM 0 TO XSTART
C                           TO FIND VALUES OF Z,U,UP
C                 IOP=2  -> USE THE VALUE OF Z TO GET
C                           VALUES OF U,UP
C
C     NOTE: IF XSTART=0 THEN VALUES OF Z,U,UP ARE FROM
C           ABOVE EQUATION EVALUATED IN THE LIMIT AS
C           X APPROACHES ZERO.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 EPS/1E-5/
      DIMENSION Y(5),DY(5),PAR(10)
      EXTERNAL ZPRIME
C
      IER=0
C
      IF (XSTART.EQ.0) THEN DO
      Z=0
      ZP=WP(0.D0)**(1.D0/3.D0)
```

```
      ZPP=ZP*WPP(0.D0)/(3.D0*WP(0.D0))
      U=1.D0/ZP**2
      UP=-.5D0*ZPP*U**3
      RETURN
      END IF
C
      IF (IOP.EQ.1) THEN DO
C
C     FIRST FIND THE VALUE OF Z,ZPRIME AT XSTART
C
      X=.001D0*XSTART
      Y(1)=X*WP(0.D0)**(1.D0/3.D0)
      H=.1D0*XSTART
      N=1
C
C     INTEGRATE FROM 0 TO XSTART VIA ROUTINE DDFSYS
C
      WHILE (DABS(XSTART-X).GT.1.D-14) DO
      H=DSIGN(DMIN1(DABS(H),DABS(XSTART-X)),H)
      CALL DDFSYS(N,ZPRIME,EPS,H,X,Y,PAR)
      END WHILE
C
      Z=Y(1)
      END IF
C
      IF (IOP.EQ.2) THEN DO
      IF (W(XSTART)/Z.GT.0) THEN DO
      X=XSTART
      Y(1)=Z
      ELSE DO
      IER=2
      RETURN
      END IF
      END IF
C
      CALL ZPRIME(X,Y,DY,PAR)
      ZP=DY(1)
      ZPP=.5D0*ZP*(WP(X)/W(X)-ZP/Z)
      U=1.D0/DSQRT(ZP)
      UP=-.5D0*ZPP*U**3
C
      RETURN
      END
C
      FUNCTION W(X)
C
C     CALCULATE THE VALUE OF THE POTENTIAL AT A LOCAL COORD X
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /ZEDVAL/ENERGY,ZED,TRANSF,MATCH,
     X                MAP,NP,ORDER,REGION,TYPE,MAXREG,MAXTYP,MAXPT
      INTEGER REGION,TYPE,MAP(20),NP(20),ORDER(20)
      REAL*8 ZED(20,4,20),TRANSF(20,2),MATCH(20,2,2)
      W=V(TRANSF(REGION,1)+TRANSF(REGION,2)*X)-ENERGY
      RETURN
```

```
      END
C
      FUNCTION WP(X)
C
C     CALCULATE THE 1ST DERIV OF THE POTENTIAL AT A LOCAL COORD X
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /ZEDVAL/ENERGY,ZED,TRANSF,MATCH,
     X               MAP,NP,ORDER,REGION,TYPE,MAXREG,MAXTYP,MAXPT
      INTEGER REGION,TYPE,MAP(20),NP(20),ORDER(20)
      REAL*8 ZED(20,4,20),TRANSF(20,2),MATCH(20,2,2)
      WP=TRANSF(REGION,2)*VP(TRANSF(REGION,1)+TRANSF(REGION,2)*X)
      RETURN
      END
C
      FUNCTION WPP(X)
C
C     CALCULATE THE 2ND DERIV OF THE POTENTIAL AT A LOCAL COORD X
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /ZEDVAL/ENERGY,ZED,TRANSF,MATCH,
     X               MAP,NP,ORDER,REGION,TYPE,MAXREG,MAXTYP,MAXPT
      INTEGER REGION,TYPE,MAP(20),NP(20),ORDER(20)
      REAL*8 ZED(20,4,20),TRANSF(20,2),MATCH(20,2,2)
      WPP=VPP(TRANSF(REGION,1)+TRANSF(REGION,2)*X)
      RETURN
      END
C
C$PRINTOFF
C
      SUBROUTINE CONVER(FUN,HI,XHI,LOH,XLOH,TARGET,TOL,BEST,XBEST)
C
C     ROUTINE TO FIND X GIVEN F(X)
C
C     ON ENTRANCE
C           FUN    =      FUNCTION F
C           XHI    =      AN UPPER BOUND FOR X
C           HI     =      F(XHI)
C           XLOH   =      A LOWER BOUND FOR X
C           LOH    =      F(XLOH)
C           TARGET =      THE GIVEN F(X)
C           TOL    =      RELATIVE TOLERENCE
C
C     ON EXIT
C           XBEST  =      AN ESTIMATE OF THE TARGET X WITHIN TOLERENCE
C                         BOUNDS
C           BEST   =      F(XBEST)
C     NOTE:
C     (1)   TARGET-TOL < BEST < TARGET+TOL
C
C     (2)   FUNCTION FUN SHOULD PLACE VALUES IN COMMON ERROR
C                 IF NO ERROR   ==> IERROR=0
C                 IF ERROR      ==> IERROR=1
C
      IMPLICIT REAL*8 (A-H,O-Z)
```

```
      COMMON /ERROR/IERROR
      REAL*8 LOH,LOW,LOWER
      DATA MAXISW/2/
C
C     INITIALIZE
C
      IF (HI.GT.TARGET) THEN DO
      HIGH=HI
      XHIGH=XHI
      LOW=LOH
      XLOW=XLOH
      ELSE DO
      HIGH=LOH
      XHIGH=XLOH
      LOW=HI
      XLOW=XHI
      END IF
      LOWER=TARGET-TOL
      UPPER=TARGET+TOL
      BEST=2.D0*UPPER
      IVAL=ISW=0
C
C     LOOP
C
      WHILE (BEST.LT.LOWER.OR.BEST.GT.UPPER.OR.ISW.EQ.0) DO
      XBEST=XLOW+(TARGET-LOW)*(XHIGH-XLOW)/(HIGH-LOW)
      BEST=FUN(XBEST)
      IVAL=IVAL+1
      WRITE(6,10) XBEST,BEST,XHIGH,HIGH,XLOW,LOW
10    FORMAT(' XBEST=',D22.10,' BEST=',D22.10,/
     X       ' XHIGH=',D22.10,' HIGH=',D22.10,/
     X       ' XLOW =',D22.10,'  LOW=',D22.10,/)
      IF (BEST.LT.LOW.OR.BEST.GT.HIGH) IERROR=2
      IF (IERROR.NE.0) RETURN
      IF (BEST.GE.TARGET) THEN DO
            IF (ISW.NE.MAXISW) THEN DO
            IF (ISW.LT.0) ISW=0
            ISW=ISW+1
            ELSE DO
            ISW=0
            XLOW=XBEST-(XHIGH-XBEST)
            LOW=FUN(XLOW)
            IVAL=IVAL+1
            END IF
            HIGH=BEST
            XHIGH=XBEST
      ELSE DO
            IF (ISW.NE.-MAXISW) THEN DO
            IF (ISW.GT.0) ISW=0
            ISW=ISW-1
            ELSE DO
            ISW=0
            XHIGH=XBEST+(XBEST-XLOW)
            HIGH=FUN(XHIGH)
            IVAL=IVAL+1
```

```
            END IF
            LOW=BEST
            XLOW=XBEST
      END IF
      END WHILE
C
      WRITE(6,40) IVAL
40    FORMAT(' A TOTAL OF',I4,' FUNCTION EVALUATIONS NEEDED')
      RETURN
      END
C
      SUBROUTINE DDFSYS(N,F,EPS,H,X,Y,PAR)
C-----------------------------------------------------------------
C     'EPS' IS SINGLE PRECISION (REAL*4), WHEREAS 'X' 'Y' AND 'H' ARE
C     DOUBLE PRECISION (REAL*8)
C     N=NUMBER OF FIRST ORDER DIFF. EQUN. (N.LE.5),WHICH ARE OF THE FORM
C     DZ=F(X,Y1,Y2,.....)AND ARE PROVIDED BY SUBROUTINE F(X,Y,DZ,PAR).
C     'PAR'=UPTO 10 DOUBLE PRECISION PARAMETERS
C     EPS=ACCURACY PER STEP,SHOULD BE LARGER THAN 10.**(-D+3)
C     WHERE D IS THE NUMBER OF DIGITS OF THE MACHINE REPRESENTATION.
C     H=SUGGESTED STEP WIDTH,H IS AUTOMATICALLY ADJUSTED.
C     RETURNING FROM DIFSYS (X,Y) IS REPLACED BY (X+H,Y(X+H)).
C     H IS THEN USED AS SUGGESTED STEP SIZE FOR NEXT INTEGRATION STEP
C-----------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      EXTERNAL F
      REAL*4 EPS
      REAL*8 DTT(5)/5*0.0D00/,DT(5,6)/30*0.0D00/,Y(5),D(6),YA(5),
     1DY(5),DZ(5),YG(8,5),YH(8,5),YL(5),YM(5),S(5),PAR(10)
      INTEGER R,SR
      LOGICAL KONV,BO,BH
      EP=ABS(EPS)
      N1=N
      HH=H
      IF(EP.LT.5.E-9)EP=5.E-9
      CALL F(X,Y,DZ,PAR)
      BH=.FALSE.
      DO 1 I=1,N1
      S(I)=0.
1     YA(I)=Y(I)
2     A=HH+X
      FC=1.5
      BO=.FALSE.
      M=1
      R=2
      SR=3
      JJ=0
      DO 3 J=1,10
      IF(BO)GO TO 10
      D(1)=2.25
      D(3)=9.
      D(5)=36.
      GO TO 12
10    D(1)=1.777777777777778
      D(3)=7.111111111111111
```

```
      D(5)=28.44444444444444
12    KONV=J.GT.5
      IF(J.GT.7)GO TO 13
      L=J-1
      IF(L-1)15,15,16
16    D(L)=M*M
      GO TO 15
13    L=6
      D(6)=64.
      FC=.6*FC
15    M=M+M
      G=HH/M
      B=G+G
      IF(BH.AND.J.LT.9)GO TO 17
      GO TO 18
17    DO 19 I=1,N1
      YM(I)=YH(J,I)
19    YL(I)=YG(J,I)
      GO TO 25
18    KK=(M-2)/2
      M=M-1
      DO 20 I=1,N1
      YL(I)=YA(I)
20    YM(I)=G*DZ(I)+YA(I)
      DO 21 K=1,M
      CALL F(X+K*G,YM,DY,PAR)
      DO 22 I=1,N1
      U=B*DY(I)+YL(I)
      YL(I)=YM(I)
      YM(I)=U
      U=DABS(U)
      IF(U-S(I))22,22,23
23    S(I)=U
22    CONTINUE
      IF(K.EQ.KK.AND.K.NE.2)GO TO 24
      GO TO 21
24    JJ=1+JJ
      DO 26 I=1,N1
      YH(JJ,I)=YM(I)
26    YG(JJ,I)=YL(I)
21    CONTINUE
25    CALL F(A,YM,DY,PAR)
      DO 30 I=1,N1
      V=DTT(I)
      C=.5*(YM(I)+YL(I)+G*DY(I))
      TA=C
      DTT(I)=C
      L=MAX0(L,1)
      DO 31 K=1,L
      B1=D(K)*V
      B=B1-C
      U=V
      IF(B.NE.0)GO TO 32
      GO TO 33
32    B=(C-V)/B
```

```
      U=C*B
      C=B1*B
   33 V=DT(I,K)
      DT(I,K)=U
   31 TA=U+TA
      IF(DABS(Y(I)-TA).GT.EPS(I))KONV=.FALSE.
   30 Y(I)=TA
      IF(KONV) GO TO 50
      D(2)=4.
      D(4)=16.
      BO=.NOT.BO
      M=R
      R=SR
    3 SR=M+M
      BH=.NOT.BH
      HH=HH/2.
      GO TO 2
   50 H=FC*HH
      X=A
      RETURN
      END
C
      SUBROUTINE LANGER(X,Y,DZ,PAR)
C
C     A DDFSYS COMPATIBLE ROUTINE USED TO CALCULATE THE EXACT
C           VALUES OF Z ETC.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(5),DZ(5),PAR(10)
      DZ(1)=Y(2)
      DZ(2)=Y(3)*Y(2)
      DZ(3)=2.D0*(W(X)-Y(1)*Y(2)**2)+.5D0*Y(3)**2
      DZ(4)=0
      RETURN
      END
C
      SUBROUTINE ZPRIME(X,Y,DZ,PAR)
C
C     A DDFSYS COMPATIBLE ROUTINE USED TO CALCULATE THE ZEROTH
C           ORDER APPROXIMATION OF Z ETC.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(5),DZ(5),PAR(10)
      DZ(1)=DSQRT(W(X)/Y(1))
      RETURN
      END
C
```

```
      SUBROUTINE AIRY(AI,BI,AIP,BIP,Z,IER)
C
C     AIRY FUNCTIONS AI(Z), BI(Z)
C     AND THEIR DERIVATIVES AIP(Z), BIP(Z).
C
C     WRITTEN BY TIM SCHOLL
C
C     SEE REFERENCE (17) FOR EXPLANATION OF SOME EXPANSIONS
C
C     ON ENTRANCE
C           IER=0     ..... DON'T PRINT ERROR MESSAGES
C           IER=1     ..... PRINT ERROR MESSAGES
C
C     ON EXIT
C           IER=0     ..... NO ERRORS
C           IER=1     ..... OVERFLOW  AI=AIP=0. ,BI=BIP=1E75
C           IER=-1    ..... ARGUEMENT TO TRIG FUNCTION TOO LARGE
C
      IMPLICIT REAL*8 (A-H,O-Z)
      IF (Z.LE.-80.0D00)    CALL AREA4(AI,BI,AIP,BIP,Z,IER,&99)
      IF (Z.LE.-3.60D00)    CALL AREA3(AI,BI,AIP,BIP,Z,IER,&99)
      IF (Z.LE.1.0D00)      CALL AREA2(AI,BI,AIP,BIP,Z,IER,&99)
      CALL AREA1(AI,BI,AIP,BIP,Z,IER)
99    RETURN
      END
C
      SUBROUTINE AREA1(AI,BI,AIP,BIP,Z,IER)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION CVAI(12),CVBI(14),CVAIP(12),CVBIP(14)
C     COEFFICIENT VECTOR--AI*****
      DATA CVAI( 1)/-5.445045903973871D-02/
      DATA CVAI( 2)/ 1.228202483907300D-02/
      DATA CVAI( 3)/-2.107999475153601D-03/
      DATA CVAI( 4)/ 4.499311380490556D-04/
      DATA CVAI( 5)/-1.107885512234551D-04/
      DATA CVAI( 6)/ 3.029414769197648D-05/
      DATA CVAI( 7)/-8.581599138196050D-06/
      DATA CVAI( 8)/ 3.124271524032186D-06/
      DATA CVAI( 9)/-5.899825616934007D-07/
      DATA CVAI(10)/ 5.786979221922356D-07/
      DATA CVAI(11)/ 4.449623915138266D-08/
      DATA CVAI(12)/ 1.155036393855280D-07/
C     COEFFICIENT VECTOR--BI****
      DATA CVBI( 1)/ 8.227956360447162D-02/
      DATA CVBI( 2)/-6.316641547042471D-03/
      DATA CVBI( 3)/-1.913197185955132D-02/
      DATA CVBI( 4)/ 1.720510408372530D-03/
      DATA CVBI( 5)/ 2.383014346077916D-03/
      DATA CVBI( 6)/-1.134023349668820D-03/
      DATA CVBI( 7)/ 1.233865816914231D-03/
      DATA CVBI( 8)/ 1.579050652106840D-03/
      DATA CVBI( 9)/ 9.623176146072881D-04/
      DATA CVBI(10)/ 1.074857412313187D-03/
      DATA CVBI(11)/ 6.749123807882790D-04/
      DATA CVBI(12)/ 3.194142054489408D-04/
```

```
      DATA CVBI(13)/ 1.7394425478044740-04/
      DATA CVBI(14)/ 2.4729848909904920-05/
C     COEFFICIENT VECTOR--AIP*****
      DATA CVAIP( 1)/ 7.897729009600677D-02/
      DATA CVAIP( 2)/-1.510260266172028D-02/
      DATA CVAIP( 3)/ 2.463279388831557D-03/
      DATA CVAIP( 4)/-5.014040093621199D-04/
      DATA CVAIP( 5)/ 1.235657928041223D-04/
      DATA CVAIP( 6)/-3.686442633555789D-05/
      DATA CVAIP( 7)/ 2.841490054306729D-06/
      DATA CVAIP( 8)/-1.283878874251164D-05/
      DATA CVAIP( 9)/-6.385677045695181D-06/
      DATA CVAIP(10)/-8.177544529803375D-06/
      DATA CVAIP(11)/-2.946959206582392D-06/
      DATA CVAIP(12)/-2.767821448973843D-06/
C     COEFFICIENT VECTOR--BIP*****
      DATA CVBIP( 1)/-1.155463163390965D-01/
      DATA CVBIP( 2)/ 2.649705118599784D-03/
      DATA CVBIP( 3)/ 2.131461302832706D-02/
      DATA CVBIP( 4)/-1.325904771351788D-03/
      DATA CVBIP( 5)/-2.703662585321253D-03/
      DATA CVBIP( 6)/ 1.150685677225992D-03/
      DATA CVBIP( 7)/-1.280847875449697D-03/
      DATA CVBIP( 8)/-1.720672591437691D-03/
      DATA CVBIP( 9)/-1.057713441085448D-03/
      DATA CVBIP(10)/-1.158533819253666D-03/
      DATA CVBIP(11)/-7.377105071484384D-04/
      DATA CVBIP(12)/-3.497253573830967D-04/
      DATA CVBIP(13)/-1.887973911279085D-04/
      DATA CVBIP(14)/-2.894453443570144D-05/
C     OTHER COEFFICIENTS
      DATA SQRTPI/1.772453850905516/
C
C     LARGE Z, Z > 1.0
      IF (Z.GT.40.8D00) THEN DO
      IF (IER.EQ.1) THEN DO
      WRITE (6,1)
1     FORMAT('-WARNING!!!  EXPONENT OVERFLOW.',/,
     *' AI  EQUALS 0',/,
     *' AIP EQUALS 0',/,
     *' BI  EQUALS 1.0E+75',/,
     *' BIP EQUALS 1.0E+75')
      END IF
      AI=AIP=0.D00
      BI=BIP=1.0D+75
      IER=+1
      ELSE DO
      SUB=.6666666666666667*Z*DSQRT(Z)
      Z0=1.D00/SUB
      ZCOR=1.33333333333333D 00*Z0-1.0D00
      CALL CNPS(Y,ZCOR,CVAI,12)
      Y=(Y/SUB)+1.0D00
      AI=+0.50D00*Y*DEXP(-SUB)/(DSQRT(DSQRT(Z))*SQRTPI)
      CALL CNPS(Y,ZCOR,CVAIP,12)
      Y=(Y/SUB)+1.0D00
```

```
      AIP=-0.50D00*Y*DEXP(-SUB)*DSQRT(DSQRT(Z))/SQRTPI
      ZCOR=1.376873909378290D 00*Z0-1.065310864067434
      CALL CNPS(Y,ZCOR,CVBI,14)
      Y=(Y/SUB)+1.0D00
      BI=Y*DEXP(SUB)/(DSQRT(DSQRT(Z))*SQRTPI)
      CALL CNPS(Y,ZCOR,CVBIP,14)
      Y=(Y/SUB)+1.0D00
      BIP=Y*DEXP(SUB)*DSQRT(DSQRT(Z))/SQRTPI
      IER=0
      END IF
      RETURN
      END
C
      SUBROUTINE AREA2(AI,BI,AIP,BIP,Z,IER,*)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION CVAI(16),CVBI(16),CVAIP(16),CVBIP(16)
C     COEFFICIENT VECTOR--AI*****
      DATA CVAI( 1)/ 8.353676583513726D-02/
      DATA CVAI( 2)/ 3.326243057241334D-01/
      DATA CVAI( 3)/-2.945453051953963D-01/
      DATA CVAI( 4)/-7.848440599679570D-02/
      DATA CVAI( 5)/ 1.232075953712138D-01/
      DATA CVAI( 6)/-2.497938206372120D-02/
      DATA CVAI( 7)/-1.158731541242598D-02/
      DATA CVAI( 8)/ 6.281749158359338D-03/
      DATA CVAI( 9)/-4.731488472076063D-04/
      DATA CVAI(10)/-4.069834887026604D-04/
      DATA CVAI(11)/ 1.253181056974273D-04/
      DATA CVAI(12)/-1.596034427055331D-06/
      DATA CVAI(13)/-6.478601227972824D-06/
      DATA CVAI(14)/ 1.321405553651502D-06/
      DATA CVAI(15)/ 3.815480415977664D-08/
      DATA CVAI(16)/-5.872827832431077D-08/
C     COEFFICIENT VECTOR--BI*****
      DATA CVBI( 1)/ 2.890781685665552D-01/
      DATA CVBI( 2)/ 6.018552281473439D-01/
      DATA CVBI( 3)/ 4.121103023807179D-01/
      DATA CVBI( 4)/-1.674068616700008D-01/
      DATA CVBI( 5)/ 3.162375338106135D-02/
      DATA CVBI( 6)/ 5.288178479510745D-02/
      DATA CVBI( 7)/-1.360448621927808D-02/
      DATA CVBI( 8)/-7.688081389564037D-04/
      DATA CVBI( 9)/ 1.989127261940178D-03/
      DATA CVBI(10)/-3.158790170775661D-04/
      DATA CVBI(11)/-4.885337970872278D-05/
      DATA CVBI(12)/ 3.403901248098371D-05/
      DATA CVBI(13)/-3.410106414046498D-06/
      DATA CVBI(14)/-8.430603277137648D-07/
      DATA CVBI(15)/ 3.302966882264039D-07/
      DATA CVBI(16)/-2.013749214127708D-08/
C     COEFFICIENT VECTOR--AIP*****
      DATA CVAIP( 1)/ 5.470559688919087D-03/
      DATA CVAIP( 2)/-1.464288889259264D-01/
      DATA CVAIP( 3)/-2.782974078363954D-01/
      DATA CVAIP( 4)/ 3.658238191524662D-01/
```

```
      DATA CVAIP( 5)/-7.355548107104228D-02/
      DATA CVAIP( 6)/-6.272432747214548D-02/
      DATA CVAIP( 7)/ 3.505053039582666D-02/
      DATA CVAIP( 8)/-2.268760121033407D-03/
      DATA CVAIP( 9)/-3.186200110356446D-03/
      DATA CVAIP(10)/ 1.022713653099007D-03/
      DATA CVAIP(11)/-1.112740955366838D-06/
      DATA CVAIP(12)/-6.700088905496098D-05/
      DATA CVAIP(13)/ 1.416872314179926D-05/
      DATA CVAIP(14)/ 5.942349252532730D-07/
      DATA CVAIP(15)/-7.625956827438861D-07/
      DATA CVAIP(16)/ 1.155578036520587D-07/
C     COEFFICIENT VECTOR--BIP*****
      DATA CVBIP( 1)/ 1.548618331681935D-01/
      DATA CVBIP( 2)/ 7.691102354013728D-01/
      DATA CVBIP( 3)/-2.136287063043775D-01/
      DATA CVBIP( 4)/ 5.239666163024924D-02/
      DATA CVBIP( 5)/ 2.230848421926327D-01/
      DATA CVBIP( 6)/-5.759900977237984D-02/
      DATA CVBIP( 7)/-6.835967313120824D-03/
      DATA CVBIP( 8)/ 1.338090814244242D-02/
      DATA CVBIP( 9)/-2.156270198900254D-03/
      DATA CVBIP(10)/-4.565047617742590D-04/
      DATA CVBIP(11)/ 3.158264692108724D-04/
      DATA CVBIP(12)/-3.170010485956567D-05/
      DATA CVBIP(13)/-9.768927175521692D-06/
      DATA CVBIP(14)/ 3.896608455151422D-06/
      DATA CVBIP(15)/-2.408393901199891D-07/
      DATA CVBIP(16)/-1.071314679028117D-07/
C
C     SMALL Z,  -3.6 < Z < 1.0 **********
      CORZ=.4347826086956521D 00*Z+.5652173913043479D 00
      CALL CNPS(AI,CORZ,CVAI,16)
      CALL CNPS(BI,CORZ,CVBI,16)
      CALL CNPS(AIP,CORZ,CVAIP,16)
      CALL CNPS(BIP,CORZ,CVBIP,16)
      IER=0
      RETURN 1
      END
C
      SUBROUTINE AREA3(AI,BI,AIP,BIP,Z,IER,*)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION CVCHI(7),CVPSI(7),CVFAX(6),CVGAX(6)
C     COEFFICIENT VECTOR--FAX*****
      DATA CVFAX( 1)/ 9.984079740170277D-01/
      DATA CVFAX( 2)/-1.566213391555513D-03/
      DATA CVFAX( 3)/ 2.441839992342595D-05/
      DATA CVFAX( 4)/-1.009622526114867D-06/
      DATA CVFAX( 5)/ 7.192640017489443D-08/
      DATA CVFAX( 6)/-7.880984124842139D-09/
C     COEFFICIENT VECTOR--GAX*****
      DATA CVGAX( 1)/ 1.002246049711181D 00/
      DATA CVGAX( 2)/ 2.215105970219821D-03/
      DATA CVGAX( 3)/-2.928594564290988D-05/
      DATA CVGAX( 4)/ 1.141573497352018D-06/
```

```
      DATA CVGAX( 5)/-7.8784990327366540-08/
      DATA CVGAX( 6)/ 8.5632341936011360-09/
C     COEFFICIENT VECTOR--CHI*****
      DATA CVCHI( 1)/-1.5449541877122600-01/
      DATA CVCHI( 2)/ 1.7030527324159390-03/
      DATA CVCHI( 3)/-4.8561598372366860-05/
      DATA CVCHI( 4)/ 2.8435999618583380-06/
      DATA CVCHI( 5)/-2.5163911534040930-07/
      DATA CVCHI( 6)/ 2.3194327234585490-08/
      DATA CVCHI( 7)/-2.8636134938945150-08/
C     COEFFICIENT VECTOR--PSI*****
      DATA CVPSI( 1)/ 2.1640732988132290-01/
      DATA CVPSI( 2)/-2.2799297426376040-03/
      DATA CVPSI( 3)/ 5.9389563473983450-05/
      DATA CVPSI( 4)/-3.3184933528119670-06/
      DATA CVPSI( 5)/ 3.8831966914958940-07/
      DATA CVPSI( 6)/-1.2962513631149410-07/
      DATA CVPSI( 7)/ 4.8344362656902960-08/
C     OTHER COEFFICIENTS
      DATA PI4/.7853981633974483/
      DATA PI/3.1415926535897930/
C
C     LARGE Z  ,-80.0 < Z < -3.6 ********
      Z=DABS(Z)
      ZCUB=1.D00/Z**3
      CORZ=9.3312000000000020 01*ZCUB-1.0D00
      CALL CNPS(CHIVAL,CORZ,CVCHI,7)
      CALL CNPS(PSIVAL,CORZ,CVPSI,7)
      CORZ=9.3320503830911600 01*ZCUB-1.0001822666090450 00
      CALL CNPS(FMXVAL,CORZ,CVFAX,6)
      CALL CNPS(GMXVAL,CORZ,CVGAX,6)
      CHIVAL=(CHIVAL*ZCUB)+1.0D00
      CHIVAL=.66666666666666670*Z*DSQRT(Z)*CHIVAL+PI4
      PSIVAL=(PSIVAL*ZCUB)+1.0D00
      PSIVAL=.66666666666666670*Z*DSQRT(Z)*PSIVAL-PI4
      FMXVAL=DSQRT(FMXVAL/(PI*DSQRT(Z)))
      GMXVAL=DSQRT(GMXVAL*DSQRT(Z)/PI)
      AI=FMXVAL*DSIN(CHIVAL)
      BI=FMXVAL*DCOS(CHIVAL)
      AIP=GMXVAL*DSIN(PSIVAL)
      BIP=GMXVAL*DCOS(PSIVAL)
      Z=-Z
      IER=0
      RETURN 1
      END
C
      SUBROUTINE AREA4(AI,BI,AIP,BIP,Z,IER,*)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION CVCHI(7),CVPSI(7),CVFAX(3),CVGAX(5)
C     COEFFICIENT VECTOR--FAX*****
      DATA CVFAX( 1)/-1.5091792141807890-01/
      DATA CVFAX( 2)/ 4.7619856143599140-03/
      DATA CVFAX( 3)/ 1.1259429240751280-05/
C     COEFFICIENT VECTOR--GAX*****
      DATA CVGAX( 1)/ 2.1253905353066550-01/
```

```
      DATA CVGAX( 2)/-5.9507760562396880-03/
      DATA CVGAX( 3)/ 2.5400584572417020-04/
      DATA CVGAX( 4)/-2.4803524291385560-05/
      DATA CVGAX( 5)/ 8.0630940160670590-06/
C     COEFFICIENT VECTOR--CHI*****
      DATA CVCHI( 1)/-1.5449541877122600-01/
      DATA CVCHI( 2)/ 1.7030527324159390-03/
      DATA CVCHI( 3)/-4.8561598372366860-05/
      DATA CVCHI( 4)/ 2.8435999618583380-06/
      DATA CVCHI( 5)/-2.5163911534040930-07/
      DATA CVCHI( 6)/ 2.3194327234585490-08/
      DATA CVCHI( 7)/-2.8636134938945150-08/
C     COEFFICIENT VECTOR--PSI*****
      DATA CVPSI( 1)/ 2.1640732988132290-01/
      DATA CVPSI( 2)/-2.2799297426376040-03/
      DATA CVPSI( 3)/ 5.9389563473983450-05/
      DATA CVPSI( 4)/-3.1849335281196700-06/
      DATA CVPSI( 5)/ 3.8831966914958940-07/
      DATA CVPSI( 6)/-1.2962513631149410-07/
      DATA CVPSI( 7)/ 4.8344362656902960-08/
C     OTHER COEFFICIENTS
      DATA PI4/.7853981633974483/
      DATA PI/3.1415926535897930/
C
C     LARGE Z , Z < -80.0 *********
      IF (Z.LT.-1.0D+10) THEN DO
      IF (IER.EQ.1) THEN DO
      WRITE (6,1) FMXVAL,GMXVAL
1     FORMAT('-WARNING!!!  OUT OF RANGE FOR TRIG FUNCTIONS.'/,
     *' ABS(AI,BI)    <= ',1PD23.16,/,
     *' ABS(AIP,BIP) <= ',1PD23.16)
      END IF
      AI=AIP=BI=BIP=0.0D00
      IER=-1
      ELSE DO
      Z=DABS(Z)
      ZCUB=1.D00/Z**3
      CORZ=9.3312000000000020 01*ZCUB-1.0D00
      CALL CNPS(CHIVAL,CORZ,CVCHI,7)
      CALL CNPS(PSIVAL,CORZ,CVPSI,7)
      CALL CNPS(FMXVAL,CORZ,CVFAX,3)
      CALL CNPS(GMXVAL,CORZ,CVGAX,5)
      CHIVAL=(CHIVAL*ZCUB)+1.0D00
      CHIVAL=.66666666666666670*Z*DSQRT(Z)*CHIVAL+PI4
      PSIVAL=(PSIVAL*ZCUB)+1.0D00
      PSIVAL=.66666666666666670*Z*DSQRT(Z)*PSIVAL-PI4
      FMXVAL=(FMXVAL*ZCUB)+1.0D00
      FMXVAL=DSQRT(FMXVAL/(PI*DSQRT(Z)))
      GMXVAL=(GMXVAL*ZCUB)+1.0D00
      GMXVAL=DSQRT(GMXVAL*DSQRT(Z)/PI)
      AI=FMXVAL*DSIN(CHIVAL)
      BI=FMXVAL*DCOS(CHIVAL)
      AIP=GMXVAL*DSIN(PSIVAL)
      BIP=GMXVAL*DCOS(PSIVAL)
      Z=-Z
```

```
      IER=0
      END IF
      RETURN 1
      END
C
      SUBROUTINE CNPS(Y,X,C,N)
C
      DIMENSION C(N)
      DOUBLE PRECISION C,Y,X,H0,H1,H2,ARG
C
C     TEST OF DIMENSION
      IF (N) 1,1,2
1     RETURN
C
2     IF (N-2) 3,4,4
3     Y=C(1)
      RETURN
C
C     INITIALIZATION
4     ARG=X+X
      H1=0.D00
      H0=0.D00
C
      DO 5 I=1,N
      K=N-I
      H2=H1
      H1=H0
5     H0=ARG*H1-H2+C(K+1)
      Y=0.5D00*(C(1)-H2+H0)
      RETURN
      END
C$PRINTON
$ENTRY
$IBSYS
$STOP
//
```

References

1. R.G.Gordon, in Methods in Computational Physics, Vol. 10, ed. B.Alder, S.Fernbach, M.Rotenberg, Academic Press, New York, 1971, pp 81-109
2. R.G.Gordon, J. Chem. Phys. 51, 229, 1969
3. P.Andresen, Chem. Phys. Let. 44, 317, 1976
4. R.de Vogelaere, J. Res. Natl. Bur. Std. US 54, 119, 1955
5. W.N.Sams and D.J.Kouri, J. Chem. Phys. 51, 4809, 1955
6. G.M.Murphy, Ordinary Differential Equations and Their Solutions, D.van Nostrand Co., Princeton, N.J., 1960
7. E.Hille, Lectures on Ordinary Differential Equations, Addison-Wesley, Reading, 1969
8. N.Froman and P.O.Froman, JWKB Approximation, North Holland Publishing Co., Amsterdam, 1965
9. W.H.Miller, J. Chem. Phys. 48, 464, 1968
10. R.E.Langer, Phys. Rev. 51, 669, 1937
11. N.Y.Saad, A New Approach to the JWKB Method, Doctoral thesis, University of Windsor, 1980
12. R.Bulirsch and J.Stoer, Numerishe Mathematik 8, 1966
13. I.M.Gelfand and S.V.Fomin, Calculus of Variations, Prentice Hill, Englewood Cliffs, N.J., 1964
14. A.Blaquiere, Non-linear Systems Analysis, Academic Press, New York, 1966
15. D.D.Siljak, Non-linear Systems, Wiley, New York, 1969
16. R.Nevanlinna, Analytic Functions, Springer-Verlag, New York, 1970

17. M.Abromowitz and I.Stegun, Handbook of Mathematical Functions, Dover Publications Inc., New York, 1970

18. R.Eisberg and R.Resnik, Quantum Physics of Atoms, Molecules, Solids, Nuclei, and Particles, John Wiley and Sons, New York, 1974

19. C.Eckart, Phys. Rev. 35, 1303, 1930

20. D.Rapp, Quantum Mechanics, Holt,Rinehart, and Winston, New York, 1971

21. S.C.Miller and R.H.Good.Jr., Phys. Rev. 91, 174, 1953

22. G.A.Korn and T.M.Korn, Mathematical Handbook for Scientists and Engineers, second edition, McGraw-Hill, New York, 1968

23. D.R.Herschbach, J. Chem. Phys. 27, 975, 1957

24. D.R.Herschbach, J. Chem. Phys. 31, 91, 1959

Vita Auctoris

I was born in Windsor on April 26, 1958. In 1981, I received a B.Sc. in Physics from the University of Windsor. In that same year, I began work on an M.Sc. degree.