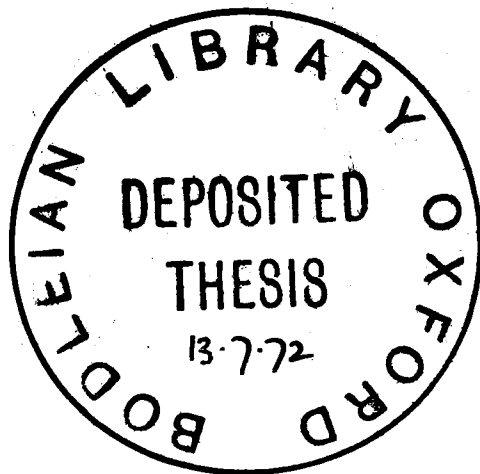




Applications of Quantum Mechanical
Hypervirial Theorems

by J.C.Nash

a thesis submitted in partial
fulfilment of the requirements
of the degree of Doctor of
Philosophy.

Mathematical Institute
Oxford
Hilary Term
1972

Addendum:

While the thesis was being prepared, Fig. 3 appeared in the "Handbook of Sensory Physiology", edited by Jung, published by Springer Verlag. It appeared as Fig. 5, page 252, volume VII/3, in the article by D. Whitteridge. I am grateful to Springer Verlag for their permission to reproduce this figure.

ABSTRACT

Two applications of the quantum mechanical hypervirial theorem are discussed. This states that for any system having eigenstates A and B of Hamiltonian H, an operator W has the matrix element relation

$$(A, (HW - WH) B) = (E_A - E_B) (A, W B)$$

providing W is such that H is Hermitean between A and W B.

The first application uses the operator $W = e^{itx}$ to derive a further relationship between certain matrix elements. For one dimensional systems in which the potential $V(x)$ involves only positive powers of x , this relationship becomes a differential equation in

$$f(t) = (A, e^{itx} B)$$

This differential equation has been solved analytically for the harmonic oscillator and in series for the quartic oscillator which were taken as example systems.

The second application is the off-diagonal hypervirial method, which seeks to require a wave function A to satisfy a number of hypervirial relations. In particular the wave function A is forced to obey as many conditions as it has variable parameters. To accomplish this a computational algorithm was developed which allowed one of these parameters to be non-linear. The results for oscillator test systems are not encouraging. The systems tested were the quartic and sixth power oscillators with a trial basis of harmonic oscillator eigenfunctions. In general the results were not as good as those obtainable by the variational method.

CONTENTS

	Page
Abstract	i
Contents	ii
Chapter 1: Introduction	1
Chapter 2: Hypervirial Theorems and Transition Moment Integrals	5
2.1 Introduction	5
2.2 Basic Equations	6
2.3 The Harmonic Oscillator	8
2.4 The Momentum Representation	13
2.5 The Quartic Oscillator	15
2.6 Conclusion	26
Chapter 3: The Off-Diagonal Hypervirial Method	27
3.1 Description of the Method	27
3.2 An Anomaly	29
3.3 Explanation	32
3.4 Wave Functions for Fixed Exponent	34
3.5 Expectation Values from Fixed k Functions	37
3.6 The Initial Function Problem	40
Chapter 4: Numerical Examples I. Methods	48
4.1 A Generalized Algebraic Eigenvalue Problem	48
4.2 Determination of a Non-linear Parameter	54

Chapter 4 (cont.)

4.3 Harmonic Oscillator Test Case	55
4.4 Hydrogen Atom Test Case	55
4.5 Calculations with Oscillators	58
4.6 Some Transformations Between Series Representations of Functions	62
4.7 The Hypervirial Operator $W = \exp(itx)$	68
4.8 The Computer Calculation	72
4.9 The Computer Program ANOSC	74
Chapter 5: Numerical Examples II. Results	76
5.1 The Test Systems	76
5.2 The Operator $W = \exp(itx)$	77
5.3 Hypervirial Operators x and d/dx	90
5.4 Excited States as Starting Data	99
5.5 Other Operator Sets	101
5.6 Perturbations in the Initial Function	113
5.7 Miscellaneous Ideas	118
5.8 Discussion	122
Appendix 1: Some Errors in Tables of Integrals	127
Appendix 2: A Theorem of Bradley and Hughes	131
Appendix 3: Some Matrix Elements for Harmonic Oscillator Eigenfunctions	133
Appendix 4: Earlier Research on Applications of Quantum Mechanical Hypervirial Theorems	134

Appendix 5: A Catalogue of Computer Programs Used

	136
Bibliography	139
Listings of Computer Programs	

Chapter 1

Introduction

CHAPTER 1

Introduction

A quantum mechanical system is defined by its Hamiltonian operator H , and possesses states which are described by vectors of a Hilbert space. In particular, there usually exist a number of stationary states which we may name ϕ_n , where n is some index, which have the particular property that they are eigenvectors of H (for H not a function of time), and we write

$$H \phi_n = E_n \phi_n \quad (1.1)$$

which is the time independent Schrödinger Equation (see, for instance, Gottfried, 1966, which is only one of many texts on Quantum Mechanics). E_n is real, and gives the energy for the system when it is in state ϕ_n .

Suppose now that there is a linear operator W defined on the Hilbert space which does not commute with H and which is not explicitly time dependent. Consider then

$$(\phi_m, [H, W] \phi_n) = (\phi_m, (HW - WH) \phi_n) \quad (1.2)$$

If W is such that

$$(\phi_m, HW \phi_n) = (H \phi_m, W \phi_n) \quad (1.3)$$

in other words, $W \phi_n$ belongs to the domain of H so that H is Hermitean between ϕ_m and $W \phi_n$, then we may apply Eq.(1.1) to get

$$(\phi_m, [H, W] \phi_n) = (E_m - E_n)(\phi_m, W \phi_n) \quad (1.4)$$

This is the quantum mechanical hypervirial theorem for

operator W . If $E_m = E_n$, and in particular when ϕ_m and ϕ_n coincide, the right hand side of Eq.(1.4) vanishes and we have what is called a diagonal hypervirial theorem. Eq.(1.4) as written is often called an off-diagonal hypervirial theorem or hypervirial relation. This terminology was introduced by Hirschfelder (1960).

In keeping with the title of this thesis, it is now pertinent to describe briefly the applications of such theorems which have been considered. There have been two major points of view. First, some attempt has been made to extend the theorem (1.4) by particular choice of the hypervirial operator W , and by this means improve the methods of calculating transition moment integrals. This work, which arose out of an idea of Professor C.A.Coulson, has been quite successful, and furnishes a fertile area for further investigation.

The second area of activity has been the off-diagonal hypervirial method. This has proceeded predominantly by means of test calculations on fairly simple systems, mainly certain types of oscillators. Numerical examples cannot, of course, yield any general rules about the use of such a method. They can, however, serve to increase or decrease confidence in various procedures, and after some time give rise to heuristic guidelines for their use. (It may be wise here to quote McMillan, 1970, "Heuristics

make problems manageable by reducing the number of alternatives. ... their greatest defense is that they 'reduce the sample space'.") Numerical tests may also lead us to suspect general rules which can then be proved formally, and this is indeed the case in Sections 3.2 and 3.3.

There have, of course, been other lines along which my work has proceeded which are not herein mentioned because they have not come to anything noteworthy.

Throughout the discussion units will be used in which the particle (usually an electron) has mass $m=1$, and $(1/2 \pi)$ times Planck's constant $\hbar = h/2 \pi = 1$. The electron charge e is also taken to be unity. These are usually called atomic or Coulomb units, though in this work they are usually used in conjunction with various oscillators.

Where publications are referred to, the names of the authors and date of publication will be given. An alphabetically ordered bibliography is included at the end of this volume. Furthermore, a briefly annotated bibliography of some other applications of hypervirial theorems is provided. A review of such material, including theorems in classical mechanics, has been submitted by the author for the Diploma in Advanced Mathematics (Nash, 1969).

The author wishes to thank those who have helped him

in this work, particularly his supervisor, Professor C.A.Coulson for suggesting a research topic which has shown itself to engage many diverse areas of physics and mathematics. The author is particularly grateful also to Dr. C.J.Bradley who acted as supervisor during Professor Coulson's absence in Hilary Term 1969, and who has encouraged several ideas in this work. Discussions with Professor S.T.Epstein, Mr. J.H.Rawnsely and Dr. J.A. Chapman have also been very helpful, and Professor L. Fox's suggestion of a computational scheme has been invaluable. The author gratefully acknowledges the support of a National Research Council of Canada Post-Graduate Scholarship and an allowance of computer time from the Oxford University Computing Service. Finally, the author would like to thank his wife Mary for her patient encouragement as well as considerable material help in the preparation of this volume.

Chapter 2

Hypervirial Theorems and Transition Moment Integrals

CHAPTER 2

Hypervirial Theorems and Transition Moment Integrals2.1 Introduction

The class of quantum mechanical matrix elements

$$(A, x^n B) = \int A^* x^n B d\tau \quad (2.1)$$

between the wave functions A and B for some system where x is one of the space coordinates, will collectively be called transition moment integrals. They have considerable interest in various forms of spectroscopy as a measure of the likelihood of absorption or emission of electromagnetic radiation (Herzberg, 1944; Brand and Speakman, 1960). The starting point for one useful application of hypervirial theorems is the Taylor expansion of the matrix element

$$(A, e^{itx} B) = \sum_{n=0}^{\infty} \frac{(it)^n}{n!} (A, x^n B) = f(t) \quad (2.2)$$

which is possible providing $(A, e^{itx} x^n B)$ converges uniformly for all finite n throughout the allowed domain of t, which shall be the real line (Widder, 1961, pages 347, 358). The convergence of the series itself will not be discussed, but presumed from the convergence of the integral $(A, e^{itx} B)$.

A proof of the uniform convergence of $(A, e^{itx} x^n B)$ may be carried out providing we accept certain conditions as to which wave functions are admissible (Kemble, 1958, pages 79, 198; Messiah, 1961, page 78). Following Kemble, any function describing a bound state of a non-relativ-

istic system may be restricted to be finite and to satisfy

$$\lim_{x \rightarrow \pm\infty} A P(x) = 0 \quad (2.3)$$

where $P(x)$ is any polynomial in x of finite degree. We only need require B to be finite, and thus $|B|$ is less than some finite number K greater than 0. In general,

$$(A, x^n e^{itx} B) = \int_{-\infty}^{\infty} A^* x^n e^{itx} B dx \quad (2.4)$$

but because A and B are both finite, the integral over any finite domain, say $[-a, a]$, will be finite, leaving only the ends of the domain, that is $(-\infty, -a)$ and (a, ∞)

Let us consider the latter of these two cases. Then

$$\int_a^{\infty} |A^* x^n e^{itx} B| dx \leq \int_a^{\infty} |A| x^n K dx \quad (2.5)$$

But from Eq.(2.3), we may choose "a" large enough that

$$|A| < \frac{1}{P(x)} \quad x > a \quad (2.6)$$

(and also large enough that no singular points of the Hamiltonian defining A occur in the region of interest.

Thus A is continuous for this region.) We can then choose

$$P(x) = x^{n+2}, \text{ so} \quad (2.7)$$

$$\int_a^{\infty} |A^* x^n e^{itx} B| dx \leq K \int_a^{\infty} x^{-2} dx = \frac{K}{a} \quad (2.8)$$

to complete the demonstration via Weierstrass' M test

(Widder, 1961, page 348).

2.2 Basic Equations

It is straightforward to show, by means of the elementary commutation relations (Newing and Cunningham, 1967) that if the Hamiltonian H is given by

$$H = \frac{1}{2} \hat{p}^2 + V(x) = -\frac{1}{2} d^2/dx^2 + V(x) \quad (2.9)$$

where $V(x)$ is the one-dimensional potential, then the single and double commutators of the operator $W = e^{itx}$ with H are

$$[H, e^{itx}] = \frac{1}{2}t(\hat{p} \cdot e^{itx} + e^{itx} \cdot \hat{p}) \quad (2.10)$$

$$[H, [H, e^{itx}]] = \frac{1}{4}t^2(\hat{p}^2 \cdot e^{itx} + 2\hat{p} \cdot e^{itx} \cdot \hat{p} + e^{itx} \cdot \hat{p}^2) + it e^{itx} V'(x) \quad (2.11)$$

where the prime (') denotes differentiation with respect to the single independent variable. Eq.(2.11) can be cast into a more useful form by application of the relationship (essentially a commutation relation)

$$\hat{p} \cdot e^{itx} = e^{itx} \cdot (\hat{p} + t) \quad (2.12)$$

yielding

$$[H, [H, e^{itx}]] = t^2(H-V) \cdot e^{itx} + t^2 \cdot e^{itx} \cdot (H-V) + it \cdot e^{itx} \cdot V' - \frac{1}{4}t^4 e^{itx} \quad (2.13)$$

where the dependence of V on x has been omitted as the meaning is clear. The off-diagonal hypervirial theorem Eq.(1.4) then becomes for this particular operator

$$(E_A - E_B)^2 f(t) = t^2(A, (H-V) \cdot e^{itx} B) + t^2(A, e^{itx} \cdot (H-V) B) + it(A, e^{itx} \cdot V'(x) B) - \frac{1}{4}t^4 f(t) \quad (2.14)$$

Eq.(2.14) is valid for all suitably well-behaved $V(x)$, that is, $V(x)$ which, with its derivative $V'(x)$ is defined at all but a finite number of points on the real line and which is such that all integrals in the matrix elements of Eq.(2.14) are uniformly convergent.

However, a particularly interesting simplification occurs for $V(x)$ a power series in x with only positive

powers. Thus if

$$V(x) = \sum a_n x^n \quad (2.15)$$

and we use the fact that

$$(A, x^n e^{itx} B) = \frac{1}{i^n} \frac{d^n f}{dt^n} \quad (2.16)$$

where $i = (-1)^{\frac{1}{2}}$, which is true providing the integral on the left hand side is uniformly convergent (see the previous section for a proof of this convergence under certain conditions on A and B), then Eq.(2.14) becomes

$$\begin{aligned} (E_A - E_B)^2 f(t) = & t^2 ((E_A + E_B) f(t) - \sum \frac{2a_n}{i^n} \frac{d^n f(t)}{dt^n}) + \\ & + it \sum \frac{na_n}{i^{n-1}} \frac{d^{n-1} f(t)}{dt^{n-1}} - \frac{1}{4} t^2 f(t) \end{aligned} \quad (2.17)$$

If this differential equation for $f(t)$ is solved analytically or numerically we obtain values for the transition moment integrals by the Taylor expansion (2.2). In general, Eq.(2.17) will be difficult to solve. In the next few pages, solutions will be given for the harmonic and quartic oscillators.

In the case of the Coulombic potential, Eq.(2.14) yields an integro-differential equation which does not involve $f(t)$ alone. Consequently some other method of solution is needed in this situation, and this problem is still under consideration.

2.3 The Harmonic Oscillator

Let us now put $V(x) = \frac{1}{2} x^2 \quad (2.18)$

in Eq.(2.17); we arrive at

$$\left(-\frac{1}{4}t^4 + (E_A + E_B)t^2 - (E_A - E_B)^2 + t\frac{d}{dt} + t^2\frac{d^2}{dt^2} \right) f(t) = 0 \quad (2.19)$$

(This equation was derived independently by Professor J.G. Taylor, and his suggestion concerning the use of $W = e^{itx}$ was communicated to the author by Professor C.A.Coulson.) This equation possesses a solution in a reasonably simple form. Indeed we may surmise this from the integral we are required to evaluate for $f(t)$, which could be performed directly as in Appendix 1. The integrand is simply the product of two harmonic oscillator wave functions modified by a multiplicative factor of modulus 1. The reader is warned that some of the standard references for the evaluation of integrals are in error and is referred to Appendix 1.

However, to return to the differential equation (2.19) the algebraic manipulation may be somewhat simplified by the substitutions

$$\alpha = \frac{1}{4} ; \quad \beta = (E_A + E_B) ; \quad \gamma = -(E_A - E_B)^2 \quad (2.20)$$

A consideration of the limiting form of the equation for large t , that is,

$$\frac{d^2 f}{dt^2} - \alpha t^2 f = 0 \quad (2.21)$$

suggests that the substitution

$$f(t) = \exp\left(-\frac{1}{2}\alpha t^2\right) F(t) \quad (2.22)$$

might be useful to remove the dominant (i.e. t^4) term

in Eq.(2.19), and the resulting equations for $F(t)$ are

$$(\beta \pm 1)t^2 F(t) + \gamma F(t) + (t+t^3)F'(t) + t^2 F''(t) = 0 \quad (2.23)$$

(Here the ' again indicates differentiation with respect to the independent variable, this time t .) Both Eqs.(2.23) can be solved in series, that is, by substitution of

$$F(t) = t^c \sum_{n=0}^{\infty} b_n t^n \quad (b_0 \neq 0) \quad (2.24)$$

and they become

$$\sum_{n=0}^{\infty} b_n \left\{ t^{n+c+2} (\beta \pm (n+c+1)) + t^{n+c} ((n+c)^2 + \gamma) \right\} = 0 \quad (2.25)$$

with the resulting indicial equations

$$b_0 (c^2 + \gamma) = 0 \quad (2.26a)$$

$$b_1 ((c+1)^2 + \gamma) = 0 \quad (2.26b)$$

Rearrangement of Eqs.(2.25) gives

$$\sum_{n=0}^{\infty} t^{n+c+2} \left\{ b_n (\beta \pm (n+c+1)) + b_{n+2} ((n+c+2)^2 + \gamma) \right\} = 0 \quad (2.27)$$

(The conditions resulting from setting the coefficients of the lowest powers of t to zero are now omitted, as they are simply the indicial equations.) Thus we have the

recurrence relations for the b_n

$$b_{n+2} = - \frac{\beta \pm (n+c+1)}{(n+c+2)^2 + \gamma} b_n \quad (2.28)$$

The indicial equation (2.26a) immediately gives

$$c = \pm \sqrt{-\gamma} = \pm (E_A - E_B) \quad (2.29)$$

since $b_0 \neq 0$. As we wish $f(t)$ continuous at the origin, $t = 0$, in order to define the Taylor series (2.2), we

take the positive root and further suppose without loss of generality that $E_A \geq E_B$. Using this value of c in Eq.(2.26b) shows b_1 , and by the recurrence relation all odd index b_n , to be zero, i.e.

$$b_{2m+1} = 0 \quad (m=0,1,2,\dots) \quad (2.29a)$$

Considering now the even indexed coefficients, we are led to

$$\begin{aligned} b_{2m+2} &= - \frac{\beta \pm (2m+c+1)}{(2m+c+2)^2 - c^2} b_{2m} \\ &= - \frac{\beta \pm (2m+c+1)}{4(m+c+1)(m+1)} b_{2m} \\ &= \frac{(\frac{1}{2}(c \mp \beta + 1) + m)}{(\pm 2)(m+c+1)(m+1)} b_{2m} \end{aligned} \quad (2.30)$$

In the last line the upper signs are taken together, as are the lower signs - there are only two choices, not four. Hence,

$$b_{2m} = \frac{(\frac{1}{2}(c \mp \beta + 1))_m}{(\pm 2)^m (c+1)_m m!} b_0 \quad (2.31)$$

where

$$(k)_r = k(k+1)(k+2)\dots(k+r-1) = \frac{\Gamma(k+r)}{\Gamma(k)} \quad (2.32)$$

(Sneddon, 1961, page 19). But the confluent hypergeometric function in y is defined (Sneddon, 1961, page 35) as

$$M(\alpha, \gamma, y) = \sum_{r=0}^{\infty} \frac{(\alpha)_r y^r}{(\gamma)_r r!} \quad (2.33)$$

Thus we have

$$f(t) = b_0 t^c \exp(\mp \frac{1}{4} t^4) M(\frac{1}{2}(c \mp \beta + 1), c+1, \pm \frac{1}{4} t^2) \quad (2.34)$$

and have demonstrated, as an afterthought, Kummer's

relation (Sneddon, 1961, page 38)

$$M(\alpha, \gamma, x) = \exp(x) M(\gamma - \alpha, \gamma, -x) \quad (2.35)$$

It is, of course, well known that the energy levels of the harmonic oscillator are, in the units used here, simply $(n+\frac{1}{2})$, where $n=0,1,2,\dots$. So the energy level differences $(E_A - E_B)$ are integers. However, since the Taylor expansion (2.2) of the matrix element $f(t)$ contains only integral powers of t , the c in Eq.(2.34) must be an integer, and we have a verification of the integer nature of the energy differences. A further conclusion from the Eq.(2.34) is that the first non-vanishing transition moment between states with energies $n_A + \frac{1}{2}$ and $n_B + \frac{1}{2}$ with n_A greater than n_B is the moment of order $n_A - n_B$. This result is of course not new; it can be proved by means of the recurrence relation (Newing and Cunningham, 1967, page 87)

$$x \psi_n = (\frac{1}{2}n)^{\frac{1}{2}} \psi_{n-1} + (\frac{1}{2}(n+1))^{\frac{1}{2}} \psi_{n+1} \quad (2.36)$$

where ψ_j is the state of energy $j + \frac{1}{2}$. The $\{\psi_j\}$ form an orthonormal set, and the moment $(\psi_{n_A}, x^c \psi_{n_B})$ will be zero unless c is at least $(n_A - n_B)$, since each operation of x on the vector $x^q \psi_{n_B}$ only raises the index of the highest energy component, i.e. of energy $q + n_B + \frac{1}{2}$, by 1.

Eq.(2.34) is as yet an incomplete determination of $f(t)$, as we have not yet found b_0 . Let us consider the matrix element between states of energy $r + \frac{1}{2}$ and $s + \frac{1}{2}$ ($r \geq s$),

and as a consequence relabel $f(t)$ as $f_{rs}(t)$ and b_0 as b_{rs} . We recall the definitions

$$\beta = E_A + E_B = r+s+1 \quad (2.37a)$$

$$c = E_A - E_B = r-s \quad (2.37b)$$

to find (taking only one form of the solutions)

$$f_{rs}(t) = b_{rs} t^{r-s} \exp(-\frac{1}{2}t^2) M(-s, r-s+1, \frac{1}{2}t^2) \quad (2.38)$$

For small values of t , both $\exp(-\frac{1}{2}t^2)$ and M are equal to $1 + O(t^2)$, so

$$(\psi_r, e^{itx} \psi_s) = b_{rs} t^{r-s} + O(t^2) \quad (2.39)$$

Differentiating with respect to t a total of $r-s$ times, then putting $t=0$, yields

$$b_{rs} = \frac{i^{r-s}}{(r-s)!} (\psi_r, x^{r-s} \psi_s) \quad (2.40)$$

and the recurrence relation (2.36) used repeatedly allows us to write

$$b_{rs} = \frac{i^{r-s}}{(r-s)!} \left[\frac{r!}{s! 2^{r-s}} \right]^{\frac{1}{2}} = i^{r-s} \left[\frac{2^{s-r}}{(r-s)!} \binom{r}{s} \right]^{\frac{1}{2}} \quad (2.41)$$

Thus $f_{rs}(t)$ is determined completely.

2.4 The Momentum Representation

While $f_{rs}(t)$ for the harmonic oscillator may have intrinsic interest, it may be applied in a simple manner to the calculation of overlap integrals of harmonic oscillator wave functions on different centres (the author is indebted to Professor C.A.Coulson for this suggestion). The harmonic oscillator Hamiltonian

$$H = \frac{1}{2}(\hat{p}^2 + x^2) = \frac{1}{2}\left(-\frac{d^2}{dy^2} + y^2\right) \quad (2.42)$$

shows complete symmetry between coordinate and momentum representations, that is, the linear differential operator for the Hamiltonian in coordinate and momentum space is the same except for the independent variable, and y in Eq.(2.42) is replaced as appropriate by x or p . Thus the momentum wave function $\phi_r(p)$, for the state of energy $(r+\frac{1}{2})$ will have precisely the same form as its counterpart coordinate space wave function $\psi_r(x)$. These two wave functions must also satisfy the Fourier transform relationship

$$\phi_r(p) = T \psi_r(x) = (2\pi)^{-\frac{1}{2}} \int_{-\infty}^{\infty} \psi_r(x) e^{-ipx} dx \quad (2.43a)$$

$$\psi_r(x) = T^* \phi_r(p) = (2\pi)^{-\frac{1}{2}} \int_{-\infty}^{\infty} \phi_r(p) e^{+ipx} dx \quad (2.43b)$$

The definition of $f_{rs}(t)$, that is,

$$f_{rs}(t) = \int_{-\infty}^{\infty} \psi_r(x) e^{itx} \psi_s(x) dx \quad (2.44)$$

strongly suggests the convolution theorem. Defining

the convolution

$$\psi_r * \psi_s = (2\pi)^{-\frac{1}{2}} \int_{-\infty}^{\infty} \psi_r(\xi) \psi_s(x-\xi) d\xi \quad (2.45)$$

the convolution theorem (Carrier, Krook and Pearson, 1966; Sokolnikoff and Redheffer, 1966) is

$$T(\psi_r * \psi_s) = (T \psi_r)(T \psi_s) = \phi_r(p) \phi_s(p) \quad (2.46)$$

So

$$\begin{aligned} \psi_r * \psi_s &= T^{-1}(\phi_r(p) \phi_s(p)) = (2\pi)^{-\frac{1}{2}} \int_{-\infty}^{\infty} \phi_r(p) \phi_s(p) e^{ipx} dp \\ &= (2\pi)^{-\frac{1}{2}} f_{rs}(x) \end{aligned} \quad (2.47)$$

since $\phi_r(p)$ and $\phi_s(p)$ are respectively identical with

$\psi_r(p)$ and $\psi_s(p)$. Thus

$$f_{rs}(x) = \int_{-\infty}^{\infty} \psi_r(\xi) \psi_s(x - \xi) d\xi \quad (2.48)$$

By the parity of ψ_s , we may write

$$\psi_s(x - \xi) = (-1)^s \psi_s(\xi - x) \quad (2.49)$$

so replacing x by x_0 we have

$$\int_{-\infty}^{\infty} \psi_r(\xi) \psi_s(\xi - x_0) d\xi = (-1)^s f_{rs}(x_0) \quad (2.50)$$

which is simply the overlap integral between two similar harmonic oscillators whose centres are a distance x_0 apart. This is an alternative derivation to that of Katriel (1970), Katriel and Adam (1970) and Witschel (1970).

The material of the above three sections is in process of publication (Coulson and Nash, accepted).

2.5 The Quartic Oscillator

Let us now make the substitution

$$V(x) = kx^4 \quad (2.51)$$

into Eq.(2.17) and use the definitions (2.20) for γ and

β , to arrive at

$$(\gamma + t^2\beta - \frac{1}{4}t^4) f(t) - 2k(t^2 \frac{d^4 f}{dt^4} + t \frac{d^3 f}{dt^3}) = 0 \quad (2.52)$$

It does little good to try to remove the highest order term here, as we are still confronted with a three term recurrence relation, and as will be shown shortly, the four-term recurrence relation derived from Eq.(2.52) has

rapid convergence. We attempt a solution with

$$f(t) = \sum_{n=0}^{\infty} b_n t^{n+c} \quad (2.53)$$

which yields

$$\sum_{n=0}^{\infty} b_n \left(\gamma t^{n+c} + \beta t^{n+c+2} - \frac{1}{4} t^{n+c+4} - 2k(n+c)(n+c-1)^2(n+c-2)t^{n+c-2} \right) = 0 \quad (2.54)$$

The indicial equation ($n=0$) is

$$c(c-1)^2(c-2) = 0 \quad (2.55)$$

so that $c = 0, 1, 2$ are the roots and the possible lowest powers of t . In order to get all the solutions to the differential equation, we should need series which involved each of these c values. However, by noting the definition of $f(t)$, Eq.(2.2), where for $A=B$ we require $b_0 = 1$, we must use $c = 0$. This choice has been used throughout (the other c values, where it is appropriate to use them, will give the same results in this application). From Eq.(2.54) and $c = 0$ we derive the recurrence relation

$$b_{n+2} = \frac{\gamma b_n + \beta b_{n-2} - \frac{1}{4} b_{n-4}}{2k(n)_3(n+1)} \quad (2.56)$$

where $(n)_j$ is defined as in Eq.(2.32).

A considerable simplification results from the parity of the eigenfunctions of the quartic oscillator, since eigenfunctions of the same parity will give

$$f_{\text{even}} = \sum_{m=0}^{\infty} \frac{(-t^2)^m}{(2m)!} \quad g_m = \sum_{m=0}^{\infty} c_m t^{2m} \quad (2.57)$$

$$\text{where } g_m = (A, x^{2m} B) \quad (2.58)$$

while those of different parity give

$$f_{\text{odd}} = it \sum_{m=0}^{\infty} \frac{(-t^2)^m}{(2m+1)!} \quad h_m = i \sum_{m=0}^{\infty} d_m t^{2m+1} \quad (2.59)$$

$$\text{where} \quad h_m = (\bar{A}, x^{2m+1} \bar{B}) \quad (2.60)$$

We then have the following recurrence relations for the

c_j, d_j, g_j , and h_j .

$$c_{j+1} = \frac{\gamma c_j + \beta c_{j-1} - \frac{1}{4} c_{j-2}}{2k (2j)_3 (2j+1)} \quad (2.61a)$$

$$d_{j+1} = \frac{\gamma d_j + \beta d_{j-1} - \frac{1}{4} d_{j-2}}{2k (2j+1)_3 (2j+2)} \quad (2.61b)$$

$$g_{j+1} = \frac{-\gamma g_j + \beta g_{j-1} (2j-1)_2 + \frac{1}{4} g_{j-2} (2j-3)_4}{2k (2j)_2} \quad (2.61c)$$

$$h_{j+1} = \frac{-\gamma h_j + \beta h_{j-1} (2j)_2 + \frac{1}{4} h_{j-2} (2j-2)_4}{2k (2j+1)_2} \quad (2.61d)$$

While the simple relationships

$$c_j = \frac{(i^{2m})}{(2m)!} g_j \quad ; \quad d_j = \frac{(i^{2m})}{(2m)!} h_j \quad (2.62)$$

appear to make two of the recurrence relations (2.61a-d) redundant, it should be pointed out that the first two relations, which give the series coefficients in the expansion for $f(t)$, converge so fast that by $j = 35$ they have become smaller than the Oxford ICL 1906A computer can handle (absolute value smaller than a number $=10^{-76}$ approximately), while $70!$ is of order 10^{99} (Stirling's formula), which is also out of range. Thus if a solution of the differential equation is desired, the first two

relations are appropriate, while the latter two avoid the multiplication of very large numbers by very small ones when matrix elements are wanted.

In order to make use of these formulae, we need some starting data in the form of one matrix element and some energies for the relevant eigenstates (or eigenstate, if diagonal elements are to be generated). Let us take $k = \frac{1}{2}$ in Eq.(2.51) for simplicity, and variationally calculate wave functions in terms of harmonic oscillator wave functions for which the Hamiltonian is

$$H_0 = \frac{1}{2}(p^2 + 4x^2) \quad (2.63)$$

(Note that this implies an exponential parameter of 2.0)¹

This was achieved by means of the computer program ANOSC, which is described in Section 4.9. Wave functions were calculated using 10, 20, 40 and 100 basis functions, although half of these have coefficient zero by symmetry (McWeeny and Coulson, 1948).

It should be mentioned that wave functions have been published for the quartic oscillator by Chan and Stelman (1963). They have included computer print-out in their paper which unfortunately uses a rather irregular representation of floating point numbers. An example is 0.0₅4321 which represents 0.4321×10^{-5} , but which appears to be a poor printing of 0.054321.

The calculated wave functions were punched onto cards

¹ see pg. 77.

(five floating point numbers with 8 digit mantissa and 2 digit exponent per card) and a program called MOMENTS was used to obtain values of the transition moment integrals. This served to obtain starting values for the recurrence relations as well as to generate data for comparison purposes. Two versions of MOMENTS were tried, one using a harmonic oscillator basis, the other a power of x type basis. The latter was regarded as inferior on two grounds:

- 1) it uses a double rather than a single summation and requires many more integrals to be evaluated with consequent slowness of execution,

- 2) because of the increased number of arithmetic operations, rounding errors are more likely. Comparisons of data for a few instances show differences in at worst the 8 th figure. (The cases examined were the matrix elements of 40 component wave functions for state 0 with states 0,1, and 2, state 1 with state 1, and state 3 with state 3.)

The recurrence relations and matrix element calculations were carried through for all 10 combinations of the states 0,1,2, and 3 of the quartic oscillator, and typical results are collected in Tables (2.1). It is interesting to note how well the recurrence generated matrix elements compare with each other and with the directly computed values. Particularly noticeable is the closeness of the values

Table (2.1a) Matrix Elements ($\chi_0(N), x^n \chi_0(N)$)
from recurrence relation (upper value) and direct
calculation (lower value).

n	N=10	N=20	N=40	N=100
2	0.3620190718 0.3620190718	0.3620226465 0.3620226465	0.3620226463 0.3620226463	0.3620226463 0.3620226463
4	0.3534547134 0.3534526769	0.3534540305 0.3534540362	0.3534540301 0.3534540274	0.3534540301 0.3534540274
6	0.5303232251 0.5303748893	0.5303250543 0.5303251019	0.5303250540 0.5303250513	0.5303250540 0.5303250513
8	1.043462799 1.043763872	1.043469425 1.043469517	1.043469424 1.043469420	1.043469424 1.043469420
10	2.499191396 2.500245826	2.499188076 2.499188572	2.499188074 2.499188055	2.499188074 2.499188055
20	1260.764593 1368.027523	1260.770749 1266.585123	1260.770748 1260.770813	1260.770748 1260.770757
30	4271210.929 51973121.97	4271225.918 27984729.86	4271225.914 4276468.622	4271225.914 4271226.251
E_0	0.5301820701	0.5301810457	0.5301810452	0.5301810453

The last two entries in the row of energies E_0 ,
violate a theorem in

MacDonald J.K.L., 1933, Phys.Rev., 43, 830-833,

which states that the entry in the N=40 column should
be greater than that in the N=100 column. If the matrix
eigenvalue problem which defines these energies is solved
by an alternative algorithm, the values are in fact inter-
changed. Other minor changes occur in other energies. This
points to small rounding errors in either the setting up
of the matrices or the solution of the eigenproblem or
both. 10 digits is near to the limit of precision on the
37 bit ICL 1906A Computer.

Table (2.1b) Matrix Elements ($\chi_0(N), x^n \chi_1(N)$)
from recurrence relation (upper value) and direct
calculation (lower value).

n	N=10	N=20	N=40	N=100
1	0.6008024454 0.6008024454	0.6008049392 0.6008049392	0.6008049398 0.6008049398	0.6008049398 0.6008049398
3	0.5635514450 0.5635508116	0.5635418491 0.5635418514	0.5635418476 0.5635418476	0.5635418476 0.5635418476
5	0.8180869004 0.8182043518	0.8180815920 0.8180816416	0.8180815912 0.8180815913	0.8180815912 0.8180815913
7	1.504926191 1.565474483	1.564905506 1.564905721	1.564905502 1.564905503	1.564905502 1.564905503
9	3.656727138 3.658395873	3.656669520 3.656669929	3.656669511 3.656669514	3.656669511 3.656669514
19	1681.960242 1868.036260	1681.932335 1684.859391	1681.932330 1681.932351	1681.932330 1681.932345
29	5341334.914 82489388.04	5341256.532 17482246.03	5341256.518 5341726.922	5341256.518 5341256.764
E_0	0.5301820701	0.5301810457	0.5301810452	0.5301810453
E_1	1.899852046	1.899836518	1.899836515	1.899836515

Table (2.1c) Matrix Elements ($\chi_0(N), x^n \chi_2(N)$)

from recurrence relation (upper value) and direct calculation (lower value).

n	N=10	N=20	N=40	N=100
2	0.4684414105 0.4684414105	0.4684904303 0.4684904303	0.4684904265 0.4684904265	0.4684904265 0.4684904265
4	0.7985403507 0.7980874738	0.7983920425 0.7983920902	0.7983920268 0.7983920255	0.7983920268 0.7983920255
6	1.605289158 1.604974735	1.605088930 1.605089192	1.605088902 1.605088898	1.605088902 1.605088898
8	3.823717629 3.826794681	3.822943351 3.822943689	3.822943273 3.822943262	3.822943273 3.822943262
10	10.51831575 10.53822884	10.51593757 10.51594283	10.51593735 10.51593731	10.51593735 10.51593731
20	7737.072901 6890.701391	7734.915651 7782.080003	7734.915468 7734.916326	7734.915468 7734.915475
30	31566725.41 112368051.3	31557503.07 234121055.4	31557502.30 31638529.95	31557502.30 31557503.35
E_0	0.5301820701	0.5301810457	0.5301810452	0.5301810453
E_2	3.728314316	3.727848988	3.727848969	3.727848969

Table (2.1d) Matrix Elements ($\chi_1(N), x^n \chi_3(N)$)
 from recurrence relation (upper value) and direct
 calculation (lower value).

n	N=10	N=20	N=40	N=100
2	0.6699565572 0.6699565572	0.6693918784 0.6698918784	0.6698918686 0.6698918686	0.6698918686 0.6698918686
4	1.718359912 1.717826169	1.717858710 1.717858850	1.717858490 1.717858485	1.717858488 1.717858485
6	4.426510722 4.425137029	4.425401069 4.425403711	4.425400614 4.425400601	4.425400610 4.425400601
8	12.53629437 12.53261827	12.53216021 12.53217182	12.53215835 12.53215831	12.53215834 12.53215831
10	39.29104594 39.28614937	39.27862628 39.27869170	39.27862078 39.27862068	39.27862074 39.27862068
20	42003.76810 49901.73413	41988.01692 42336.59663	41988.00966 41988.01052	41988.00960 41988.00958
30	207687996.4 4808039692.	207605240.5 1875413616.	207605201.9 207688192.5	207605201.6 207605197.8
E_1	1.899852046	1.899836518	1.899836515	1.899836515
E_3	5.822771311	5.822372982	5.822372757	5.822372755

Table (2.1e) Matrix Elements ($\chi_2(N), x^n \chi_3(N)$)
 from recurrence relation (upper value) and direct
 calculation (lower value).

n	N=10	N=20	N=40	N=100
1	0.8381751205 0.8381751205	0.8389033085 0.8389033085	0.8389032896 0.8389032896	0.8389032896 0.8389032896
3	1.838432398 1.838570791	1.840147310 1.840147298	1.840146907 1.840146904	1.840146903 1.840146904
5	4.674803133 4.681344250	4.678588308 4.678591176	4.678587836 4.678587830	4.678587833 4.678587830
7	13.22776506 13.27750835	13.23895052 13.23896166	13.23894743 13.23894739	13.23894740 13.23894739
9	41.41739989 41.66548268	41.44885725 41.44890836	41.44885105 41.44885094	41.44885100 41.44885094
19	43289.77789 40458.49026	43319.64988 43591.68600	43319.64073 43319.64418	43319.64065 43319.64033
29	1298674382. -2986929774.	207746882.6 1461777192.	207746834.3 208126804.7	207746833.9 207746825.9
E_2	3.728314316	3.727848988	3.727848969	3.727848969
E_3	5.822771311	5.822372982	5.822372757	5.822372755

Table (2.1f) Matrix Elements ($\chi_3(N), x^n \chi_3(N)$)
 from recurrence relation (upper value) and direct
 calculation (lower value).

n	N=10	N=20	N=40	N=100
2	1.55658385 1.556583851	1.557909374 1.557909374	1.557909205 1.557909205	1.557909205 1.557909205
4	3.881847541 3.878576542	3.881581988 3.881585881	3.881581838 3.881581868	3.881581837 3.881581868
6	11.17635815 11.18980446	11.18487534 11.18489861	11.18487374 11.18487374	11.18487373 11.18487374
8	35.62569467 35.70027993	35.62411736 35.62420193	35.62411451 35.62411475	35.62411448 35.62411475
10	123.8755869 124.4079185	123.9442531 123.9449391	123.9442338 123.9442340	123.9442337 123.9442340
20	188018.0423 203902.2999	188065.7161 192107.0971	188065.6833 188065.7282	188065.6831 188065.6824
30	1134698424. 14034956500.	1134891837. 21530940480.	1134891618. 1139620675.	1134891616. 1134891558.
E_3	5.822771311	5.822372982	5.822372757	5.822372755

calculated using the recurrence relations and starting information based on a 10 component wave function and the elements calculated directly with more accurate wave functions. It is unwise to generalize from such numerical examples, but the success in this particular calculation is a hopeful sign that such recurrence relation methods may prove to give useful results from quite approximate wave functions. The amount of computation needed for the recurrence relations is, of course, very small beside that of calculating matrix elements directly.

2.6 Conclusion

The material discussed in the preceding sections offers at the moment only an interesting application of the hypervirial theorem. Before atomic or molecular systems can be approached by this means, some way will have to be found to solve the Eq.(2.14) or a related equation for the Coulombic potential. This is still being studied at time of writing. While it may be possible to find $(\psi_r, e^{itx} \psi_s)$ for the hydrogen atom by direct integration, other systems will require a technique which involves the wave function much less explicitly. Indeed, one of the virtues of the method described in this chapter is its limited dependence on the wave function.

Chapter 3

The Off-Diagonal Hypervirial

Method

CHAPTER 3

The Off-Diagonal Hypervirial Method3.1 Description of the Method

The area of interest which has occupied the majority of the research time on this project has been the Off-Diagonal Hypervirial Method. The foundation of this method is the sufficiency theorem (Coulson, 1965; Bradley and Hughes, 1969) which states that if the (initial) function B and its eigenvalue E_B are an exact solution of the Schrödinger equation, and a function A and a real number E_A satisfy the hypervirial relation

$$(A, [H, W] B) = (E_A - E_B) (A, WB) \quad (3.1)$$

for a diverse enough set of operators, W , then A and E_A are an eigensolution of the Hamiltonian. Bradley and Hughes have given a more precise statement about the choice of the operators, and their theorem is reproduced in Appendix 2. This requires us to satisfy a very large, usually infinite, number of hypervirial relations, so that $(H - E_A)A$ is orthogonal to the whole of the Hilbert space, or else to find a subspace invariant under the Hamiltonian. This subspace is generated by the operators W acting on B , and must contain A . Both these possibilities present considerable difficulties, and it has been suggested (Oppenheim, 1962; Coulson, 1965; Bradley and Hughes, 1969) that requiring a wave function to satisfy only a finite number of off-

diagonal hypervirials might give a useful approximation to a particular eigensolution of the Hamiltonian. The convergence of such a method has not, however, been demonstrated, and so far there exists no set of heuristic rules for the choice of the hypervirial operators to be used in such calculations. Furthermore, the possibility of the initial state being inaccurate had not been investigated, and suitable methods for numerical solution of the equations which the method produces were not highly developed at the time this work began.

The particular variant of the off-diagonal hypervirial method chosen for study was as follows: a trial wave function with $N+1$ linear parameters, together with a real number λ , is required to satisfy $N+1$ off-diagonal hypervirial relations plus a normalization condition (Coulson, 1965). λ represents the energy difference ($E_A - E_B$). The possibility of more hypervirials than parameters in the wave function was ruled out for the reasons which follow. First, because of the Hermiticity of the Hamiltonian, a variational calculation employing a similar trial wave function to that above requires approximately one half the number of matrix elements as the hypervirial method (one quarter if the basis is orthogonal). Second, each hypervirial relation may require a different type of integral to be evaluated, which could prove difficult or

near-impossible to incorporate effectively in a suitably compact computer program. Third, the solution of overdetermined systems of equations requires a much slower and more cumbersome technique than that for systems where the number of equations is exactly equal to the number of unknowns. Fourth, the results of Bradley and Hughes, 1967, for the Kronig-Penney model suggest that overdetermined systems yield unsatisfactory solutions. As is described later in Section 4.2, the method was slightly generalized to take account of one non-linear parameter in the trial wave function. In the rest of this chapter, problems will be examined which have an explicit, rather than numerical, solution.

3.2 An Anomaly

Applying the off-diagonal hypervirial method to the linear harmonic oscillator, we may reasonably expect to find the correct closed form solutions. We employ

$$H = \frac{1}{2}(\hat{p}^2 + x^2) \quad (3.2)$$

$$\text{with } B = \exp(-\frac{1}{2}x^2) \text{ , } E_B = \frac{1}{2} \quad (3.3)$$

$$\text{and } W_m = x^{2m+p} \quad (m=0,1,2, \dots N) \quad (3.4)$$

with p a parity indicator of the state to be determined, 1 for odd, 0 for even parity. The wave function for the final state A is taken in the linear trial form

$$A = \exp(-k x^2) \sum_{j=0}^N a_j x^{2j+p} \quad (3.5)$$

At this stage the value for k is not necessarily taken to be $\frac{1}{2}$, so one additional W (at least) would be required to fix it. The $N+1$ linear coefficients and the energy difference $(E_A - E_B)$ are determined by the satisfaction of $N+1$ hypervirial relations of the form (3.1) plus a suitable normalization. It should be noted here that the usual Euclidean norm is not required and, particularly in calculations, other norms may be more appropriate. The result of a calculation will be some approximation to the stationary state of the harmonic oscillator indexed by an integer less than or equal to $2N+p$, and of parity given by p , and for $k=\frac{1}{2}$ we expect a set of correct wave functions and energy differences. So far this method has yielded closed form solutions only for the state indexed $2N+p$, though computer calculations have been performed for lower states.

However, a curious anomaly was observed when k was held fixed at some value other than $\frac{1}{2}$ and the hypervirial equations then solved (see Section 4.3). In every case the energy difference, hence E_A , was correct to the precision of the calculation, though as Figure(3.1) and Table(3.1) show for the example of the second excited state, the wave functions were incorrect. Since the correct wave functions require $k=\frac{1}{2}$, we shall need an infinite number of terms in Eq.(3.5) to get them. With a finite

Figure (3.1) & Table (3.1) Hypervirial Wave Functions for the Harmonic Oscillator with Various Exponent Parameters

k	a	b	KE PE = $\langle V \rangle$ E	Overlap with $\psi_2(0.5)$
0.1	-0.1071	0.1285	0.1591 6.7045 6.8636	0.4338
0.3	-0.3452	0.5524	0.6591 2.2348 2.8939	0.9084
0.5	-0.5311	1.0623	1.250 1.250 2.500	1.0000
0.7	-0.6694	1.6065	1.8459 0.8244 2.6703	0.9588
0.9	-0.7753	2.1709	2.4265 0.5964 3.0229	0.8812
1.1	-0.8597	2.7511	2.9609 0.4587 3.4496	0.7998

$$\psi_2(k) = (a + bx^2) \exp(-kx^2)$$

number of terms (in fact 2), the second excited state approximation behaves regularly with k ; the node, maximum and expectation value $\langle x^2 \rangle$ decrease smoothly as k increases. Similar behaviour was found for other excited state approximations. The anomalous situation in which the eigenvalue is correctly determined even though the eigenfunction is grossly in error has already been noted by Coulson (1965). An explanation follows.

3.3 Explanation

With the hypervirial operator $W=x^{2m+p}$ and the harmonic oscillator Hamiltonian we obtain

$$[H, W_m] = -(2m+p) x^{2m+p-1} \frac{d}{dx} - \frac{1}{2}(2m+p)(2m+p-1) x^{2(m-1)+p} \quad (3.6)$$

If we now define

$$I_s(k+\nu) = \int_{-\infty}^{\infty} x^s \exp(-(k+\nu)x^2) dx$$

$$= \begin{cases} 0 & ; s \text{ odd} \\ \frac{s! \sqrt{\pi}}{(\frac{1}{2}s)! 2^s (k+\nu)^{\frac{1}{2}(s+1)}} & ; s \text{ even} \end{cases} \quad (3.7)$$

$$\text{and put } \lambda = (E_A - E_B) \quad (3.8)$$

the off-diagonal hypervirial relation (3.1) takes the form

$$\sum_{j=0}^N \left[(2m+p) I_{2(m+j+p)} - \frac{1}{2}(2m+p)(2m+p-1) I_{2(m+j+p-1)} \right] a_j$$

$$= \lambda \sum_{j=0}^N I_{2(m+j+p)} a_j \quad (3.9)$$

Note that if the definitions of A and W_m had not taken account of parity, the form of the integrals (3.7) would automatically have yielded two sets of equations, one for

even and one for odd parity.

If we write the equations in determinantal form, we have as general element (row m, column j)

$$\left\{ (2m+p-\lambda) I_{2(m+j+p)} - \frac{1}{2}(2m+p)(2m+p-1) I_{2(m+j+p-1)} \right\} \quad (3.10)$$

For the harmonic oscillator states of odd parity this is

$$\det \begin{bmatrix} (1-\lambda)I_2 & (1-\lambda)I_4 & \dots & (1-\lambda)I_{N+1} \\ (3-\lambda)I_4-3I_2 & (3-\lambda)I_6-3I_4 & \dots & (3-\lambda)I_{N+3}-3I_{N+1} \\ (5-\lambda)I_6-10I_4 & (5-\lambda)I_8-10I_6 & \dots & (5-\lambda)I_{N+5}-10I_{N+3} \\ \dots & \dots & \dots & \dots \\ (N-\lambda)I_{N+1}-\frac{1}{2}N(N-1)I_{N-1} & \dots & (N-\lambda)I_{2N}-\frac{1}{2}N(N-1)I_{2N-2} \end{bmatrix} = 0 \quad (3.11)$$

In element (3.10) we note: 1) The $m=0$ row elements are monomials for either parity. 2) The coefficients of the integrals are determined by m only and are thus uniform across any given row. By elementary row operations we can reduce the determinant to

$$(p-\lambda)(p+2-\lambda)\dots(p+2N-\lambda) \det \left\{ I_{2(m+j+p)} \right\} \quad (3.12)$$

Again for the odd states this is

$$(1-\lambda)(3-\lambda)(5-\lambda)\dots(N-\lambda) \det \begin{bmatrix} I_2 & I_4 & \dots & I_{N+1} \\ I_4 & I_6 & \dots & I_{N+3} \\ \dots & \dots & \dots & \dots \\ I_{N+1} & I_{N+3} & \dots & I_{2N} \end{bmatrix} \quad (3.13)$$

Thus we get the correct values for the energy differences despite an incorrect exponential factor in the trial wave function. The precise form of the wave functions is given in the next section.

The discussion above can be generalized to other excited state trial function types. If, instead of Eq.(3.5) we were to use

$$A = f(|x|) \sum_{j=0}^N a_j x^{2j+p} \quad (3.14)$$

and the new definition

$$I_s = \int_{-\infty}^{\infty} f(|x|) \exp(-\frac{1}{2}x^2) x^s dx$$

$$= 0 ; s \text{ odd}$$

$$< \infty ; s \text{ even} \quad (3.15)$$

the whole of the above discussion is still applicable, and the result concerning the energy differences is the same. (Note that A must belong to the domain of H .)

3.4 Wave Functions for Fixed Exponent

Let us look for a closed form expression for the wave function belonging to the energy difference

$$\lambda = 2M+p \quad (3.16)$$

The last row of a determinant such as (3.10) is then null, and we may use the previous M rows plus a normalization condition to fix the $M+1$ coefficients a_j in Eq.(3.5). The theorem whereby these a_j are calculated is fairly long and tedious and will only be sketched here.

1) The matrix in (3.12) is brought into a triangular

form by row operations. It is possible to find expressions for the elements of the reduced matrix, and this was accomplished by trying several examples, guessing the expression, and inductively proving it to be true.

2) The back substitution to obtain the vector of coefficients was performed for several examples, i.e. the last row of the now triangular matrix was removed, a_M was set to 1, and the other a_j found recursively. A form for the general $a_j^{(M)}$ was then guessed and verified by induction. (The superscript indicates that we wish the wave function approximating state $2M+p$). Inspection revealed the wave functions to be expressible in terms of Hermite polynomials.

Through this somewhat cumbersome construction it is reasonably straightforward to obtain the result

$$A_n = \exp(-k x^2) H_n((k+\frac{1}{2})^{\frac{1}{2}} x) \quad (3.17)$$

where $H_n(y)$ is the Hermite polynomial in y of order $n=2M+p$.

Strict substitution of the coefficients a_j as defined by the Hermite polynomial in (3.17) into the set of M equations resulting from determinant (3.12) by removal of the last row, that is,

$$\sum_{j=0}^M a_j I_{2(m+j+p)} = 0 \quad (m=0,1,\dots,M-1) \quad (3.18)$$

shows that in order to verify Eq.(3.17) we are required to prove

$$Q(m, M) = \sum_{j=0}^M \frac{(-1)^j (2(m+M-j))!}{(m+M-j)!(2(M-j))!j!} = 0 \quad (3.19)$$

for all m and M such that $M > m \geq 0$. As this last requirement is forced by the nature of the equations in that we need M relationships (and no more) to define the ratios of $M+1$ quantities, we may define

$$Q(m, M) = 0 \quad \text{when } m \geq M \quad (3.20)$$

Then the Binomial Theorem shows that

$$Q(0, M) = 0 \quad \text{for all } M \quad (3.21)$$

Moreover, from the definition (3.19) of $Q(m, M)$

$$Q(m+1, M) = (4m+4M+2) Q(m, M) + 4 Q(m, M-1) \quad (3.22)$$

Since $Q(0, M) = 0$ for all M , this implies $Q(1, M) = 0$ for all M also. If then $Q(n, M) = 0$ for all M and all n less than or equal to 1, Eq.(3.22) shows $Q(1+1, M) = 0$ for all M also, and we have completed the induction which proves $Q(m, M) = 0$ for all m and M .

It will be appreciated that if we decide to put $k=\frac{1}{2}$ (and only then) the functions A_n of Eq.(3.17) become identical with the true eigenfunctions of the harmonic oscillator.

Prof. S.T.Epstein has pointed out that the result for the wave function is a consequence of the orthogonality properties of Hermite polynomials, since any polynomial of degree $(n-1)$, say $f_{n-1}(x)$, may be written as

$$f_{n-1}(x) = \sum_{j=0}^{n-1} c_j H_j((k+\frac{1}{2})^{\frac{1}{2}} x) \quad (3.23)$$

The set $\{H_j((k+\frac{1}{2})^{\frac{1}{2}} x)\}$ is orthogonal with respect to the weighting factor $\exp(-(k+\frac{1}{2}) x^2)$. The set of equations derived from determinant (3.12) can be written

$$\int_{-\infty}^{\infty} \exp(-(k+\frac{1}{2}) x^2) x^{2m+p} A_n'(x) dx = 0 \quad (3.24)$$

where $n=2M+p$ and $m=0,1,\dots,M-1$. Combining all these equations, and noting that polynomials of different parity are automatically orthogonal under the weighting function $\exp(-(k+\frac{1}{2}) x^2)$, it is equivalent to require $A_n'(x)$ to satisfy

$$\int_{-\infty}^{\infty} \exp(-(k+\frac{1}{2}) x^2) f_{n-1}(x) A_n'(x) dx = 0 \quad (3.25)$$

for all polynomials $f_{n-1}(x)$ of degree $n-1$ or less.

Substitution of Eq.(3.23) shows finally

$$A_n'(x) = H_n((k+\frac{1}{2})^{\frac{1}{2}} x) \quad (3.26)$$

In the preceeding paragraph,

$$A_n'(x) = \exp(+k x^2) A_n(x) \quad (3.27)$$

3.5 Expectation Values from Fixed k Functions

The form of the function (3.5) lends itself to a straightforward calculation of matrix elements of the form

$$(A_r(x), x^q A_s(x)) \quad (3.28)$$

as these appear in the double sum form

$$(A_r, x^q A_s) = \sum_{j=0}^{[\frac{r}{2}]} \sum_{l=0}^{[\frac{s}{2}]} a_j^{(r)} a_l^{(s)} I_{2(j+1)+p^{(r)}+p^{(s)}+q} \quad (3.29)$$

where the integrals I depend on the respective exponent

parameters $k^{(r)}$ and $k^{(s)}$ of functions A_r and A_s as in definition (3.7). In Eq.(3.29) we cannot immediately substitute the Hermite polynomial coefficients for $a_j^{(r)}$ or $a_1^{(s)}$ without suitable normalization factors. However, these are easily calculated for say, the approximation to state s by putting $r=s$ and $q=0$ into Eq.(3.29). It is also straightforward to calculate the expectation value of $\hat{p}^2$ and hence the kinetic energy. A sample of values has already been given in Table (3.1).

One major hope for the off-diagonal hypervirial method has been that it would yield better values for transition probabilities than variational or other techniques. To test this hypothesis, transition probabilities can be calculated using the fixed exponent wave functions as above, and for comparison similar wave functions calculated by means of the Rayleigh-Ritz variational method and the trial function (3.5). Only approximations to the second excited state will be considered here. The transition moment integral

$$T_{12} = \left| (\psi_1(\nu), x \psi_2(k)) \right| \quad (3.30)$$

is given in Table (3.2), where $\psi_1(\nu)$ is the approximate ($\nu=k$) or exact ($\nu=\frac{1}{2}$) excited state function

$$\psi_1(\nu) = N(\nu) x \exp(-\nu x^2) \quad (3.31)$$

and $\psi_2(k)$ are the hypervirial wave functions listed in Table (3.1),

Table (3.2) Transition Moment Integrals $|(\psi_1(\nu), x \psi_2(k))|$
for Variational and Hypervirial Approximations to the
Second Excited State of the Harmonic Oscillator.

k	$\nu = 0.5$		$\nu = k$	
	Variational	Hypervirial	Variational	Hypervirial
0.1	0.290	0.434	2.594	2.697
0.2	0.697	0.743	1.799	1.837
0.3	0.898	0.908	1.420	1.430
0.4	0.981	0.982	1.176	1.177
0.5	1.000	1.000	1.000	1.000
0.6	0.988	0.988	0.870	0.870
0.7	0.962	0.959	0.773	0.769
0.8	0.930	0.922	0.698	0.690
0.9	0.895	0.881	0.640	0.626
1.0	0.860	0.840	0.593	0.574
1.1	0.826	0.800	0.555	0.529

where $N(\nu)$ is a normalization constant. The results are given in Table (3.2).

These results are disappointing as the hypervirial and variational approximate wave functions give similar values for the transition moment integrals. Perhaps this should not be any surprise, as altering exponents drastically changes the wave functions.

3.6 The Initial Function Problem

It should be mentioned that the choice of a correct initial state function B in Eq.(3.3) is crucial to obtaining correct eigenvalues and, for proper values of the exponent parameters, the true eigenfunctions. We may legitimately ask what effect use of an approximate rather than exact initial state function has on the results of a calculation. Indeed, this question is central to the study, by the hypervirial method, of systems for which there are no exact representations of the eigenstates, and reappears in Section 5.6. Nonetheless, some simple examples may be appropriate here.

Consider first an initial function

$$B = \exp(-\nu x^2) \quad (3.32)$$

which is incorrect unless $\nu = \frac{1}{2}$. Presuming that the integrals needed are convergent (k, ν greater than zero is sufficient) we may carry through the analysis of Sections 3.3 and 3.4 to find energy differences given

by

$$\lambda = 2 \cup 1 \quad (l=0,1,2,3,\dots) \quad (3.33)$$

and wave functions

$$A_l(x) = \exp(-k x^2) H_l((\cup + k)^{\frac{1}{2}} x) \quad (3.34)$$

Let us look next at the problem of a "contaminated" initial wave function such as

$$B = \phi_0 + c \phi_1 \quad (3.35)$$

where ϕ_j shall be the j th eigenfunction of the harmonic oscillator Hamiltonian $2H = \hat{p}^2 + x^2$. In order to see the effect the choice of operator set has, let us take various combinations of the hypervirial operators

$$1, \hat{p}, x, \hat{p}^2, x^2, \text{ and } \hat{p}x + x\hat{p}.$$

Let the final state function be

$$A = a_0 \phi_0 + a_1 \phi_1 + a_2 \phi_2 \quad (3.36)$$

There are $\binom{6}{3} = 20$ combinations of the six operators, but not all combinations give a full set of 3 eigensolutions of the resulting hypervirial equations. The combinations are labelled in Table (3.3) with some information about the solutions they yield when $c = 0$, that is, for an exact initial function.

In order to derive the hypervirial equations, we need the commutators of the six operators with

$$H = \frac{1}{2}(\hat{p}^2 + x^2) \quad (3.37)$$

and make use of

$$[ab, c] = a [b, c] + [a, c] b \quad (3.38)$$

and

Table (3.3) Operator Sets Employed in Section 3.6

(See text for explanation.)

Label	Operators	λ values for $c = 0$	Remarks concerning wave function coefficients when $c = 0$
1	$1, \hat{p}, x$	0, 1	a_2 not defined by equations
2	$1, \hat{p}, p^2$	0, 1, 2	Correct Vectors
3	$1, \hat{p}, x^2$	0, 1, 2	" "
4	$1, \hat{p}, \hat{p}x + x\hat{p}$	0, 1, 2	" "
5	$1, x, \hat{p}^2$	0, 1, 2	" "
6	$1, x, x^2$	0, 1, 2	" "
7	$1, x, \hat{p}x + x\hat{p}$	0, 1, 2	" "
8	$1, \hat{p}^2, x^2$	0, 2	} a_1 not defined by equations
9	$1, \hat{p}^2, \hat{p}x + x\hat{p}$	0, 2	
10	$1, x^2, \hat{p}x + x\hat{p}$	0, 2	
11	$\hat{p}, x, \hat{p}^2$	1, 2	} $\lambda=1$: one arbitrary parameter in coefficient vector
12	$\hat{p}, x, x^2$	1, 2	
13	$\hat{p}, x, \hat{p}x + x\hat{p}$	1, 2	a_0 not defined by equations
14	$\hat{p}, \hat{p}^2, x^2$	0, 1, 2	Correct Vectors
15	$\hat{p}, \hat{p}^2, \hat{p}x + x\hat{p}$	0, 1, 2	" "
16	$\hat{p}, x^2, \hat{p}x + x\hat{p}$	0, 1, 2	" "
17	$x, \hat{p}^2, x^2$	0, 1, 2	" "
18	$x, \hat{p}^2, \hat{p}x + x\hat{p}$	0, 1, 2	" "
19	$x, x^2, \hat{p}x + x\hat{p}$	0, 1, 2	" "
20	$\hat{p}^2, x^2, \hat{p}x + x\hat{p}$	0, 2	a_1 not defined by equations

$$[x, \hat{p}] = i ; [F(x), \hat{p}] = i \frac{dF}{dx} ; [x, G(\hat{p})] = i \frac{dG}{dp} \quad (3.39)$$

(Newing and Cunningham, 1967). Accordingly the commutators are

$$\begin{aligned} [H, 1] &= 0 & [H, \hat{p}^2] &= i(\hat{p}x + x\hat{p}) \\ [H, \hat{p}] &= i x & [H, x^2] &= -i(\hat{p}x + x\hat{p}) \\ [H, x] &= -i \hat{p} & [H, \hat{p}x + x\hat{p}] &= 2i(x^2 - \hat{p}^2) \end{aligned} \quad (3.40)$$

Further we may use the operator equations

$$x \phi_j = (2\alpha)^{-\frac{1}{2}} ((j+1)^{\frac{1}{2}} \phi_{j+1} + j^{\frac{1}{2}} \phi_{j-1}) \quad (3.41)$$

$$\hat{p} \phi_j = i (\frac{1}{2}\alpha)^{\frac{1}{2}} ((j+1)^{\frac{1}{2}} \phi_{j+1} - j^{\frac{1}{2}} \phi_{j-1}) \quad (3.42)$$

(Newing and Cunningham, 1967) with $\alpha = 1$ to obtain

after a little algebraic manipulation:

$$\begin{aligned} \text{for } W=\hat{p}, \quad ca_0 + a_1 + 2^{\frac{1}{2}}ca_2 &= \lambda (-ca_0 + a_1 + 2^{\frac{1}{2}}ca_2) \\ \text{for } W=x, \quad -ca_0 + a_1 + 2^{\frac{1}{2}}ca_2 &= \lambda (ca_0 + a_1 + 2^{\frac{1}{2}}ca_2) \\ \text{for } W=\hat{p}^2, \quad -2^{\frac{1}{2}} \cdot 2a_2 &= \lambda (a_0 + 3ca_1 - 2^{\frac{1}{2}}a_2) \\ \text{for } W=x^2, \quad 2^{\frac{1}{2}} \cdot 2a_2 &= \lambda (a_0 + 3ca_1 + 2^{\frac{1}{2}}a_2) \\ \text{for } W=\hat{p}x + x\hat{p}, \quad 2a_2 &= \lambda a_2 \\ \text{and for } W=1 \quad 0 &= \lambda (a_0 + ca_1) \end{aligned} \quad (3.43)$$

Taking appropriate combinations of these equations, it is then possible to solve for the eigensolutions, that is,

λ and the vector of coefficients $\underline{a}$. These solutions are collected in Table (3.4). In some cases λ appears in the expressions for the coefficients a_j for reasons of conciseness of form.

There are a wide variety of solutions, but sometimes the correct λ and vector coefficients appear together

Table (3.4) : Solutions of Hypervirial Calculations Using Perturbations

λ	a_0	a_1	a_2	Set Labels	Remarks
-1	$\sqrt{2} c^2$	$-\sqrt{2} c$	1	1	Disappear when $c = 0$
	$-\sqrt{2} \cdot 3 (1+c^2)$	$-\sqrt{2} c$	1	11	
	$-\sqrt{2} \cdot 3 (1-c^2)$	$-\sqrt{2} c$	1	12	
	1	0	0	13	
0	0	$-\sqrt{2} c$	1	1	{ A = $b\delta_0 + b'\delta_1$ subject to some norm
	1	-c	0	2,3,4,14,15,16	
	1	c	0	5,6,7,17,18,19	
	b	b'	0	8,9,10	
+1	0	0	1	1	{ Not valid at $c = 0$ where a_1 may be non-zero
	0	1	$-3c\frac{1}{\sqrt{2}}$	11	
	0	1	$+3c\frac{1}{\sqrt{2}}$	12	
	0	1	0	13	
2	$\frac{-\sqrt{2} c^2}{1+3c^2}$	$\frac{-\sqrt{2} c}{1+3c^2}$	1	4	
	$\frac{-\sqrt{2} c^2}{1-3c^2}$	$\frac{-\sqrt{2} c}{1-3c^2}$	1	7	
	0	0	1	8,9,10	
	0	$-\sqrt{2} c$	1	13	
	$\frac{3 \cdot \sqrt{2} c^2}{1+9c^2}$	$\frac{-\sqrt{2} c}{1+9c^2}$	1	14,15,16	

Table (3.4) Continued

λ	a_0	a_1	a_2	Set Labels	Remarks
2	$\frac{3\sqrt{2}c^2}{(1-9c^2)}$	$\frac{-\sqrt{2}c}{(1-9c^2)}$	1	17, 18, 19	Correct limit for a_j as $c \rightarrow 1/3$
	$-3cb$	b	1	20	
$\frac{1-c^2}{1+c^2}$	$-c$	1	0	4	
$\frac{1+c^2}{1-c^2}$	$-c$	1	0	7	Correct limit for a_j as $c \rightarrow 1$
$\frac{2}{1+3c^2}$	0	$-\sqrt{2}c$	1	11	
$\frac{2}{1-3c^2}$	0	$-\sqrt{2}c$	1	12	Correct limit for a_j as $c \rightarrow (1/3)^{\frac{1}{2}}$

Table (3.4) Continued
 λ a_0 a_1 a_2 Set Labels Remarks

$\frac{1-3c^2}{1+3c^2}$	$-3c$	$ $	0	14, 15, 16	
$\frac{1+3c^2}{1-3c^2}$	$-3c$	$ $	0	17, 18, 19	Correct limit for a_j as $c \rightarrow (1/3)^{\frac{1}{2}}$
$\frac{3(1-c^2) \pm \sqrt{1-2c^2+33c^4}}{2(1-3c^2)}$	$\frac{-(2-\lambda)}{\sqrt{2}\lambda}$	$\frac{(2-\lambda)}{\sqrt{2}c\lambda}$	$ $	6	Correct limit for a_j as $c \rightarrow (1/3)^{\frac{1}{2}}$
$\frac{3(1+c^2) \pm \sqrt{1+2c^2+33c^4}}{2(1+3c^2)}$	$\frac{(2-\lambda)}{\sqrt{2}\lambda}$	$\frac{-(2-\lambda)}{\sqrt{2}c\lambda}$	$ $	2	
$\frac{(3-c^2) \pm \sqrt{1+10c^2-7c^4}}{2(1-c^2)}$	$\frac{-(2-\lambda)}{\sqrt{2}\lambda}$	$\frac{(2-\lambda)}{\sqrt{2}c\lambda}$	$ $	3	Correct limit for a_j as $c \rightarrow 1$
$\frac{(3+c^2) \pm \sqrt{1-10c^2-7c^4}}{2(1+c^2)}$	$\frac{(2-\lambda)}{\sqrt{2}\lambda}$	$\frac{-(2-\lambda)}{\sqrt{2}c\lambda}$	$ $	5	

for all values of the parameter c . For example, the operator sets 8, 9, and 10 yield state 2 of the harmonic oscillator exactly, and set 13 gives state 1 similarly. Furthermore, the Bradley and Hughes theorem by and large leads to the right qualitative ideas - sets of operators such as 1 ($\hat{p}, x$, and 1) and 20 ($\hat{p}^2, x^2$, and $\hat{p}x + x\hat{p}$) have too little scope in transforming B , and the space of vectors thus created by these operators acting on B is limited to such an extent that the resultant A function and energy difference λ only poorly approximate the truth. In many instances, there are solutions which disappear at $c = 0$, and are a result of the perturbation to the initial function. Finally, the bulk of the solutions transform as c tends to zero into the correct solutions for the exact initial function.

CHAPTER 4

Numerical Examples I.

4.1. Generalized Linear Models

In the previous chapter we discussed the general theory of the generalized linear model. In this chapter we shall illustrate the theory by means of several numerical examples. The first example is the logistic model, which is a special case of the generalized linear model. The second example is the multinomial logit model, which is another special case. The third example is the Poisson logit model, which is a more general case.

Chapter 4

Numerical Examples I.

Methods

CHAPTER 4

Numerical Examples I. Methods4.1 A Generalized Algebraic Eigenvalue Problem

In the previous chapter the off-diagonal hypervirial method was described and the exclusion of the possibility of over-determined systems of equations explained. Given the problem of N linear eigenvalue equations as well as a normalization condition, each in the form

$$(P_{km} - \lambda Q_{km}) c_m = 0 \quad (4.1)$$

or in matrix form

$$(P - \lambda Q) \underline{c} = 0 \quad (4.2)$$

how are we to find solutions λ , $\underline{c}$? If either P or Q is non-singular, we may invert to give, for example, an ordinary algebraic eigenvalue problem

$$(Q^{-1}P - \lambda) \underline{c} = 0 \quad (4.3)$$

While seemingly straightforward, this method has two disadvantages: 1) It is expensive in computing effort and time (inversion, matrix multiplication and solution of the conventional algebraic eigenvalue problem are all of order N^3 operations).

2) In the case of near-singular matrices, for example Q , the eigensolutions of the conventional eigenproblem $(Q^{-1}P - \lambda)$ are very poor approximations to those of the original problem $(P - \lambda Q)$.

In investigating the latter point, the author found

one P, Q set for which the elements were all of order unity. Q^{-1} , however, had all elements of order 10^{15} , and was in fact singular in exact arithmetic. The eigensolutions of $Q^{-1}P$ were naturally in error, though it may be pointed out that three eigenvalues of the system $(P - \lambda Q)$ were calculated correctly along with two spurious values. The origin of the matrices is discussed in Section 4.5 and the reason for the singularity is mentioned in Section 5.3. At least one other example of this behaviour was observed.

A method for solving the eigenproblem (4.2) which takes account of overdetermined sets of equations was developed by Bradley and Hughes (1967), using a least squares principle (a similar method had been used earlier in the Metallurgy Laboratory in Oxford). This method was not used in the current work as it also is expensive in computer time and requires some care in finding the eigensolution.

The method chosen was inverse iteration, and was suggested by Professor L. Fox of the Oxford University Computing Laboratory. If we have a set of eigensolutions $\{\phi_r\}$ with corresponding eigenvalues λ_r (assumed real and non-degenerate), then any vector may be written

$$\underline{x} = \sum_{r=0}^N a_r \phi_r \quad (4.4)$$

Thus, solution of

$$P \underline{y} = Q \underline{x} \quad (4.5)$$

$$\text{gives } P \underline{y} = \sum_{r=0}^N a_r Q \phi_r = \sum_{r=0}^N \frac{a_r}{\lambda_r} P \phi_r \quad (4.6)$$

$$\text{so } \underline{y} = \sum_{r=0}^N \frac{a_r}{\lambda_r} \phi_r \quad (4.7)$$

If we then replace $\underline{x}$ by $\underline{y}$ (usually after normalizing to prevent computational difficulties from overflow) we have an iteration scheme which converges to the eigensolution belonging to the smallest eigenvalue in magnitude. When we wish the solution having eigenvalue closest to some real number z , we simply define

$$P' = P - zQ \quad (4.8)$$

so that the iteration scheme gives convergence to the eigensolution with smallest

$$|\lambda'_r| = |\lambda_r - z| \quad (4.9)$$

This iterative method has been found highly satisfactory in several respects: 1) it is programmable in a highly compact form,

2) it is reasonably fast,

3) the convergence may be gauged

by having the successive norms $\|\underline{y}\|$ printed out by the computer (these converge to $\pm 1/(\lambda'_{\min})$).

The infinity norm of $\underline{y}$ has been found most useful in this respect, since it is easy to obtain as the modulus of the element of largest magnitude, and the sign of this element may also be preserved, which is needed in the determination of $\lambda = \lambda_{\min} + z$.

Some refinements of the method were attempted to try to speed convergence. Trials of exponential extrapolation and damping worked well when the system gave exponential or oscillatory convergence respectively, but caused serious errors or slowing of convergence when used in the wrong situation, which is hardly surprising. It was hoped to be able to recognize the "type" of convergence within the computer program, but testing procedures are more time consuming than allowing the iterative scheme to proceed to its natural conclusion. The most successful method found for speeding convergence is simply a good guess, z , to the eigenvalue desired. Thus one eigenvalue of the system $(P' - \lambda' Q)$ will be very near zero, and Eq.(4.7) shows the relevant ϕ_r to dominate in only a few iterations. On this tack, when the eigenvalue desired is only known to a poor degree of approximation, an attempt was made to stop the iteration and redefine z using the current approximation to the eigenvalue. This requires P to be triangularized again, which besides being time consuming - each iteration is of order N^2 , while triangularizing by Gauss elimination is of order N^3 - may introduce extra rounding errors. Again, this is more expensive of computing time than allowing the iteration to proceed, so a further variant in which sub-diagonal elements are permitted in P was tried. These elements

arise in the redefinition of z , and Eq.(4.8) yields a square matrix P' from a triangular matrix P and a square matrix Q . Hopefully, if we have at the point of redefinition of z a reasonably good approximation to the eigenvalue, the increment in z will be small, and so will the sub-diagonal elements introduced into P' in this way.

The iterative scheme

$$P' \underline{y}_{j+1} = Q \underline{x}_j ; \quad \underline{x}_{j+1} = \frac{\underline{y}_{j+1}}{\|\underline{y}_{j+1}\|} \quad (4.5a)$$

now uses these sub-diagonal elements with $\underline{y}_j$ (from the previous step) to find $\underline{y}_{j+1}$. The elements of $\underline{y}_j$ are thus assumed to be reasonable approximations to those of $\underline{y}_{j+1}$, which will not always be the case!

Thus redefinition of z becomes of order N^2 operations, though the iterations are each slower. Convergence, however, is slower than in the ordinary iteration despite various choices of the number of steps permitted before redefinition e.g. we may increment z every 5, 10 or 20 iterations. This indicates that the assumption made a few lines above about the elements of $\underline{y}_j$ is not justified. No attempt was made to prove convergence for this computational scheme. No excuse is made for this - the programs were so easily modified that it made sense to try to support or condemn the scheme first.

Attempts to reduce the number of iterations are not always worthwhile. The author has found that it is

advisable to require at least six iterations to be performed in order to ensure convergence of the vector (the convergence test is on the eigenvalue only, to save time and storage space). In the conventional eigenproblem, one iteration is sufficient to compute the eigenvector when the eigenvalue has been accurately determined, providing the initial vector is not too deficient in the required eigenvector. However, in this generalized eigenproblem, the presence of the matrix Q seems to disrupt the rapid convergence of the vector. A number of test cases were run on the computer and residuals calculated and satisfactory behaviour found for a minimum of six iterations.

Recently, on December 8, 1971, Dr. J.H. Wilkinson, F.R.S., of the National Physical Laboratory, Teddington, Middlesex, described a new algorithm developed in Wisconsin by C. Moler and G.W. Stewart, and given the name "the QZ algorithm". This determines all the eigensolutions of the system $(P - \lambda Q)$ in a very elegant manner comparable to the QR algorithm for the conventional eigenproblem (Wilkinson, 1965). As yet this algorithm is unpublished, and knowledge of it came too late for application to the calculations reported here.

As somewhat of a postscript, it should be mentioned that the eigensystem may have complex eigenvalues. These are conveniently ignored, since on physical grounds they

are not desired. If the mathematical problem derived from the physical model is inaccurate, or the model inadequate, however, complex solutions may exist, and these may well account for some of the few convergence failures encountered.

It is also worthwhile to note that the usual refinement of solution based on residuals which is commonly used for linear equations is not applicable here. This is because $(P - \lambda Q)$ is too near-singular (Wilkinson, 1963, page 121). This "refinement" was tried and failed, as the condition for applicability was not mentioned Fox and Mayers (1968) from which the idea had been drawn in connection with the linear equation refinement scheme.

4.2 Determination of a Non-linear Parameter

The determination of non-linear parameters, particularly exponents in wave functions, has been one of the most troublesome problems in wave mechanics. The computational method outlined above allows us to determine one non-linear parameter fairly easily providing we can generate an extra row to P and Q . Then, if the non-linear parameter is γ , we solve the eigenproblem

$$\sum_{m=1}^N (P_{km}(\gamma) - \lambda(\gamma) Q_{km}(\gamma)) c_m(\gamma) = 0$$

$$k = 1, 2, \dots, N \quad (4.10)$$

and calculate

$$\tilde{R}(\gamma) = \sum_{m=1}^N (P_{N+1,m}(\gamma) - \lambda(\gamma) Q_{N+1,m}(\gamma)) c_m(\gamma).$$

$$(4.11)$$

We desire γ_0 such that $\tilde{R}(\gamma_0) = 0$ (4.12)

and use inverse interpolation (Ralston, 1965; Fox and Mayers, 1968) to find γ_0 after calculating $\tilde{R}(\gamma)$ for several γ . This method is cumbersome, but fairly simple to carry through. The program for inverse interpolation was based on an Everett formula given by Abramowitz and Stegun, 1965, page 882.

4.3 Harmonic Oscillator Test Case

A simple computer program was written to test these ideas on the Oxford University KDF9 Computer (decommissioned in December, 1971). This used the ideas of Section 3.2 and Section 3.3, that is,

$$B=e^{-\frac{1}{2}x^2}; W_m=x^m, m=0,1,2,\dots; A=\sum c_j x^j e^{-kx^2} \quad (4.13)$$

The wave functions of the harmonic oscillator were calculated correctly to machine accuracy for $k=\frac{1}{2}$ (which inverse interpolation yielded to 5 significant figures). The anomaly that energy differences were calculated correct to 8 figures despite the value of k led to the analysis reported in Section 3.3. One slight difference in the calculation from the ideas of that section is that the trial functions were allowed to be of mixed symmetry in this calculation.

4.4 Hydrogen Atom Test Case

As a further test of the computer programs the non-relativistic Hydrogen atom was used. The Hamiltonian is

$$H = -\frac{1}{2} \nabla^2 - \frac{1}{r} \quad (4.14)$$

Following Bradley and Hughes (1969) we use as initial state

$$B = e^{-r} ; \quad E_B = -\frac{1}{2} \quad (4.15)$$

and choose

$$W_m = a_s(r) P_q^{|p|}(\cos \theta) e^{ip\phi} \quad (4.16)$$

where we have yet to choose the radial function $a_s(r)$. The variables, r , θ and ϕ , are the usual spherical polar coordinates; see, for instance, Newing and Cunningham, 1967. The associated Legendre polynomial $P_q^{|p|}$ is defined as in Sneddon, 1961. If we try a final state function of the form

$$A = R(r) P_l^{|m|}(\cos \theta) \quad (4.17)$$

we find that the orthogonality of the associated Legendre functions causes the hypervirial relation (1.4) to be automatically obeyed unless $l=q$ and $|p| = |m|$. Still following Bradley and Hughes, we define

$$R(r) = \sum_{t=0}^N e^{-\alpha r} c_t r^t \quad (4.18)$$

$$\text{and} \quad a_s(r) = e^{(1-\alpha)r} r^s \quad (4.19)$$

but since the object is to test computer programs we now form the matrices of section 4.1 (i.e. P and Q) rather than a determinant in λ and α . For a given α , we then have

$$\begin{aligned} P_{st} = & \frac{1}{2}(1-\alpha^2) \frac{(s+t+2)!}{(2\alpha)^{t+2}} + (\alpha(s+1)-1) \frac{(s+t+1)!}{(2\alpha)^{t+1}} + \\ & + \frac{1}{2}(1(1+1) - s(s+1)) \frac{(s+t)!}{(2\alpha)^t} \end{aligned} \quad (4.20)$$

and

$$Q_{st} = \frac{(s+t+2)!}{(2\alpha)^{t+2}} \quad (4.21)$$

(It is hoped that there will be no confusion between the two uses of the symbol P .) Let us try to calculate the $2p$ eigenstate of the Hydrogen atom, or at least some approximation thereto. We allow the parameter α to take on a set of values and apply inverse interpolation to the extra row values (Section 4.2).

α	Extra Row Value (Eq.(4.11))
0.05	-12.9973
0.25	2.9692
0.45	0.6375
0.65	- 0.8165
0.85	- 1.0574
1.05	5.3942

A six point inverse interpolation gives $\alpha_0 = 0.5040$, while a quadratic interpolation on the four central points yields the slightly better value 0.5018. If we now put in the alternative set of values for α

α	Extra Row Value
0.1	17.4164
0.3	3.3283
0.5	-0.6×10^{-4}
0.7	0.9223
0.9	-1.0697
1.1	5.3811

we obtain from the inverse interpolation formula the value $\alpha_0 = 0.49999$ which is essentially the correct

value of 0.5, and for $\alpha = 0.5$ the calculated energy difference is 0.37500039 which compares favourably with the true value of $0.375 = (-1/2 \times 2^2) - (-1/2 \times 1^2)$. For this value of α we also calculate the radial wave function

$$R(r) = (1.106 \times 10^{-4} + 1.0 r) e^{-0.5 r}$$

The same sort of analysis can be carried through for the 3p state e.g.

α	Extra Row Value
0.3	2.6672
0.5	-6.6984
0.7	-3.3871

where linear inverse interpolation gives $0.33397 = \alpha_0$ and quadratic inverse interpolation 0.33629. The true value is $1/3$. Six point inverse interpolation failed due to $\tilde{R}(0.1) = -2.97$, $\tilde{R}(0.9) = -5.22$ and $\tilde{R}(1.1) = -3.86$; in short, the 'function' $\tilde{R}(\alpha)$ has rapid fluctuations. These cause successive approximations to the root of the function to be wildly divergent. An inverse Newton formula was used for the two values above in preference to the Everett formula (the reason for preference was ease of application to a desk calculator).

4.5 Calculations with Oscillators

In order to further test the off-diagonal hypervirial method, various problems concerning harmonic, quartic and sixth power oscillators were chosen for study. Mixed power oscillators were eminently suitable for study also,

but the variety of possibilities was daunting, so while all calculation schemes and derived computer programs countenance the consideration of mixed power (2,4 or 6) potentials, no results for such will be reported here. The 4th and 6th power potentials have no reported exact wave functions, but extremely accurate approximations can be derived in terms of harmonic oscillator wave functions (McWeeny and Coulson, 1948; Heilbronner et al, 1956, 1959a and 1959b). They have a further nice property that, for a positive coefficient of the power in the potential, their eigenvalue spectrum has no continuum to cause confusion, that is, we do not have to worry that the hypervirial method will yield valid states of the system in question which are not directly of interest because they belong to the continuum (this in fact occurred by mistake in an early test calculation on the Hydrogen atom). Additionally, being one-dimensional systems, these oscillators allow plots of their wave functions to be made fairly easily.

The variety of possible hypervirial operators remained and the choice of a suitable set was decided partly on grounds of ease of computation. Various combinations of powers of the coordinate x or momentum $\hat{p}$ (strictly speaking $\frac{d}{dx}$) seemed a sensible starting point. Later on the unitary operator $\exp(itx)$ was used with some success. Admittedly, these are rather specific choices, but it is

to be hoped that the method would not be too sensitive to the choice of the set of operators.

Even restricting a hypervirial operator to be some combination of powers of x and $\frac{d}{dx}$ allows a considerable choice. In order not to require further restriction, the computer was programmed to build each hypervirial relation given the operators as character strings on punched cards, one operator per card, each operator allowed to be some combination of 40 or less characters from the set $+$, $-$, P and X , where P stands for d/dx and the other symbols have obvious meanings. Thus given $XXP+PP$, the computer forms the hypervirial relation for $W = x^2 \frac{d}{dx} + \frac{d^2}{dx^2}$.

The basis in which these calculations were carried out (by a program called OSC5) was

$$\chi_n = x^n \exp(-\frac{1}{2} \alpha x^2) \quad (4.22)$$

and we describe a wave function by a vector of coefficients, say $\underline{c}$. Then the bulk of the computation is carried out in three subroutines which manipulate this vector to effect multiplication by x (subroutine "EKS"), differentiation with respect to x (subroutine "PEA") and addition or subtraction (subroutine "VSUM").

Let us follow the calculation for a vector described by $\underline{c}$, which we allow to have a maximum possible length M . The operations of multiplication by x and differentiation with respect to x will both increase the length of $\underline{c}$, so initially we will add zero components to $\underline{c}$ to bring it to

length M . The steps are as follows:

- 1) Form $H\underline{c} = (-\frac{1}{2}(d/dx)^2 + A_2 \cdot x^2 + A_4 \cdot x^4 + A_6 \cdot x^6)\underline{c}$ and store in $\underline{d}$.
- 2) Form $W_k \underline{c}$ and store in $\underline{f}$.
- 3) Form $HW_k \underline{c}$, store in $\underline{g}$, and $W_k H \underline{c} = W_k \underline{d}$, store in $\underline{d}$ (overwritten).
- 4) Form $(HW_k - W_k H) \underline{c} = \underline{g} - \underline{d}$ and store in $\underline{g}$.
- 5) The matrices P and Q of Section 4.1 are then given

$$\text{by } P_{kj} = \sum_{l=0}^M g_l \int_{-\infty}^{\infty} \exp(-\alpha x^2) x^{l+j} dx \quad (4.23)$$

$$Q_{kj} = \sum_{l=0}^M f_l \int_{-\infty}^{\infty} \exp(-\alpha x^2) x^{l+j} dx \quad (4.24)$$

The parameter α may be allowed to be an average of the exponent parameters for the initial and final states if these are different, as in Sections 3.2 and 3.3.

This procedure is fairly flexible, though it is quite expensive of computer time, since all commutators have to be explicitly worked out from the definition, whereas in many cases it is possible to derive formulae for the commutators $(HW_k - W_k H)$. Nevertheless, we do not have to restrict the form of the hypervirial operators to some specific form or another. One possible inadequacy is that constants cannot be introduced into the operator which would make it possible, for instance, to input the Hamiltonian as a character string. Also, in order to keep the program reasonably short, there are fewer checks to prohibit errors than would be ideal, though some quite

important tests are performed to prevent array bounds from being exceeded in the operations x and (d/dx) . The program described can, in a version using vectors with maximum length 20 (c input usually restricted for this version to length 7), be run by the FORTRAN fast turn-around service FLAIR of the Oxford University Computing Service, which implies that the program plus storage occupies less than 10,000 words of memory in the computer. In the trial and error testing of operator sets, it is naturally desirable to use this facility.

4.6 Some Transformations Between Series Representations of Functions

In many branches of applied mathematics, and in particular wave mechanics it is useful to be able to transform readily between the various bases in which functions may be represented. These bases are simply sets of functions which are complete for the space we are considering (Denner and Krzywicki, 1967). Computationally, for instance, it may be more useful to know a wave function in terms of powers of a coordinate times an exponential factor rather than as a series of orthogonal polynomials multiplied by the same weighting. On the other hand, the orthogonality properties of the latter as a basis are sometimes attractive. For example, the norm of an eigenfunction expressed as a series of normalized

harmonic oscillator eigenfunctions is simply the Euclidean, or square, norm of the vector of coefficients.

Let us suppose then that we have a function $f(x)$ which may be expressed in the following ways

$$f(x) = \sum_{i=0}^N a_i x^i = \sum_{i=0}^N c_i H_i(x) = \sum_{i=0}^N d_i L_i^{(\phi)}(x) \quad (4.25)$$

where N is finite, though it may be large. The $\{H_i\}$ are Hermite polynomials and are defined

$$H_i(x) = \sum_{m=0}^{\left[\frac{i}{2}\right]} \frac{(-1)^m i! 2^{i-2m}}{m! (i-2m)!} x^{i-2m} \quad (4.26)$$

(Abramowitz and Stegun, 1965; Erdelyi et al, 1954;

Gradshteyn and Ryzhik, 1965). $\left[\frac{i}{2}\right]$ is the largest integer

less than or equal to $\frac{1}{2}i$. Likewise, the definition of

the generalized Laguerre polynomial is

$$L_i^{(\phi)}(x) = L_i^{(\phi)}(x) = \sum_{m=0}^i \frac{(-1)^m}{m!} \binom{i+\phi}{i-m} x^m \quad (4.27)$$

where

$$\binom{a}{b} = \frac{\Gamma(a+1)}{\Gamma(a-b+1) \Gamma(b+1)} = \frac{a!}{(a-b)! b!} \quad (4.28)$$

The literature contains two major definitions of the generalized (or associated) Laguerre polynomial (see for instance, Sneddon, 1961, page 166, or Messiah, 1961, page 483). The definition used here is that of the sources quoted above in connection with the definition of the Hermite polynomial. Now we may note that $f(x)$ is represented by the vectors $\underline{a}$, $\underline{c}$ or $\underline{d}$, depending on the basis we happen to choose (providing the basis is acceptable, of course, in the range of x on which f is to be defined).

Thus there exist transformations R and S such that

$$\underline{a} = R \underline{c} = S \underline{d} \quad (4.29)$$

From Eqs. (4.26), (4.27) and (4.28) it is easy to see

that R and S are upper triangular matrices and that

$$R_{ij} = \frac{(-1)^{\frac{1}{2}(j-i)} j! 2^i}{(\frac{1}{2}(j-i))! i!} \quad \text{for } (j-i) \text{ non-negative and EVEN}$$

$$= 0 \quad \text{otherwise} \quad (4.30)$$

and

$$S_{ij} = \frac{(-1)^i}{i!} \binom{j+i}{j-i} \quad \text{for } (j-i) \text{ non-negative}$$

$$= 0 \quad \text{otherwise} \quad (4.31)$$

Note that the rows of these matrices correspond to the powers of x in the ordinary polynomial, while the columns each belong to one order of the orthogonal polynomial. These matrices have the attractive property that their inverses are derivable from the matrix elements themselves.

Before giving the proof it is useful to recall two lemmas, which are often overlooked in books on linear algebra because of their simplicity (see however Fox, 1964; Mostow and Sampson, 1969).

Lemma 1: If T is an N by N triangular matrix, then T^{-1} is triangular and of the same kind (upper or lower), and

$$T_{ii}^{-1} = \frac{1}{T_{ii}} \quad (4.32)$$

For a proof see Mostow and Sampson, 1969.

Lemma 2: If T and U are upper triangular, $C=TU$ has

elements

$$C_{ij} = \sum_{k=i}^j T_{ik} U_{kj} .$$

The proof of this relies solely on $T_{ik}=0$ for i greater

than k , and $U_{kj}=0$ for k greater than j .

Theorem 1:

$$R_{ij}^{-1} = \frac{|R_{ij}|}{R_{ii} R_{jj}} \quad (4.33)$$

Proof: If the hypothesis is true, then

$$q = \sum_{k=i}^j R_{ik}^{-1} R_{kj}$$

will simply give the Kronecker delta. There are two cases:

($j-i$) EVEN

$$q = \sum_{m=0}^{\frac{1}{2}(j-i)} R_{i,i+2m}^{-1} R_{i+2m,j}$$

$$= \sum_{m=0}^n \frac{R_{i,i+2m} R_{i+2m,j}}{R_{ij} R_{i+2m,i+2m}} \quad (4.34)$$

where $n = \frac{1}{2}(j-i)$ and is an integer. Thus

$$q = \sum_{m=0}^n \frac{(i+2m)! 2^i (-1)^{n-m} j! 2^{i+2m}}{m! i! (n-m)! (i+2m)! 2^i 2^{i+2m}} = \frac{j!}{n! i!} (1-1)^n = 0 \quad (4.35)$$

by the Binomial theorem, providing $i \neq j$. If $i = j$, then

Eq.(4.34) shows $q = 1$. (This is also true from Eq.(4.35)

with $n=0$.)

($j-i$) ODD As $R_{kj}=0$ for $(k-j)$ ODD,

$$q = \sum_{k=i+1}^j (2) \frac{|R_{i,k}|^{kj} R_{k,j}}{R_{ii} R_{kk}} = \sum_{m=0}^n \frac{|R_{i,i+2m+1}| R_{i+2m+1,j}}{R_{ii} R_{i+2m+1,i+2m+1}}$$

$$= 0 \quad (4.36)$$

since $R_{i,i+2m+1} = 0$ from Eq.(4.30). But if $(j-i)$ is ODD, and hence non-zero, we require $q = 0$ for a correct inverse, and the proof is complete.

Theorem 2:

$$S_{ij}^{-1} = \frac{(-1)^{j-i} S_{ij}}{S_{ii} S_{jj}} \quad (4.37)$$

Proof: By construction as before in Theorem 1, let

$$q = \sum_{k=i}^j S_{ik}^{-1} S_{kj}$$

So

$$q = \sum_{k=i}^j \frac{(-1)^{k-i} (-1)^i (k+\phi)! i! k! (-1)^k (j+\phi)!}{i! (k-i)! (i+\phi)! (-1)^{i+k} k! (j-k)! (k+\phi)!}$$

$$= \sum_{k=i}^j \frac{(-1)^{k-i} (j+\phi)!}{(k-i)! (i+\phi)! (j-k)!} \quad (4.38)$$

Now let $m = k-i$, so $k = i+m$ and

$$q = \sum_{m=0}^{j-i} \frac{(-1)^m (j+\phi)!}{m! (i+\phi)! (j-i-m)!} = \frac{(j+\phi)!}{(i+\phi)! (j-i)!} (1-1)^{j-i} \quad (4.39)$$

Thus $q = 0$ unless $i = j$, when $q = 1$. These are the correct conditions for the inverse, and the proof is complete.

Theorem 1 has been used to transform between a basis of harmonic oscillator eigenfunctions

$$\phi_n = N_n H_n((\alpha)^{\frac{1}{2}} x) \exp(-\frac{1}{2} \alpha x^2) \quad (4.40)$$

where

$$N_n = \frac{(\pi)^{\frac{1}{4}}}{(2^n n! (\alpha)^{\frac{1}{2}})^{\frac{1}{2}}} \quad (4.41)$$

and the "polynomial" type basis

$$X_n = x^n \exp(-\frac{1}{2} \alpha x^2) \quad (4.42)$$

The transformation matrix elements are then

$$F_{ij} = N_j \alpha^{\frac{1}{2}i} R_{ij} = N_j \frac{2^i \alpha^{\frac{1}{2}i} (-1)^{\frac{1}{2}(j-i)} j!}{(\frac{1}{2}(j-i))! i!} \quad (4.43)$$

Thus

$$F_{ij}^{-1} = \frac{1}{N_i \alpha^{\frac{1}{2}j}} \frac{R_{ij}}{R_{ii} R_{jj}} \quad (4.43a)$$

It is then easy to show

$$F_{ij}^{-1} = \frac{|F_{ij}|}{F_{ii} F_{jj}} = \frac{N_j \alpha^{\frac{1}{2}i} |R_{ij}|}{N_i \alpha^{\frac{1}{2}i} R_{ii} N_j \alpha^{\frac{1}{2}j} R_{jj}} \quad (4.44)$$

(Continuation of Eq.(4.44))

$$F_{ij}^{-1} = \frac{1}{N_i} \frac{|R_{ij}|}{\alpha^{\frac{1}{2}j} R_{ii} R_{jj}} = \frac{j!}{N_i} \frac{(2(\alpha)^{\frac{1}{2}})^{-j}}{i! (\frac{1}{2}(j-i))!} \quad (4.44)$$

So that F has the inversion property of R. To recapitulate, F_{ij}^{-1} gives the contribution to ϕ_i due to X_j , hence we need $\frac{1}{N_i}$ to take account of the normalization of the ϕ_i , and an $\alpha^{-\frac{1}{2}j}$ to correct for the scaling factor in the x^j term of $H_i((\alpha)^{\frac{1}{2}}x)$. The discussion above shows that it is easy to transform between the bases, and a computer sub-program (called "HP") has been written which rather than compute explicit matrices F and/or F^{-1} , performs the transformation directly using the expressions above to effect

$$a'_i = (2) \sum_{k=1}^M F_{ik} c'_k ; \quad c'_j = (2) \sum_{k=1}^M F_{jk}^{-1} a'_k \quad (4.45)$$

where the a'_i and c'_j are the coefficients of X_i and ϕ_j respectively in expansions similar to (4.25). This subroutine has been in use for approximately one year at time of writing. It has two deficiencies which should be mentioned: 1) It calculates the factorials and powers of 2 it requires on each "call" because the storage was needed for other purposes elsewhere in the "calling" program. This trading of computing time for memory space is fairly common in programs, and if storage were not at a premium (in order to keep the total program less than 10,000 words), the calculation of factorials and powers would only need

to be performed once during the entire calculation.

2) A practical limit of about 50 is set on M in Eqs.(4.45) because $50!$ is approaching overflow level for the Oxford University ICL 1906A computer (in fact, the calculation of factorials fails at $57!$, roughly 10^{76}).

The author has made considerable attempts to find in the literature the theorems of the earlier part of this section. The failure of this search led to the seeking of advice from a number of people whose work brought them into contact with orthogonal polynomials. While none reported they had come across the theorems previously, Dr. D. Ross remembered a discussion with the author, and when reading Szegő (1959) noted Exercises 75 and 77, which ask for proofs of the relationships

$$\frac{x^n}{n!} = \sum_{\nu=0}^n \binom{n+\alpha}{n-\nu} (-1)^\nu L_{\nu}^{(\alpha)}(x) \quad (4.46)$$

and

$$\frac{(2x)^n}{n!} = \sum_{\nu=0}^{[n/2]} \frac{1}{\nu!} \frac{H_{n-2\nu}(x)}{(n-2\nu)!} \quad (4.47)$$

Thus the theorems are not new, which might be expected from their simplicity. However, in view of their usefulness, particularly in relation to the material given in this volume, it seems appropriate to include them here.

4.7 The Hypervirial Operator $W = \exp(itx)$

In Chapter 2 the hypervirial operator $W = \exp(itx)$ was found to be quite useful in the calculation of transition moment integrals. Since we now have several

formulae for the commutators of this operator with the Hamiltonian $H = \frac{1}{2}p^2 + V(x)$, i.e. Eqs.(2.10) and (2.11), as well as the appropriate (correct) integrals for oscillator calculations in Appendix 1, it is tempting to see if this operator is equally useful when applied to the off-diagonal hypervirial method. Dr. C.J.Bradley suggested periodic functions might be good choices for the hypervirial operators in that they might act as a "vernier" to select a good approximation to the correct wave function. A slightly different reason for this choice is that e^{itx} encompasses all powers of x in its Taylor expansion, so hopefully may be hoped to yield better results than a specific set of individual powers of x . In a basis of harmonic oscillator eigenfunctions, as it turns out, the form in which the integrals appear (call them I_{rs}), requiring us to always know which of r and s is larger, makes programs cumbersome. Furthermore, the matrices P and Q (Section 4.1) generated will contain a large number of zero elements, which while mainly desirable in pencil and paper calculations, cannot be employed to advantage within the algorithms described in Section 4.1, and may occasionally cause difficulty. Consequently, calculations were performed with the basis

$$X_n = x^n \exp(-\frac{1}{2} \alpha x^2) \quad (4.48)$$

(also (4.22) and (4.42)). Before proceeding with this, however, let us look at the form of the hypervirial

relations for the single commutator hypervirial applied to harmonic oscillator wave functions with exponent parameter $\alpha = 1$. As initial state let us take $B = \phi_0$, the ground state with energy $\frac{1}{2}$. Using Eq.(2.10) together with

$$\frac{d\phi_n}{dx} = \left(\frac{1}{2}\right)^{\frac{1}{2}} \left(n^{\frac{1}{2}} \phi_{n-1} - (n+1)^{\frac{1}{2}} \phi_{n+1} \right) \quad (4.49)$$

(Newing and Cunningham, 1967) for the function ϕ_n belonging to energy $(n+\frac{1}{2})$, we easily arrive at

$$(\phi_r, [H, C \exp(itx)] \phi_0) = C \exp(-\frac{1}{4}t^2) (2^r r!)^{-\frac{1}{2}} (it)^r \cdot r \quad (4.51)$$

and

$$(\phi_r, C \exp(itx) \phi_0) = C \exp(-\frac{1}{4}t^2) (2^r r!)^{-\frac{1}{2}} (it)^r \quad (4.52)$$

where C is a constant yet to be determined. This uses normalized ϕ_i , so

$$(\phi_r, \exp(itx) \phi_s) = (2^s s!)^{\frac{1}{2}} (2^r r!)^{-\frac{1}{2}} \exp(-\frac{1}{4}t^2) (it)^{r-s} L_s^{r-s}(\frac{1}{2}t^2) \quad (r \geq s) \quad (4.53)$$

where the definition of the generalized Laguerre polynomial is that given by Eq.(4.27) (also Eq.(44.5) of Sneddon, 1961). For r less than s in Eq.(4.53), r and s should be interchanged on the right hand side.

From Eqs.(4.51) and (4.52) we are able to deduce the correct harmonic oscillator eigensolutions using ϕ_0 as initial state and various t values t_j , $j=1,2,\dots$. Let us set $C = \exp(+\frac{1}{4}t^2)$ to simplify the equations and then allow the final state wave function to take the form

$$A = \sum_{n=0}^{N-1} a_n \phi_n \quad (4.54)$$

Thus we have a set of secular equations

$$\sum_{r=0}^{N-1} (r - \lambda) (2^r r!)^{-\frac{1}{2}} (it_j)^r a_r = 0 \quad (j=1,2,\dots,N) \quad (4.55)$$

(only N "t" values) where λ is the energy difference to be determined. This set of equations yields a secular determinant which, omitting some non-zero multiplicative factors, can immediately be written

$$(0 - \lambda)(1 - \lambda)(2 - \lambda) \dots (N-1 - \lambda) \det(t_j^{r-1}) = 0 \quad (4.56)$$

where now both r and j take values $1,2,\dots,N$ in the determinant. Thus $\lambda = 0,1,\dots,(N-1)$ providing the determinant

$$\begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 & \dots & t_1^N \\ 1 & t_2 & t_2^2 & t_2^3 & \dots & t_2^N \\ & & \dots & \dots & \dots & \dots \\ 1 & t_N & t_N^2 & t_N^3 & \dots & t_N^N \end{bmatrix} \quad (4.57)$$

is non-zero. But this has the value

$$\prod_{i>j} (t_k - t_j) \equiv \prod_{j=0}^{n-1} \left(\prod_{i=j+1}^N (t_k - t_j) \right) \quad (4.58)$$

(see, for instance, Isaacson and Keller, 1966; Finkbeiner, 1966) because it is a Vandermonde determinant. Thus it is non-zero as long as all the t_j are different.

Of course, returning to the secular equations (4.55) we immediately have the solution for $\lambda = k$

$$a_r = 1, \quad r = k; \quad a_r = 0, \quad r \neq k \quad (4.59)$$

which is as we expect. It is interesting, perhaps, that

it does not matter what we choose as values for the t_j , except that they must all be different.

It should be noted that even for Hamiltonians with fourth and sixth power terms, Eq.(2.10) still holds. Thus in such cases it is the form of the initial wave function which carries information about the potential.

4.8 The Computer Calculation

In the basis defined by Eq.(4.48), we wish to evaluate the matrix elements $(X_r, C \bar{e}^{itx} X_s)$, $(X_r, [H, C \bar{e}^{itx}] X_s)$ and $(X_r, [H, [H, C \bar{e}^{itx}]] X_s)$ in order to set up hypervirial relations, one for each value of t chosen. The minus sign on t is purely for convenience. As a simplification, we shall first require that the wave functions we are using be of even or odd parity and write

$$A = \sum_{k=0}^M a_k X_{2k+p} \quad (4.60)$$

$$B = \sum_{m=0}^N b_m X_{2m+q} \quad (4.61)$$

where p and q are 0 for even parity and 1 for odd parity.

If the Hamiltonian is then written

$$H = \frac{1}{2}\hat{p}^2 + \frac{\delta}{2} x^2 + \frac{\beta}{4} x^4 + \frac{\gamma}{6} x^6 \quad (4.62)$$

it is straightforward to apply the formulae

$$[H, e^{itx}] = e^{itx}(t\hat{p} + \frac{1}{2}t^2) \quad (4.63)$$

and

$$[H, [H, e^{itx}]] = e^{itx} (t^2\hat{p}^2 + t^3\hat{p} + \frac{1}{4}t^4 + it V'(x)) \quad (4.64)$$

to an initial wave function B . The only integral then

required is

$$\int_{-\infty}^{\infty} x^n \exp(-(x-b)^2) dx = (2i)^{-n} (\pi)^{\frac{1}{2}} H_n(ib) \quad (4.65)$$

(Gradshteyn and Ryzhik, 1965, Formula 3.462.4, page 338)

where $H_n(y)$ is the n th order Hermite polynomial in y .

After some algebra, and substitution of

$$C = (\pi)^{-\frac{1}{2}} (-4\alpha)^{\frac{1}{2}(p+q)} (\alpha)^{\frac{1}{2}} \exp\left(\frac{t^2}{4\alpha}\right) \quad (4.66)$$

we obtain

$$(X_{2r+p}, C e^{-itx} X_{2m+q}) = (-4\alpha)^{-(m+r)} H_1(u) \quad (4.67)$$

$$\begin{aligned} (X_{2r+p}, [H, C e^{-itx}] X_{2m+q}) &= (-4\alpha)^{-(m+r)} \\ &\cdot t\left(\frac{1}{2}t H_1(u) + 2(\alpha)^{\frac{1}{2}}(2m+q) H_{1-1}(u) + \frac{1}{2}(\alpha)^{\frac{1}{2}} H_{1+1}(u)\right) \end{aligned} \quad (4.68)$$

$$\begin{aligned} (X_{2r+p}, [H, [H, C e^{-itx}]] X_{2m+q}) &= \\ &= (-4\alpha)^{-(m+r)} (4\alpha t^2(2m+q)(2m+q-1) H_{1-2}(u) + \frac{1}{4}\alpha t^2 H_{1+2}(u) - \\ &\quad - 2(\alpha)^{\frac{1}{2}} t^3(2m+q) H_{1-1}(u) - \frac{1}{2}(\alpha)^{\frac{1}{2}} t^3 H_{1+1}(u) + \\ &\quad + (\alpha t^2(4m+2q+1) + \frac{1}{4}t^4) H_1(u) - \frac{1}{2}(\alpha)^{-\frac{1}{2}} (\delta H_{1+1}(u) + \\ &\quad + (-4\alpha)^{-1} \beta H_{1+3}(u) + (-4\alpha)^{-2} \gamma H_{1+5}(u)) \end{aligned} \quad (4.69)$$

where $l = 2(m+r)+p+q$ and $u = \frac{1}{2}(\alpha)^{-\frac{1}{2}}t$. These are then elementary, though somewhat tedious, to program for computer calculation. There is some saving of effort in that we may perform two single commutator calculations and one double commutator calculation using in all only three different matrices. These calculations are described in Section 5.2.

4.9 The Computer Program ANOSC

In order to generate data both for input to and comparison with calculations using the hypervirial method, some program to compute oscillator wave functions by means of the Rayleigh-Ritz method was needed. At the time ANOSC was written, the author had not encountered in the literature sixth power oscillator wave functions calculated variationally, and in consequence the matrix element $(\phi_r, x^6 \phi_s)$ was derived for the harmonic oscillator eigenfunctions ϕ_j of the Hamiltonian $H_0 = \frac{1}{2}(p^2 + \alpha_0^2 x^2)$ which were to be used as basis. Later an extensive literature search revealed some earlier work by Heilbronner and coworkers (1956, 1959a, 1959b) in which these matrix elements were published and an application to the sixth power oscillator described. Some of these matrix elements are listed in Appendix 3.

The next stage in this problem is the solution of the symmetric matrix eigenvalue problem. At the time this stage was begun, the Oxford ICL 1906A Computer had just been delivered, and in the subsequent chaos, users were left without a trustworthy subroutine for obtaining eigenvalues and vectors of symmetric matrices, particularly when only a few eigensolutions were wanted. The routine which was temporarily offered was unable to handle close or coincident eigenvalues, and the author, partly for his

own interest, undertook the coding of a Householder/ Sturm sequence algorithm for the solution of the eigenproblem for symmetric real matrices (Wilkinson, 1965)

The subroutine, called JNH, has proved faster than expected. Certain minor inadequacies still exist, such as failure if a diagonal matrix is input (i.e. no solution necessary, but the computer does not recognize this), and it is hoped to cure these in the near future with a thorough overhaul of the program. Present users of the program include Dr.R.F.Weaver and Dr.M.W.Thomas, who use it in conjunction with a CNDO semi-empirical molecular orbital program, Mr. James Port, Mr.Roger New and Miss M. Black, who are performing theoretical studies of biologically active molecules, Mr. Brian Hills, who is developing a non-empirical molecular orbital program, Mr. Alistair Todd, who is investigating the Nesbet Molecular Orbital method, and Mrs. Elizabeth Hewat, who is performing electron microscopy studies of metals.

Tests have been carried out to ensure the accuracy of computed eigenvalues and vectors. Recently, a further series of examples in which the answers were stored in advance in the computer memory and compared to those found by JNH, the comparison being given in terms of binary bits, were carried out. The performance was at least as good as other matrix eigenvalue programs in use at Oxford. The program to carry out these tests was supplied by Mr.F.Ris.

Chapter 5

Numerical Examples II.

Results

CHAPTER 5

Numerical Examples II. Results

5.1 The Test Systems

In this chapter assorted results of the various types of calculation described in Chapter 4 will be presented. Essentially they are experiments with the off-diagonal hypervirial method. As mentioned in the Introduction (Chapter 1) they must be viewed as establishing no general rules for this method, but only as a guide for further calculation or the presentation of hypotheses to be subjected to more rigorous examination. However, if the method fails in a significant number of instances, we may make the somewhat negative conclusion that it is not a reliable procedure.

All the results given are for calculations on the harmonic, quartic or sixth power oscillators. For the latter two systems exact wave functions have not been reported, and for comparison purposes variationally calculated approximations in terms of harmonic oscillator eigenfunctions will be used. Furthermore the number of basis functions employed in these variational approximations needs to be specified in order to fix which comparison is being made. So as to have a simple way of denoting these comparison functions, the comparison function approximated by n harmonic oscillator eigenfunctions

for the quartic oscillator eigenstate j will be called

$\chi_j(n)$; that for the sixth power oscillator j will be called $\psi_j(n)$. If n is an even integer, half of the components of these approximations will have zero coefficients by symmetry. The basis is defined by the Hamiltonian

$$H_0 = \frac{1}{2}(\hat{p}^2 + a_0^2 x^2) \quad (5.1)$$

The parameter a_0 is chosen to improve the convergence of the approximations (McWeeny and Coulson, 1948). The values chosen are $a_0 = 2.0$ for the quartic oscillator and $a_0 = 4.0$ for the sixth power oscillator. A trial and error method was used to make these choices, and integers only were used, since these have the advantage of being simple to input to the computer without introducing error in converting a decimal number into its binary representation.

5.2 The Operator $W = \exp(itx)$

Let us start with some calculations with the operator $W = \exp(itx)$ which has already given service in Chapter 2. There are three possibilities

$$(I) \quad (A, [H, W] B) = \lambda (A, W B) \quad (5.2)$$

$$(II) \quad (A, [H, [H, W]] B) = \lambda^2 (A, W B) \quad (5.3)$$

$$(III) \quad (A, [H, [H, W]] B) = \lambda (A, [H, W] B) \quad (5.4)$$

which will be identified by the Roman numerals at the left.

The harmonic oscillator furnishes a check on the computer

program (called OSC6). The ground state ϕ_0 is used as initial state B, $E_B = \frac{1}{2}$. Using $t_j = j$, $j=1,2,3,\dots,10$, we accurately approximate the nineteenth state of the harmonic oscillator, obtaining (Case I)

$$\lambda = 18.999989, \quad E_A = \lambda + \frac{1}{2} = 19.499989$$

The function A, after Euclidean normalization, has component 0.99999945 in the direction of the true eigenstate ϕ_{19} . The guess z used for the inverse iteration algorithm was 19.05. The Rayleigh Ratio gives the alternative energy value $(A, HA) = 19.499992$. The results for Cases II and III are given in Table (5.1).

Table (5.1)

Harmonic Oscillator Test of $W=e^{itx}$

	Case II	Case III
z	362.9025	19.05
λ	$(360.99959)^{\frac{1}{2}}$	18.999993
	=18.999989	
E_A	19.499989	19.499993
(A, HA)	19.499987	19.499992
(A, ϕ_{19})	0.99999922	0.99999949

Ref. FLOORMOP, 3JAN72, 11.09.33

References are given to the name of the file which runs the appropriate computer program together with a date and time. This is merely to enable checking to be carried out easily.

Of course the true solution is $A = \phi_{19}$, $E_A = 19.50$.

This is apparently approximated successfully in all three cases with very small errors which may reasonably be attributed to rounding errors within the computation.

The quartic oscillator furnishes a less clear picture. Using as initial state function the comparison function

$\chi_0(10)$ (i.e. 10 component variational approximation to the ground state). Let us try to see the effect of different sets of t values and use in addition to the set

$$t_j = j \quad (j=1,2,3,4,5) \quad (a)$$

the set

$$t'_j = 2^j \quad (j=1,2,3,4,5) \quad (b)$$

and refer to these as sets a and b respectively. As an additional check on the solutions, we may calculate "residuals" for each of Cases I, II, and III, though for simplicity of the computer program these are all calculated according to the formula

$$\tilde{R}_k = \sum_{j=1}^N (\tilde{P}_{kj} - \lambda \tilde{Q}_{kj}) c_j \quad (5.5)$$

where $\tilde{R}_k$ is the k th "residual" and $\underline{c}$ is the vector (in the infinity norm) obtained by solving the eigenproblem but $\tilde{P}$ and $\tilde{Q}$ are the matrices generated for Case I (hence the quotation marks, since for Cases II and III we are not obtaining a true residual). Only the modulus of the largest "residual" in magnitude is presented and is labelled $R_{\max}$. The wave function coefficient indices refer to the eigenstates ϕ_j of H_0 of Eq.(5.1). The

results are presented in Table (5.2). z is the guess to the energy difference which is used in the calculation (Section 4.1, Eq.(4.8)). Overlaps between calculated wave functions and the variationally calculated comparison functions are also given. From these overlaps it would also be simple to evaluate the square of the length of the "difference" vector between a calculated wave function A and a comparison function, e.g. $\chi_j(10)$

$$\|A - \chi_j\|^2 = (A - \chi_j, A - \chi_j) = 2 - 2\text{Re}(A, \chi_j) \quad (5.6)$$

The results here will be misleading if the phases of A and χ_j are different. For real wave functions, the phase angle difference can only be 0 or π , and rather than adjust the phase of A , we may use

$$\|A - \chi_j\|^2 = 2 - 2|(A, \chi_j)| \quad (5.7)$$

in the case when A and χ_j are real. Because of this simple relationship, the length of the difference vector will not be included in the tables.

An examination of Table (5.2) shows that the Case IIIa has failed to give a wave function A which even partially approximates the second excited state comparison function

$\chi_2(10)$. The overlap with this comparison function is very small. The Rayleigh energy of the final state function A confirms a suspicion that, in fact, the method has converged to an approximation to the ground state; the

Table (5.2) Hypervirial Calculations with $W_j = \exp(it_j x)$
for the Quartic Oscillator, $2H = \hat{p}^2 + x^4$

$B = \chi_0(10)$, $E_B = 0.53018207$

Set (a) $t_j = j = 1, 2, 3, 4, 5$

Case	I	II	III
z	3.25	10.625	3.25
λ	3.2022784	$(10.415233)^2$ =3.2272640	3.1979122
Iterations	10	10	10
E_A	3.7324605	3.7574461	3.7280943
(A, H_A)	3.7324161	3.7399792	0.56891680
$ (A, \chi_0(10)) $	0.000361	0.004618	0.993926
$ (A, \chi_2(10)) $	0.999886	0.999567	0.110047
$R_{\max}$	1.2E-11	3.8E-2	2.7
Coefficients:			
c_0	-0.13574918	-0.13174599	0.99936847
c_2	0.98644224	0.98687509	0.02778234
c_4	0.05794877	0.04549430	-0.02202641
c_6	-0.07160871	-0.08122974	0.00199413
c_8	-0.00423950	-0.00724729	0.00129192

Ref. STUFF, 3JAN72, 20.52.14

Note: $1.2345E-2 = 1.2345 \times 10^{-2}$

Table (5.2) Continued.

$B = \chi_0(10)$, $E_B = 0.53018207$
 Set (b) $t_j = 2^j$, $j=1,2,3,4,5$

Case	I	II	III
z	3.25	10.5625	3.25
λ	2.9521095	$(8.5659277)^{\frac{1}{2}}$ =2.9267606	2.6563405
Iterations	12	16	15
E_A	3.4822916	3.4569427	3.1865226
(A, HA)	3.9271464	3.9920001	3.4045049
$ (A, \chi_0(10)) $	0.037192	0.041558	0.429858
$ (A, \chi_2(10)) $	0.984428	0.977781	0.878784
$R_{\max}$	2.0E-2	6.6E+8	1.1E+9
Coefficients:			
c_0	-0.16754210	-0.03776637	-0.54107012
c_2	0.96271777	0.96481477	0.80183462
c_4	0.21092788	0.24646173	0.25188316
c_6	0.02473081	0.02618525	0.02930277
c_8	0.00138442	0.00066406	0.00075823

Ref. NASHMOP, 3JAN72, 16.25.16

value of 0.56891680 compares with $(\chi_0, H \chi_0) = 0.53018207$ for $\chi_0(10)$. Furthermore, the overlap of the function found with the ground state comparison function is $(A, \chi_0(10)) = 0.993926$.

The operator set (b), which uses powers of 2, gives much less satisfactory results than set (a) for Cases I and II, with fairly large discrepancies between the energy E_A calculated by means of the energy difference λ , and the Rayleigh energy. This is reflected in the overlaps with the comparison functions, which are roughly 0.98 with the function $\chi_2(10)$ for Cases I and II. The "residuals" may be somewhat misleading for the operator set (b); while they are quite large this is mainly a reflection of the large magnitude of some elements in the P and Q matrices. For Case Ib these range over 12 orders of magnitude (10^0 to 10^{12}), while for Case IIb over 16 orders for the P matrix (10^1 to 10^{17}) and 12 for the Q matrix (10^0 to 10^{12}). For the operator set (a) meanwhile, the matrices have elements ranging over at most 3 orders of magnitude, and the largest element has magnitude approximately 500. Thus we would expect $R_{\max}$ to be larger for operator set (b) than for set (a). However, the difference in magnitude of the elements is a very possible cause of inaccuracy in the results for set (b), since the method of solution outlined in Section

4.1 requires some manipulations of the matrix elements and in particular subtractions which cannot be performed without some loss of precision (unless the calculating device has a double length accumulator; see Wilkinson, 1963, page 11). Not surprisingly, other logarithmically distributed sets of t_j were also unsatisfactory, and both positive and negative powers of 10 caused run-time failures of the computer program OSC6 when numbers left the acceptable floating point range (for the ICL 1906A this is 10^{-76} to 10^{+76} approximately).

The sixth power oscillator may be treated in similar fashion and the results are presented in Table (5.3). This time for Case III both operator sets (a) and (b) exhibit a failure to converge to the correct vector, though the energy differences and consequent E_A values are within 0.02% for set (a) and 1.3% for set (b) of the comparison value of $4.5393139 = (\int_2(10), H \int_2(10))$. The Rayleigh energies and overlap matrix elements show the vectors of coefficients to be closer to $\int_0(10)$. The values of $R_{\max}$ for set (b) are again much larger than those for set (a) and this arises once more from the size of the matrix elements, which for set (a) range from 10^{-1} to 10^2 in magnitude for set (a), while from 10^0 to 10^{13} for set (b). Besides affecting $R_{\max}$, this leads to less accurate approximation of the comparison

Table (5.3) Hypervirial Calculations for the Sixth Power Oscillator, $2H = \hat{p}^2 + x^6$, Using $W_j = \exp(it_j x)$

$$B = \int_0^{10} \rho_0(10) \, , \quad E_B = 0.5724441$$

Set (a) $t_j = j = 1, 2, 3, 4, 5$.

Case	I	II	III
z	4.0	16.0	4.0
λ	3.9650206	$(15.644889)^{\frac{1}{2}}$ =3.9553621	3.9659175
Iterations	10	10	10
E_A	4.5374647	4.5278062	4.5383616
(A, H_A)	4.5396731	4.5403229	0.66793657
$ (A, \rho_0(10)) $	0.000086	0.001274	0.987890
$ (A, \rho_2(10)) $	0.999989	0.999961	0.155153
$ (A, \rho_4(10)) $	0.001757	0.060721	0.000095
$R_{\max}$	1.0E-11	1.6E-2	4.4
Coefficients:			
c_0	-0.31773528	-0.31864582	0.98391109
c_2	0.85087723	0.84932324	0.17789736
c_4	0.41340006	0.41484383	0.01531775
c_6	0.05935311	0.06388086	-0.00590457
c_8	-0.02509574	-0.03063789	-0.00141039

Ref. STUFF, 3JAN72, 20.52.14

Table (5.3) Continued

$$B = \int_0^{10} \rho_0(10) , \quad E_B = 0.5724441$$

$$\text{Set (b)} \quad t_j = 2^j , \quad j = 1, 2, 3, 4, 5$$

Case	I	II	III
z	4.0	16.0	4.0
λ	3.2691029	$(17.209501)^{\frac{1}{2}}$ =4.1484335	4.0270359
Iterations	25	14	13
E_A	3.8415470	4.7208776	4.5994800
(A, HA)	4.6777151	4.5431587	0.62113220
$ (A, \int_0^{10} \rho_0(10)) $	0.164918	0.115529	0.993868
$ (A, \int_2^{10} \rho_2(10)) $	0.974084	0.991357	0.110516
$ (A, \int_4^{10} \rho_4(10)) $	0.124975	0.033301	0.002257
$R_{\max}$	1.3E-3	8.7E+7	1.6E+7
Coefficients:			
c_0	-0.45824876	-0.20391398	0.97552371
c_2	0.73391433	0.87070295	0.21691200
c_4	0.47315810	0.43534787	0.03604864
c_6	0.16355547	0.10358639	-0.00176433
c_8	0.02736435	0.00612549	-0.00025178

Ref. NASHMOP, 3JAN72, 16.25.16

energy ($\int_2(10), H \int_2(10)) = 4.5393139$ and function $\int_2(10)$ which we assume to be reasonably close to the true state 2 eigensolution of the sixth power oscillator Hamiltonian. Even without comparison, however, the fact that E_A and (A, HA) are not close for set (b) suggests inaccuracies in the results. That $E_A = \lambda + E_B$ and the Rayleigh energy (A, HA) be identical is nonetheless only a necessary condition on exact eigenfunctions, and the coalescence of the two values does not therefore imply that either the wave function or energy is exact. From this point of view, while we may be sure of some error in the results for set (b), we cannot on the same grounds say anything about the accuracy of set (a) .

Further to this it may be tempting to suggest that the arithmetic mean of E_A and (A, HA) is some approximation to the true energy of the eigenstate desired. Such hopes are immediately dashed by such counter examples as Table (5.2), Case IIa, or Table (5.3), Case II b. A great many examples have been observed where $E_A \leq E \leq (A, HA)$, and the counter examples were not found until quite late in the research. In fact, the calculations described in this section were the last to be attempted.

Table (5.4) Some Comparison Functions for the Quartic
Oscillator in Terms of Eigenfunctions of the
Harmonic Oscillator Hamiltonian $2H = \hat{p}^2 + 4x^2$

EVEN	$\chi_0(10)$	$\chi_2(10)$	$\chi_4(10)$
2E	1.0603641	7.4566286	16.309201
E	0.53018207	3.7283143	8.1546004
c_0	0.99041329	0.13604635	-0.23918153E-1
c_2	0.13719080	-0.98670445	0.66942671E-1
c_4	-0.15207629E-1	-0.62836472E-1	-0.93800540
c_6	-0.51098244E-2	0.62542117E-1	0.12222043
c_8	0.16791419E-2	-0.67667602E-2	0.62081386E-1
ODD	$\chi_1(10)$	$\chi_3(10)$	$\chi_5(10)$
2E	3.7997041	11.645543	21.333123
E	1.8998520	5.8227713	10.666561
c_1	0.99208051	0.11976154	-0.37764450
c_3	0.11897133	-0.98938236	-0.65769635E-2
c_5	-0.40053108E-1	0.20917579E-1	-0.96834056
c_7	-0.26206348E-2	0.77191618E-1	0.24466471
c_9	0.33106130E-2	-0.19586356E-1	0.31405948E-1

Ref. MOPPING, 11 JAN 72, 18.15.26

Notation: $0.12345E-2 = 0.12345 \times 10^{-2}$

Table (5.5) Some Comparison Functions for the Sixth Power Oscillator in Terms of Eigenfunctions of the Harmonic Oscillator Hamiltonian $2H = \hat{p}^2 + 16x^2$

EVEN	$\mathcal{P}_0(10)$	$\mathcal{P}_2(10)$	$\mathcal{P}_4(10)$
2E	1.1448882	9.0786278	21.857172
E	0.57244410	4.5393130	10.928586
c_0	0.94607447	0.31776336	-0.62672745E-1
c_2	0.31374493	-0.85130999	0.40313814
c_4	0.80408534E-1	-0.41273685	-0.83261746
c_6	0.28708494E-2	-0.55920589E-1	-0.34995143
c_8	-0.57800155E-2	0.28113618E-1	0.44976121E-1
ODD	$\mathcal{P}_1(10)$	$\mathcal{P}_3(10)$	$\mathcal{P}_5(10)$
2E	4.3389270	14.059755	29.437636
E	2.1694635	7.4630775	14.713818
c_1	0.91427094	0.39438955	-0.01903998E-1
c_3	0.39622568	-0.82510272	0.38319265
c_5	0.83213018E-1	-0.40213791	-0.85513654
c_7	-0.10766442E-1	-0.15395153E-1	-0.23623600
c_9	-0.85753596E-2	0.41358007E-1	0.10138652

Ref. HEX, 8NOV71, 10.25.59

5.3 Hypervirial Operators x and d/dx

For the oscillator systems we are considering, combinations of the operators x and $D = d/dx$ should be quite important as they are the basic operators used to generate all other operators in the Schrödinger coordinate representation. The computational procedure has already been described in Section 4.5. Let us begin again with the harmonic oscillator to check the program. We use $B = \phi_0$, $E_B = \frac{1}{2}$, and $W_j = x^j$, $j=0,1,2,3,4$ with a 5 element trial wave function A to obtain $\lambda = 4.0000000$ (seven zeros) (Ref. OSC3,4/8/71,11.28.12) and a vector of coefficients which approximates ϕ_4 to one part in 10^{10} . On the other hand, the operator set $W: 1, x, D, Dx, x^2$ where $D = d/dx$, fails to yield this energy difference at all (using a guess $z = 4.0000000$), but instead gives convergence to $\lambda = 2.0000000$ with the somewhat odd final function

$$A = 0.9799 \phi_2 + 0.1977 \phi_4 \quad (\text{Ref. OSC3,04/08/71,11.30.14}).$$

This has a fairly simple explanation if we look at the matrices P and Q in the harmonic oscillator basis.

$$\begin{array}{cc} P & Q \\ \left[\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -2^{3/2} & 0 & 0 \\ 0 & 0 & 2^{3/2} & 0 & 0 \end{array} \right] & \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & -2^{1/2} & 0 & 0 \\ 1 & 0 & 2^{1/2} & 0 & 0 \end{array} \right] \end{array} \quad (5.8)$$

These matrices are quite sparse, and the determinant $(P - \lambda Q)$ is singular (the power of x type basis used in the computer programs gives more full matrices which lead to less "zero-divide" situations which the computer cannot work around without extensive checking procedures and consequent extra branches in the program). Nevertheless we see that the solutions are

$$\begin{aligned}\lambda = 0 : A &= \phi_0 + a_3 \phi_3 + a_4 \phi_4 \\ \lambda = 1 : A &= \phi_1 + a_3' \phi_3 + a_4' \phi_4 \\ \lambda = 2 : A &= \phi_2 + a_3'' \phi_3 + a_4'' \phi_4\end{aligned}\tag{5.9}$$

where the values a_3, a_3' and a_3'' and a_4, a_4' and a_4'' are arbitrary real numbers. Note that here the matrix eigenvalue problem does not define two of the vector a components.

Thus the computational algorithm has worked successfully. That the results are not acceptable is due to a bad choice of the hypervirial operators, which in this case is fairly obvious. In more complicated examples, it may not be possible to make sure the functions $W_k B$ ($k=1,2,3,\dots,N$) span the subspace of basis functions in terms of which the trial function A is expressed, nor may it be possible to ensure the functions $W_k B$ are all independent.

Let us now turn our attention to the quartic oscillator, for which the comparison function $\chi_0(10)$ will be used as initial function B . The results for the operator set

$W_j = x^j$, $j=0,1,2,3,\dots,9$ follow in Table (5.6). The extra row values are calculated as in Section 4.2, and are a test of how well other off-diagonal hypervirial relations are obeyed. Similarly, the elements $(A, [H,W] A)$ give a test of the satisfaction of diagonal hypervirial theorems. The extra row values for approximations to even parity states will automatically be small providing the algorithm has reasonably effected a separation of symmetries. There exists the possibility that a wave function will not be purely a combination of only even or only odd basis functions. In these examples, the extra row values for $W = x^{11}$ are consequently small for the approximations to states 2 and 4. The fairly large extra row value of 0.23 for the approximation to state 3 is a result of quite large elements in the matrices P and Q which appear in Eq.(4.11) (of order 10^5). However, for the approximation to state 7, the discrepancy between the energy E_A calculated from the energy difference λ and the Rayleigh energy reinforces a very large extra row value in suggesting a poor approximation has been obtained. The overlap matrix elements show the function A in this case to be a combination of mainly the comparison functions $\chi_7(10)$ and $\chi_9(10)$. In the case of the approximation to state 8, the overlap element $(A, \chi_8(10))$ shows we have quite a good approximation to one comparison function.

Table (5.6) Hypervirial Calculations for the Quartic Oscillator, $2H = \hat{p}^2 + x^4$

$B = \chi_0(10)$, $E_B = 0.53018207$, $W_k = x^k$, $k=0,1,2,\dots,9$

State (Approx.)	3	2	2
z	5.822	3.728	3.100
λ	5.2920483	3.1978217	3.1978217
Iterations	14	15	50 (converged)
E_A	5.8222304	3.7280037	3.7280037
(A, HA)	5.8229611	3.7284253	3.7284253
Comparison	5.8227713	3.7283143	3.7283143
Overlaps with $\chi_j(10)$	0.999993 (with3) 0.0 (with2)	0.999997(with2) 0.0 (with3)	0.999997(with2) 0.0 (with3)
ERV (x^{11})	0.23	1.0E-5	0(10 ⁻⁶¹)
$ (A, [H, XD] A) $	1.964E-2	4.302E-3	4.302E-3
Coefficients:			
c_0		-0.13602001	-0.13602001
c_1	0.11966939	0.2E-7	
c_2	0.2E-8	0.98655839	0.98655839
c_3	-0.98919424	-0.2E-6	
c_4	-0.3E-7	0.63612407E-1	0.63612407E-1
c_5	0.19562492E-1	0.3E-8	
c_6	0.4E-8	-0.64266499E-1	-0.64266499E-1
c_7	0.80366825E-1	0.1E-7	
c_8	0.2E-8	0.49349795E-2	0.49349795E-2
c_9	-0.18232879E-1	-0.3E-8	

Ref. SLUGGUSH, 6JAN72, 12.03.14

Ref. JAYCEE,
9NOV71, 20.07.56

Table (5.6) Continued

State (Approx.)	4	7	8
z	8.100	15.24	20.27
λ	7.5902933	15.411971	18.736607
Iterations	13	10	17
E_A	8.1204753	15.942153	19.266789
(A,HA)	8.1659317	18.328538	23.481203
Comparison	8.1546004	16.239372	23.469080
Overlaps with $\chi_j(10)$	0.999605(with4)	0.914239(with7) 0.405176(with9)	0.998170(with8)
ERV (x^{11})	0(10^{-5})	11.9	2.49
$ (A,[H,XD]A) $	6.269E-2	0.6618	60.65
Coefficients:			
c_0	-0.23984874E-1		0.71143282E-3
c_1	0.2E-8	-0.15407961E-2	0.10227522E-3
c_2	0.68241500E-1		0.96914336E-2
c_3	-0.3E-9	-0.66664178E-1	0.19839127E-2
c_4	-0.98401642		0.11976388
c_5	0.4E-7	-0.13262843	0.10206806E-1
c_6	0.13944752		0.34675019
c_7	-0.1E-7	-0.62527283	0.32851870E-1
c_8	0.83867942E-1		0.92843215
c_9	-0.3E-8	0.76615728	0.46397642E-1
Ref. JAYCEE, 9NOV71, 20.07.56		Ref. JUMP, 10NOV71, 12.56.02	

However, there is only a poor separation of symmetries in the wave function, and this is reflected in the large extra row value for the operator $W = x^{11}$. In this particular case the value of the "guess" z was chosen by a mistake in reading some tabular data, and is not especially close to the resulting value of λ , and is between the comparison values $E_7 - E_0 \simeq 15.71$ and $E_8 - E_0 \simeq 22.94$; energies are obtained using $\chi_j(10)$.

All the results reported so far in this section are for a 10 basis^{function} approximation to the initial function B , i.e. $B = \chi_0(10)$. It is to be hoped that the method will yield better results if a more accurate initial function is used. Therefore, two of the calculations in Table (5.6) were repeated with $B = \chi_0(20)$ for which $E_B = 0.53018105$. The results are given in Table (5.7). The trial function A is still expanded in only 10 basis functions.

If we compare the energies E_A calculated by the hypervirial method to variationally calculated energies (10 basis functions), then the increased accuracy in the initial function B has resulted in a better value of E_A for state 4 but a marginally poorer one for state 2. The variationally calculated comparison values for approximations in 20 basis functions are, however, $E_2(20) = 3.7278490$ and $E_4(20) = 8.1309150$, to which the E_A values in Table (5.7) are closer than the corresponding values in

Table (5.7) Calculations for the Quartic Oscillator

$$2H = \hat{p}^2 + x^4$$

$$W_j = x^j, \quad j=0,1,2,\dots,9$$

$$B = \chi_0(20) \quad E_B = 0.53013105$$

Ref. JAYCEE, 9NOV71, 11.46.08

	State 2	State 4
z	3.10	8.10
λ	3.1977601(8 Itns)	7.5930664(13 Itns)
E_A	3.7279411	8.1232474
(A, HA)	3.7283703	8.1595990
Comparison (Variational)	3.7283143	8.1546004
ERV for $W=x^{11}$	0.4E-9	0.4E-5
$ (A, [H, XD]A) $	1.630E-4	5.299E-2

Coefficients:

c_0	-0.13602261	-0.23980457E-1
c_2	0.98653821	0.68145599E-1
c_4	0.63651768E-1	-0.98506191
c_6	-0.64436796E-1	0.13750235
c_8	0.60433763E-2	0.74345222E-1

Overlaps:

$$(A, \chi_2(10)) = 0.999998 \quad (A, \chi_4(10)) = 0.999803$$

$$(A, \chi_6(10)) = 0.010085$$

$$(A, \chi_8(10)) = 0.017090$$

Table (5.6) . The overlap calculations and Rayleigh energies also show that the more accurate initial function has yielded slightly better final functions A, though the degree of improvement is rather small in view of the extra effort involved in the calculation.

On the other hand, are such minor improvements of any consequence when put beside the large discrepancies between E_A and (A, HA) for states 7 and 8 in Table (5.6)? In these latter calculations, however, the truncation of the basis has become the major factor causing error. If we look at the variationally calculated energies for approximations in 20 basis functions, that is, $E_7(20) = 16.049680$ and $E_8(20) = 18.962378$, it is apparent that the hypervirial method has succeeded reasonably well in finding an approximation to the energy of the states in question.

The sixth power oscillator allows a further test of the same set of operators. Again the 10 basis function approximation to the ground state is used as initial function B . The results of the calculation are presented in Table (5.8) , and are similar in nature to those for the quartic oscillator. One general observation that may be made from perusing Tables (5.6), (5.7) and (5.8) is that states of lower energy are approximated more successfully than those of higher energy. This corresponds to experience with other methods of calculation in quantum mechanics.

Table (5.8) Hypervirial Calculations on the Sixth Power Oscillator $2H=\hat{p}^2+x^6$
 $w_j = x^j, j=0,1,2,3,\dots,9, B= \int_0^{\infty} (10), E_B=0.57244411, \text{Ref.MOPS,10JAN72,15.35.13}$

State	1	2	4
z	1.60	4.00	10.30.
λ	1.5972173	3.9663001	10.332280
Iterations	5	7	7
E_A	2.1696614	4.5387443	10.904724
(A,HA)	2.1696013	4.5406369	11.047620
Var'n'l Comparison	2.1694635	4.5393139	10.928586
ERV $w=x^{11}$	0.011	0.3E-11	0.3E-12
Coefficients			
c_0 or c_1	0.91436584	-0.31776463	0.61911509E-1
c_2 or c_3	0.39600904	0.85184905	-0.39768730
c_4 or c_5	0.83224553E-1	0.41038493	0.80160869
c_6 or c_7	-0.11843214E-1	0.62517883E-1	0.43358399
c_8 or c_9	-0.67333185E-2	-0.32473449E-1	-0.86256284E-1
Overlap with			
$\int_j (10)$	0.999998(with 1)	0.999967(with 2)	0.994873(with 4)
with j			0.083912(with 6)
$ (A, [H, XD]A) $	5.152E-3	9.281E-3	0.8549

5.4 Excited States as Starting Data

There is nothing in the off-diagonal hypervirial method which requires ~~which requires~~ that the ground state be used as the starting function B. With the harmonic oscillator, for instance, we may put $B = \phi_3$, and use the operators $W_j = x^j$, $j=0,1,2,3$, and successfully obtain wave functions, energy differences and Rayleigh energies for states 0, 1, and 2 which are exact to machine tolerances (Ref. SHORTY, 10JAN72, 11.47.48).

For the quartic oscillator ($2H = \hat{p}^2 + x^4$) let us take as initial function B the comparison function $\chi_4(10)$ for which the energy E_B is 8.1546004. The operator set $W_j = x^j$, $j=0,1,2,3,\dots,9$, then yields in an attempt to approximate the ground state χ_0 :

Table (5.9)

$z = -7.62$	Coefficients:
$\lambda = -7.5770549$	$c_0 = 0.99013927$
at 8 iterations	$c_1 = 0.164 \times 10^{-5}$
$E_A = 0.5774550$	$c_2 = 0.13914762$
$(A, HA) = 0.53021011$	$c_3 = -0.54 \times 10^{-6}$
Variational comparison	$c_4 = -0.15050013 \times 10^{-1}$
$= 0.53018207$	$c_5 = 0.27 \times 10^{-6}$
E.R.V. for $W = x^{11}$	$c_6 = -0.54013532 \times 10^{-2}$
$= 2.35 \times 10^{-5}$	$c_7 = -0.12 \times 10^{-6}$
	$c_8 = 0.25458569 \times 10^{-2}$
	$(A, \chi_0(10)) = 0.999828$
$ (A, [H, XD] A) $	$= 9.290 \times 10^{-3}$

An attempt to obtain an approximation to state 5 (using $z = +2.51$) gives a convergence failure in which successive $1/\lambda'$ values printed by the computer program have suddenly changed from monotonic progression towards a value approximately -2.2251 to an oscillatory behaviour about a value near 1.9773 (these values have ^{direct}no _^meaning by themselves, but their convergence pattern may give a clue to the source of failures). The resulting vector is of mixed symmetry (Ref. JOY, 10JAN72, 11.48.52). Furthermore, the value of z is close to the energy difference $E_5 - E_4 = 10.666561 - 8.154600 = 2.511961$ for 10 basis function approximate wave functions, so that an explanation of this behaviour cannot be found in the "guess" z being near the arithmetic ^{mean} of two acceptable energy differences. If this were the case, it would imply the inverse iteration algorithm of Section 4.1 should converge to some mixture of two approximate eigenstates. There remain the possibilities that the set of functions $W_k B$, $k=0,1,2,\dots,9$, cannot adequately approximate the wave function desired, or that the initial vector in the inverse iteration is nearly orthogonal to the solution sought. An attempt has not been made to resolve the difficulty, since the method is in any event weak in allowing failures in special cases. No calculations starting with an approximate excited state function of the sixth power oscillator were performed.

5.5 Other Operator Sets

So far only one set of operators has been considered in looking at the quartic and sixth power oscillators. Let us now consider the same problems stated in Section 5.3 for these systems but employ the set of hypervirial operators

$$W_k = D^k \quad (k = 0, 1, 2, 3, \dots, 9) \quad (5.10)$$

The results for the quartic oscillator are given for both 10 and 20 basis function approximations to the ground (and initial) state B in Table (5.10). The results show a disturbing failure of the method. In the case of $z = 3.10$ the resulting functions are of different symmetries for different length initial functions, while for the case of $z = 8.10$, $B = \chi_0(20)$, the resulting function A is a combination of mainly three odd state wave functions. In other words, use of a more accurate initial function has not resulted in even as good an approximation as $\chi_0(10)$ yields. It should be noted that the extra row value for the operator $W = x^{11}$ is only a true check of the results, that is a test of how well an off-diagonal hypervirial theorem is obeyed, when the final function A is of odd symmetry. The exceptionally low value for the case $B = \chi_0(20)$, $z = 3.10$ is merely a consequence of the symmetry of the final function.

Let us now consider two operator sets closely related

Table (5.10) Hypervirial Calculations for the Quartic Oscillator, $2H = \hat{p}^2 + x^4$ with $W_k = D^k$, $k=0,1,2,\dots,9$

B	$\chi_0(10)$	$\chi_0(20)$
E_B	0.53018207	0.53018105
z	3.10	3.10
λ	3.8766159	3.8466476
Iterations	31	18
E_A	4.4067980	4.3768287
(A, HA)	5.9610021	4.0132101
$ (A, [H, AD] A) $	0.1960	4.563
ERV (x^{11})	37.6	0.6E-5
Overlap with		
$\chi_j(10)$ for (j)	0.981156 (with 3) 0.115385 (with 1)	0.972010 (with 2)
Coefficients: c_0	0.1300E-4	-0.12704697
c_1	-0.26376285E-2	0.22E-6
c_2	-0.8412E-4	0.94845295
c_3	0.97634505	0.3E-7
c_4	0.7592E-4	0.28366571
c_5	-0.38870191E-1	-0.9E-8
c_6	0.2537E-4	-0.23138245E-1
c_7	-0.20028358	-0.2E-8
c_8	-0.3001E-4	-0.57397036E-1
c_9	0.71546647E-1	0.1E-8

Refs. JAYCEE,9NOV71,20.07.56 and JAYCEE,9NOV71,11.46.08

Table (5.10) Continued

B	$\chi_0(10)$	$\chi_0(20)$
E_B	0.53018207	0.53018105
z	8.10	8.10
λ	6.7973260	9.8902934
Iterations	50 (itn. limit) converged to 3 figures	50 (itn. limit) converged to 8 figures
E_A	7.3280081	10.420474
(A, HA)	6.1786808	9.6450383
ERV (x^{11})	42.9	81.9
$ (A, [H, XD]A) $	0.8106	24.91
Overlaps with $\chi_j(10)$	0.980367 (with 3)	0.688084 (with 3) 0.537490 (with 5) 0.478772 (with 7)
Coefficients: c_0	0.471E-5	-0.5E-8
c_1	-0.17421508	0.13714292
c_2	-0.2855E-4	0.53E-7
c_3	0.97856728	-0.71497305
c_4	0.7278E-4	0.12E-6
c_5	-0.54508158E-2	-0.61407027
c_6	-0.6153E-4	0.6E-9
c_7	0.97680060E-1	-0.25335312
c_8	0.2672E-4	-0.20E-7
c_9	-0.53622762E-1	0.16951458

Refs. JAYCEE, 9NOV71, 20.07.56

JAYCEE, 9NOV71, 11.46.08

to those discussed in the previous few pages. These are

$$W_k = x^k \quad (k = 1, 2, 3, \dots, 10) \quad (5.11)$$

and

$$W_k = D^k \quad (k = 1, 2, 3, \dots, 10) \quad (5.12)$$

The results for the quartic oscillator are presented in Tables (5.11) and (5.12) respectively. In no case are the excited states of the quartic oscillator approximated as well as in corresponding calculations in Section 5.3 .

Yet one more possible operator set is

$$W_k = x^k D \quad (k = 1, 2, 3, \dots, 10) \quad (5.13)$$

for which the results are presented in Table (5.13). If we try the same operator set and look for a root at $z = 20.27$ (in fact, this value was input by a keypunch error), the same answers arise, save only that the vector of coefficients has even symmetry components of order 10^{-6} . (This is merely a result of convergence in fewer iterations in the inverse iteration algorithm.) If the comparison of $z - \lambda = 20.27 - 16.38 = 3.89$ and the value $E_8(10) - E_0(10) - z = \frac{23.47 - 0.53 - 20.27}{= 2.67}$ is made, it is reasonable to expect convergence to an approximation to the 8 th eigenstate. However, the truncation error which arises from requiring wave functions approximated in only 10 basis functions is now important, and may be introducing spurious results. Indeed, $E_8(20) = 18.962378$, which would tend to reinforce the suspicion that convergence should be to an approximation to state 8, except that the large

Table (5.11) Hypervirial Calculations for the Quartic Oscillator, $2H = \hat{p}^2 + x^4$, using $W_k = x^k$, $k=1,2,3,\dots,10$

$$B = \chi_0(20) , E_B = 0.53018105$$

z	3.10	8.10
λ	3.3611582	6.8676341
Iterations	13	21
E_A	3.8913392	7.3978151
(A, HA)	3.7258824	7.8851006
ERV (x^{11})	0.56E-9	0.0174
Overlaps with	0.997641 (with 2)	0.962175 (with 4)
$\chi_j(10)$		0.211689 (with 2)
		0.132567 (with 0)
		0.108079 (with 6)
$ (A, [H, XD] A) $	0.1729	0.2168
Coefficients: c_0	-0.19103021	0.13699426
c_1		0.485E-5
c_2	0.97342903	-0.12044974
c_3		0.59E-6
c_4	0.97707991E-1	-0.95686974
c_5		0.12345E-3
c_6	-0.79017105E-1	0.22596081
c_7		-0.3396E-4
c_8	0.12361180E-1	0.81460869E-2
c_9		-0.661E-5

Ref. JAYCEE, 8NOV71, 17.32.23

Table (5.12) Hypervirial Calculations for the Quartic Oscillator, $2H = \hat{p}^2 + x^4$ using $W_k = D^k$, $k=1,2,3,\dots,10$
 $B = \chi_0(20)$, $E_B = 0.53018105$

z	3.10	8.10
λ	3.6266370	9.8902934
Iterations	14	50 (Itn. limit) converged to 8 figures
E_A	4.1568181	10.420474
(A, HA)	3.9324520	9.6450383
ERV (x^{11})	0.14E-5	81.86
$ (A, [H, XD] A) $	3.724	24.91
Overlaps with $\chi_j(10)$	0.977252 (with 2) 0.194019 (with 4)	0.688084 (with 3) 0.537490 (with 5) 0.478772 (with 7)
Coefficients: c_0	-0.17954121	
c_1	0.6E-7	0.13714292
c_2	0.94778535	-0.1E-7
c_3		-0.71497305
c_4	0.25920629	-0.3E-7
c_5		-0.61407027
c_6	-0.25990521E-1	
c_7		-0.25335312
c_8	-0.40055496E-1	
c_9		0.16951458

Ref. JAYCEE, 8NOV71, 17.32.23

Table (5.13) Hypervirial Calculations for the Quartic Oscillator, $2H = \hat{p}^2 + x^4$, $W_k = x^k D$, $k=1,2,3,\dots,10$

$$B = \chi_0(10), \quad E_B = 0.53018207$$

z	15.24
λ	16.377103
Iterations	50 (converged to 8 figures)
E_A	16.907285
(A, HA)	15.993188
ERV (x^{11})	33.04
ERV (x^0)	0.8E-17
$ (A, [H, XD] A) $	2.200
Overlaps with	0.176110 (with 5)
$\chi_j(10)$	0.979930 (with 7)
Variational Comparison	16.239372
Coefficients:	
c_1	-0.52980963E-1
c_3	0.15072148
c_5	0.25930599E-1
c_7	0.87650108
c_9	-0.45337557

Ref. JUMP, 10NOV71, 12.56.02

The calculation for $z = 20.27$ gives approximately the same results.

change from $E_0(10)$ underlines the inadequacy of the length 10 basis set in approximating these higher excited states.

We may also start with an approximation to an excited state wave function as B, and use operator sets (5.10) and (5.13). A "guess" $z = -7.62$ yields a fairly good approximation to the ground state for the former set, but with a "guess" $z = +2.51$ we fail to get an approximation to state 5 of the quartic oscillator, but obtain instead a poorly converged vector of coefficients which has an overlap of 0.993024 with the vector for the starting function! (See Table (5.10a)). The latter set of operators meanwhile gives at least some approximation to each of the states desired, but the accuracy of these is by no means very high (Table (5.13a)).

Turning attention now to the sixth power oscillator, calculation results for the operator sets (5.10) and (5.13) (i.e. D^k and $x^k D$ types) are given in Tables (5.14) and (5.15) respectively. In none of the examples has the energy E_A been obtained with any degree of accuracy, and in only one case does the Rayleigh energy correspond with the variational comparison to more than 2 figures. The case for $z = 10.30$ in Table (5.15) has furthermore converged to an approximation to state 3 rather than state 4.

Table (5.10a) Hypervirial Calculations for the Quartic Oscillator, $2H = \hat{p}^2 + x^4$ with $W_k = D^k$, $k=0,1,2,3,\dots,9$
 $B = \chi_4(10)$, $E_B = 8.1546004$

z	-7.62	2.51
λ	-7.5791617	1.1885696E-3
Iterations	8	50 (converged only to 4 figures at iteration limit)
E_A	0.57543873	8.1557890
(A, HA)	0.53083670	8.1458225
Var'nl E	0.53018207	0.53018207
ERV (x^{11})	1.657E-10	798.8
$ (A, [H, XD]A) $	6.580E-2	0.5098
Coefficients:		
c_0	0.98841708	-0.24309501E-1
c_1		-0.22538658E-1
c_2	0.15087180	0.59633140E-1
c_3		-0.87796275E-1
c_4	-0.14517140E-1	-0.98180897
c_5		0.74380617E-1
c_6	-0.74525106E-2	0.11992037
c_7		0.10063044E-1
c_8	0.17587475E-2	0.60592247E-1
c_9		-0.13338376E-2
Overlaps		
with $\chi_j(10)$	0.999901 (with 0)	0.993024 (with 4)
		0.086523 (with 3)

Ref. JOY, 10JAN72, 11.48.52

Table (5.13a) Hypervirial Calculations for the Quartic Oscillator, $2H = \hat{p}^2 + x^4$, using $W_k = x^k D$, $k = 1, 2, \dots, 10$
 $B = \chi_4(10)$, $E_B = 8.1546004$

z	-7.62	2.51
λ	-7.7272538	2.9398254
Iterations	9	13
E_A	0.62734657	11.094426
(A, HA)	0.68585902	10.134377
ERV (x^{11})	4.599E-10	1277.3
$ (A, [H, XD] A) $	3.721E-2	5.659
Coefficients: c_0	0.95441818	
c_1		0.11981580
c_2	0.27715766	
c_3		-0.31267741
c_4	-0.10680496	
c_5		-0.90612967
c_6	0.28794331E-1	
c_7		0.25237750
c_8	-0.57577498E-2	
c_9		-0.55781363E-1
Overlaps with		
$\chi_j(10)$ (j)	0.984759 ($j=0$)	0.117115 ($j=1$)
	0.135077 ($j=2$)	0.325327 ($j=3$)
	0.104411 ($j=4$)	0.934970 ($j=5$)

Ref. JOY, 10JAN72, 11.48.52

Table (5.14) and Table (5.15) Hypervirial Calculations
for the Sixth Power Oscillator, $2H = \hat{p}^2 + x^6$
 $B = \rho_0(10)$, $E_B = 0.57244411$, Ref. MOPS, 10JAN72, 13.35.13

Table (5.14) $W_k = D^k$, $k=0,1,2,3,\dots,9$			
z	1.60	4.00	10.30
λ	2.0630037	4.3502220	8.516797
Iterations	100 (not converged, sign oscillating)	12	23
E_A	2.6354478	4.9226661	8.8892411
(A, HA)	2.3516399	4.5623019	9.5321477
ERV (x^{11})	3.293	2.0E-9	3.831E-4
$ (A, [H, XD] A) $	2.983	1.432	
Coefficients: c_0		-0.31347464	-0.94858471E-1
c_1	0.84725110		-0.736E-5
c_2		0.82579696	0.45965885E-1
c_3	0.49641909		0.1710E-4
c_4		0.46140067	0.92141634
c_5	0.18251977		-0.130E-5
c_6		0.78676333E-1	0.37296599
c_7	0.17653468E-1		-0.126E-5
c_8		-0.26691857E-1	-0.27878983E-1
c_9	-0.45918606E-1		0.23E-6
Overlaps with $\rho_j(10)$ (j)	0.986703(j=1) 0.151019(j=3)	0.998224(j=2) 0.880703(j=4)	0.471261(j=2)
Variational comparison	2.1694635	4.5393139	10.928586

Table (5.15) $W_k = x^k D$, $k=1,2,3,\dots,10$

z	1.60	4.00	10.30
λ	1.2737652	4.5380813	7.7317613
Iterations	12	12	34
E_A	1.8462093	5.1105254	8.3042054
(A, HA)	2.2906954	4.5365371	7.7968603
ERV (x^{11})	0.3947	2.6E-8	1.835
$ (A, [H, XD] A) $	1.132		
Coefficients: c_0		-0.43566298	
c_1	0.95137420		-0.44342950
c_2		0.75417471	
c_3	0.28862440		0.69790068
c_4		0.48544536	
c_5	0.10316201		0.56098711
c_6		0.64779655E-1	
c_7	-0.30663069E-1		-0.20037678E-1
c_8		-0.39555567E-1	
c_9	0.65938716		-0.34595244E-1
Overlaps with $\int_j(10) (j)$	0.993083, j=1 0.096081, j=3	0.985586, j=2 0.096880, j=4	0.977440, j=3 0.171475, j=5
Variational Comparison	2.1694635	4.5393139	10.928586= E_4 7.4698775= E_3

5.6 Perturbations in the Initial Function

In a manner similar to that of Section 3.6, we may examine the effect of adding a small amount of "error" to the initial state wave function B. Let us consider again the harmonic oscillator and use

$$B = \phi_0 + c \phi_1 \quad (5.14)$$

where we are at liberty to choose various values for c.

As a trial function, let us put

$$A = \sum_{j=0}^3 a_j \phi_j \quad (5.15)$$

and try to extract approximations to ϕ_3 (the ϕ_j are defined as eigenfunctions of $2H = \hat{p}^2 + x^2$). The operator set used is $W_k = x^k$, $k = 0, 1, 2, 3$. The results are given in Table (5.16). The same problem using

$$B = \phi_0 + c \phi_2 \quad (5.16)$$

gives the results which appear in Table (5.17).

The initial wave functions B are not normalized, but if we talk of $c = 0.1$ as giving a roughly 10% admixture of the excited state, then the combination of different symmetries in the initial function as in Table (5.16) leads to an energy E_A roughly 2% above the true value of 3.50, with the Rayleigh energy slightly closer though below this value, while the wave function A has an almost 20% "impurity". On the other hand with the initial function of only one symmetry, as in Table (5.17), the wave function A has only about 13% "impurity" and a

Tables (5.16) and (5.17) Hypervirial Calculations for
the Harmonic Oscillator with Perturbed Initial Function
Hamiltonian : $2H = \hat{p}^2 + x^2$, $W_k = x^k$, $k=0,1,2,3$

Table (5.16): $B = \phi_0 + c \phi_1$, Ref. SNAILPACE, 7JAN72, 20.43.11

c	0.1	0.01
E_B	0.50990099	0.50009999
z	3.01	3.01
λ	3.0658389	3.0006005
Iterations	10	7
E_A	3.5757399	3.5007005
(A, HA)	3.4658916	3.4996996
a_0	0.26097161E-2	0.24510818E-5
a_1	-0.26097161E-1	-0.24510819E-3
a_2	0.18090291	0.17328302E-1
a_3	-0.98315117	-0.99984980
ERV (x^4)	0.2567	2.451E-2
$ (A, [H, XD]A) $	0.1270	1.201E-3

Table (5.17): $B = \phi_0 + c \phi_2$, Ref. SNAILPACE, 6DEC.71, 22.50.32

c	0.1	0.01
E_B	0.51980198	0.50019998
z	3.01	3.01
λ	2.4587939	2.9196971
Iterations	19	10
E_A	2.4587939	3.4198970
(A, HA)	3.4669066	3.4994334
a_0	0.13042091E-8	0.15848053E-12
a_1	0.12863395	0.16832016E-1
a_2	-0.13042335E-7	-0.15954517E-10
a_3	-0.99169212	-0.99985831
ERV (x^4)	4.032E-7	3.531E-10
$ (A, [H, XD]A) $	0.6249	8.245E-2

Rayleigh energy roughly 1% below the exact value, but the energy value E_A calculated by the hypervirial method is drastically out of tune in being over 20% too low. Similar perturbations in the final energy and wave function are found when $c = 0.01$ in Tables (5.16) and (5.17), with the magnitudes appropriately reduced.

It is harder to see what is going on when perturbing the quartic oscillator initial functions, and a crude expedient was tried. This was simply to lower the precision of the initial wave function:

1) by employing as B an approximate wave function determined variationally on a desk calculator to 4 figures, and using only 5 basis functions, two of which were chosen to have coefficient zero by symmetry;

2) by deliberately truncating the normally 8 figure input $B = \mathcal{X}_0(10)$ to 6 and 4 figures respectively.

In the first instance, consider the ground state approximation

$$B = \mathcal{X}_0(5) = 0.9904 \phi_0 + 0.1372 \phi_2 - 0.0152 \phi_4 \quad (5.17)$$

$$\text{where } \phi_j = N_j \exp(-x^2) H_j(2^{\frac{1}{2}} x) \quad (5.18)$$

The use of $W_j = x^j$, $j = 0, 1, 2, 3, 4$, with a 5 term trial function A yields, using $z = 3.50$,

$$A = -0.13563520 \phi_0 + 0.98770946 \phi_2 + 0.07767314 \phi_4$$

and an energy difference $\lambda = 3.1977892$ after 12 iterations

(Refs. OSC3,08/07/71,10,36.54 and OSTE3,31/01/72,11.29.09).

$E_B = 0.53055075$ so $E_A = \lambda + E_B = 3.7283400$ compared to a variational energy $E_2(10) = 3.7283143$. The Rayleigh energy $(A, HA) = 3.7711293$ is not quite so good. Eq.(5.17) in fact gives a good approximation to $\chi_0(10)$, the overlap being 0.999959. The overlap of function A with $\chi_2(10)$ is 0.997911.

The results for the second test are given in Table (5.18). The comparison function energies are $E_3(10) = 5.8227713$ and $E_2(10) = 3.7283143$. Contrary to expectation, the least precise input function has yielded the best approximation to the comparison functions and energies, though the improvement is only very slight and within the possible rounding errors for the not inconsiderable number of arithmetic operations employed in setting up and solving the matrix eigenvalue problem. The stability of the calculation to changes of the sort described suggests that the method is stable with respect to at least some induced instabilities, that is, rounding "errors" (in this case deliberate) do not appear to affect the answers unduly, though a much more detailed analysis would be required to confirm this.

No perturbations of the initial state were considered for the sixth power oscillator.

Table (5.18) Hypervirial Calculations for the Quartic Oscillator Using Initial Functions of Reduced Precision

$$2H = \hat{p}^2 + x^4, \quad W_k = x^k, \quad k=0,1,2,\dots,9$$

$B = \mathcal{X}_0(10)$ with precision reduced to 6 or 4 figures.

Approximation of State 3

Precision	4	6	8
E_E	0.53018210	0.53018208	0.53018207
z	5.822	5.822	5.822
λ	5.2924374	5.2920526	5.2920483
Iterations	14	14	14
E_A	5.8226195	5.8222347	5.8222304
(A, HA)	5.8229604	5.8229611	5.8229611
Selected	c_1 0.11961671	0.11966885	0.11966939
Coefficients	c_3 -0.98920053	-0.98919428	-0.98919424
	c_5 0.19620152E-1	0.19563385E-1	0.19562492E-1

Overlap
 $(A, \mathcal{X}_3(10))$ 0.99999319 0.99999308 0.99999308

ERV (x^{11}) 0.22991790 0.22991566 0.22991751
 $|(A, [H, XD]A)|$ 1.8927030E-2 1.9624261E-2 1.9639356E-2

Approximation of State 2

z	3.728	3.728	3.728
λ	3.1981481	3.1978249	3.1978217
Iterations	15	15	15
E_A	3.7283302	3.7280070	3.7280037
(A, HA)	3.7284251	3.7284253	3.7284253
Selected	c_0 -0.13593480	-0.13601928	-0.13602001
Coefficients	c_2 0.98657433	0.98655853	0.98655839
	c_4 0.63558046E-1	0.63611725E-1	0.63612407E-1

Overlap
 $(A, \mathcal{X}_2(10))$ 0.99999656 0.99999651 0.99999652

ERV (x^{11}) 1.020E-5 1.034E-5 1.034E-5
 $|(A, [H, XD]A)|$ 3.889E-3 4.294E-3 4.302E-3

Ref. SLUGGUSH, 6JAN72, 12.03.14

5.7 Miscellaneous Ideas

As the calculations described in this chapter were performed, several opportunities for minor alterations in the procedure arose. In this section these will be briefly mentioned.

In calculating the elements of the P and Q matrices for more complicated systems, analytic forms for certain integrals required may be difficult or impossible to obtain, or may be so cumbersome that formula evaluation is more time consuming and prone to error than a numerical integration. Accordingly a very crude fixed interval numerical integration using the formula

$$n \int_{-a}^a f(x) dx \simeq \sum_{i=1}^n f(x_j) \quad (5.19)$$

was tried with n, the number of points, = 11, 21 or 101 in a few calculations. For instance, for the harmonic oscillator, using $B = \phi_0$ and $W_k = D^k$ with a 21 point quadrature at equal intervals in the range $[-4, 0]$ gives $\lambda = 4.0000000$ ($= z$) in 6 iterations, as well as the function

$$A = 0.058938656 \phi_0 - 0.041709939 \phi_2 - 0.99738985 \phi_4$$

(Ref. OSC3, 15/07/71, 12.38.57) (5.20)

An 11 point quadrature at equal intervals over the range $[-4, -4]$ yields for the same initial function and operator set $\lambda = 4.0000000$ ($= z$) again, with the somewhat more accurate function (also after 6 iterations)

$$A = 0.7791 \times 10^{-4} \phi_0 - 0.128148 \times 10^{-2} \phi_2 - 0.99999918 \phi_4$$

(Ref. OSC3,15/07/71,10.10.35) (5.21)

In both cases, the energy difference has been obtained correctly, though the wave function depends on the quadrature.

For the quartic oscillator, if we use $\chi_0(5)$ (four figures) as in Eq.(5.17) as initial function B, then the operator set $W_k = x^k$, $k=0,1,2,3,4$, with a "guess" $z = 3.5$ gives the following results:

for a 101 point equal interval quadrature over the range $[-4,4]$, $\lambda = 3.1977892$ at 12 iterations, and

$$A = -0.13563520 \phi_0 + 0.98770946 \phi_2 + 0.77673126 \times 10^{-1} \phi_4$$

(Ref. OSC3,14/07/71,13.06.23) (5.23);

for a 21 point quadrature over the range $[-4,0]$

$\lambda = 3.1849335$ at 12 iterations and

$$A = -0.40378099 \times 10^{-1} \phi_0 + 0.99194777 \phi_2 + 0.12003849 \phi_4$$

(Ref. OSC3,15/07/71,12.38.57) (5.24);

for 11 point quadrature over the range $[-4,4]$

$\lambda = 3.0894458$ at 13 iterations and

$$A = -0.14101859 \phi_0 + 0.98748557 \phi_2 + 0.70611645 \times 10^{-1} \phi_4$$

(Ref. OSC3,14/07/71,13.08.37) (5.25).

These wave functions have overlaps with $\chi_2(10)$ of 0.997911, 0.991795, and 0.997978 respectively. These results suggest that numerical integration might be used

without incurring a great deal of error, but on the other hand the crudest form of the quadrature has produced the best vector but the worst energy difference of the last three examples.

Another idea, which was already mentioned in Section 5.2, arises from the observation that the energy calculated by means of the hypervirial method, E_A , and the Rayleigh energy, (A, HA) , appear to straddle the true energy (or some comparison value). As has been pointed out, this proved not to be a general phenomenon.

Since it is possible to use any state as the origin of the starting function B, it is feasible to devise an iterative scheme whereby the hypervirial method is used to calculate for instance function A in a procedure starting with B, then further to return to another approximation B from A, and so on. This was attempted and several example calculations follow. Prof. C.A. Coulson suggested this scheme might converge and so improve both the functions A and B simultaneously. For the harmonic oscillator the idea works admirably providing the initial function of the whole procedure is exact, and exact solutions are found at each step, and one calculation "up" and one "down" (i.e. from a lower to a higher energy state or vice-versa) are sufficient to yield "convergence".

In order to see if such a calculative system will

genuinely converge, we must try other examples with more complexity such as the quartic oscillator or perturbed harmonic oscillator, where it is the initial wave function rather than the Hamiltonian which is perturbed by the addition of some "impurity". Chronologically the quartic oscillator was tried first, and at each jump "up" or "down" the guess z for the energy difference at the next step was taken to be

$$z_{\text{new}} = -\lambda \quad (5.26)$$

The initial function B was taken to be $\chi_0(5)$ as defined by Eq.(5.17) and the results of the first jump with $z = 5.5$ have already been given in Section 5.6. The results are then used as input for the "down" stage of the iteration, and $z = -3.1977892$ giving $\lambda = -3.1977892$ as well, with the wave function

$$A = 0.99042623 \phi_0 + 0.13720363 \phi_2 - 0.1520040 \times 10^{-1} \phi_4 \quad (5.27)$$

This function has overlap 0.999985 with $\chi_0(10)$. In subsequent steps no new information emerges in the energy differences or the wave functions for either the upper or lower state. The new function (5.27) approximating the ground state has the same energy as that defined by Eq.(5.17) but the overlap with $\chi_0(10)$ is slightly bigger, and the value of $|(A, [H, XD] A)|$ is $0.47788435 \times 10^{-3}$ compared to $0.47789899 \times 10^{-3}$.

For the harmonic oscillator, we may choose "impurities" of like or different symmetry to the function of interest as in Section 5.6. The guess for the energy difference is calculated by the formula

$$z_{\text{new}} = -(0.6 z_{\text{old}} + 0.4 \lambda) \quad (5.28)$$

The results are given in Tables (5.19) and (5.20).

From these examples it appears that perturbations neither grow nor decrease in the hypervirial calculation, and more than one iteration is not warranted. No example was found where the procedure caused perturbations to diminish, though several cases were found in which the operator set was inadequate, and the results rapidly become nonsensical, in particular the set $1, D, x, Dx, x^2$ discussed in section 5.3.

No examples were attempted employing the sixth power oscillator.

5.8 Discussion

In this chapter various example calculations have been performed using the off-diagonal hypervirial method, and a number of failures and mere partial successes encountered. Much could be said about these shortcomings of the method, though the distillate of such discussion would at best be only some guidelines for the calculation procedures, and to be confident of his results, a user of the method in the form described in this work would have to undertake careful checking of solutions. The method is already

Table (5.19) Perturbed Harmonic Oscillator Wave Functions
 $2H = \hat{p}^2 + x^2$; $B = \phi_0 + 0.1 \phi_1$

W_k , $k=0,1,2,3,4$	x^k	D^k
Step 1 "up"		
z	3.02	3.02
λ	3.0628989	2.9430363
Iterations	10	11
Coefficients: c_0	0.25474520E-2	0.21984463E-2
c_1	-0.25474521E-1	-0.21984463E-1
c_2	0.17658434	0.15860827
c_3	-0.95957235	-0.97174457
c_4	-0.21767735	0.17340056
Step 2 "down"		
z	-3.0371596	-2.9892145
λ	-3.0628989	-2.9430363
Iterations	8	8
Coefficients: c_0	0.99503719	0.99503719
c_1	0.99503719E-1	0.99593719E-1
c_2	-0.7E-10	-0.14E-8
c_3	-0.3E-10	-0.29E-9
c_4	0.4E-10	-0.51E-9

These coefficients are a normalized representation of the initial function B given at the top of the page. Subsequent steps "up" or "down" yield no new information.

Ref. OS3A, 11/01/72, 17.56.17

OS3A, 11/01/72, 18.02.13

Table (5.20) Perturbed Harmonic Oscillator Wave Functions

$$2H = \hat{p}^2 + x^2 \quad ; \quad B = \phi_0 + 0.1 \phi_2$$

W_k , $k = 0,1,2,3,4$	x^k	D^k
Step 1 "up"		
z	3.02	3.02
λ	3.3548030	2.7516486
Iterations	19	12
Coefficients: c_0	-0.14588951E-1	0.92363069E-1
c_1	-0.53381449E-5	-0.15E-9
c_2	0.14588951	-0.92363090
c_3	0.41153962E-4	0.5E-10
c_4	-0.98919331	-0.37198793
Step 2 "down"		
z	-3.1539212	-2.9126594
λ	-3.3548029	-2.7516486
Iterations	18	10
Coefficients: c_0	0.99503719	0.99503719
c_1	0.2611E-4	-0.6E-10
c_2	0.99503720E-1	0.99503719E-1
c_3	0.3358E-5	-0.3E-10
c_4	0.13E-9	0.18E-9

These coefficients approximately represent the initial function B after normalization. Subsequent steps "up" or "down" yield no new information.

Ref. OS3A, 11/01/72, 18.05.50

OS3A, 11/01/72, 18.06.59

cumbersome, and if on top of calculation of twice as many integrals (of more diverse types) than in the similar variational problem a further checking procedure is required, it is to be discarded unless it provides a significantly better approximation to the eigenstates wanted of the quantum mechanical system under study. This, as shown in the previous pages, it often fails to do. Sometimes, by a choice of operators which in retrospect we must call bad, the results are poor or the computational scheme fails to find a solution at the point desired in the energy difference spectrum. Yet choices such as $(d/dx)^k$, $k=0,1,2,\dots$, cannot be regarded as poor from the point of view that in operating on a single wave function they generate a subspace of the total Hilbert space (providing the wave function is suitably differentiable). Formally, then, such operators should satisfy the conditions of the Bradley and Hughes theorem (Appendix 2). Indeed, if we do not attempt to make the subspace of wave functions $W_k B$, $k=0,1,2,\dots$, as large as possible, or do not ensure linear independence among this set, we will have difficulties such as those described in Section 5.3 .

Even more disheartening is the observation that the operator sets which are normally satisfactory, for instance powers of the coordinate x (starting with x^0), may have

failures in situations where less likely sets succeed, albeit not well, as in the case where for the quartic oscillator initially in a state approximated by $\chi_4(10)$, x^k type operators fail to give any approximation to $\chi_5(10)$, whereas $x^k D$ type operators achieve at least a vector of coefficients of completely the right symmetry, as described in Section 5.4, in Table (5.13a).

It is therefore the conclusion of this chapter that if these numerical trials have any general validity, the off-diagonal hypervirial method is unlikely to produce accurate wave functions and energy differences without an amount of work out of proportion to the difficulty of the problem.

Appendices

APPENDIX 1

Some Errors in Tables of Integrals

In the derivation of Eq.(2.38) it was mentioned that the result

$$\int_{-\infty}^{\infty} \psi_r(x) \exp(itx) \psi_s(x) dx \quad (A1.1)$$

where ψ_r and ψ_s are harmonic oscillator wave functions belonging to energies $(r+\frac{1}{2})$ and $(s+\frac{1}{2})$ respectively, could be obtained by direct integration. There are, however, errors in certain of the tables, and some effort was wasted trying to "correct" the method in Chapter 2, and later to find true expressions for diverse integrals from which (A1.1) may be derived. There follows a transcription of a letter to the editor of Mathematics of Computation. (Addendum: this contribution is to be published in the April, 1972 issue.)

Mathematical Institute

10th November, 1971

Dear Sir,

Reference: Table Errata

Formula 7.374.7 on page 837 of I.S.Gradshteyn and I.M. Ryzhik, Table of Integrals, Series and Products, 4th Edition, (Trans. A. Jeffrey), Academic Press, New York, 1965, is , I believe, in error.

$$7.374.7 \int_{-\infty}^{\infty} e^{-(x-y)^2} H_m(x) H_n(x) dx = 2^n \pi^{\frac{1}{2}} m! y^{n-m} L_n^{n-m}(-2y^2) \\ (m \leq n)$$

It should read

$$\int_{-\infty}^{\infty} e^{-(x-y)^2} H_m(x) H_n(x) dx = 2^n \pi^{\frac{1}{2}} m! y^{n-m} L_m^{n-m}(-2y^2)$$

In other words replace n ("N") by m ("M"). This may be shown by performing the integration. I have carried this through by two methods, one an integration by parts using

$$dv = H_m(x) dx, \quad u = e^{-(x-y)^2} H_n(x);$$

the other uses the generating function. The integration by parts was performed in the specific case $y = \frac{1}{2}it$ where $i = \sqrt{-1}$, and t is a new variable. The reason for this choice is simply the form in which the integral appeared in a research problem. The integration gives a recurrence relation in m and n which on substitution of appropriate initial values gives the quoted result. The first few integrals (small m and n) were carried out explicitly as a check.

The generating function method (suggested by Professor C.A.Coulson) is more elegant.

$$\sum_{n=0}^{\infty} \frac{t^n}{n!} H_n(z) = \exp(-t^2 + 2tz)$$

Labelling our initial integral I_{nm} we can then quickly

show

$$\sum_{n=0}^{\infty} \sum_{m=0}^{\infty} \frac{u^n v^m}{m! n!} I_{nm} = \sqrt{\pi} e^{2uv} e^{2y(u+v)}$$

The k^{th} term of e^{2uv} (McLaurin Series) gives

$$\frac{2^k u^k v^k}{k!}, \quad \text{which leaves us to find the coefficient of}$$

$u^{n-k}v^{m-k}$ in $e^{2y(u+v)}$ ($k \leq m \leq n$). These powers of u and v can only arise in combination from the $(m+n-2k)$ th term of $e^{2y(u+v)}$ and thus has coefficient

$$(2y)^{n+m-2k} \binom{n+m-2k}{n-k} \frac{1}{(n+m-2k)!}$$

Thus

$$I_{nm} = n!m! \sqrt{\pi} \sum_{k=0}^m \frac{2^k}{k!} (2y)^{n+m-2k} \binom{n+m-2k}{n-k} \frac{1}{(n+m-2k)!}$$

which, using the definition of $L_m^{n-m}(x)$ given in Gradshteyn and Ryzhik, page 1037, formula 8.970.1, reduces to the expression above.

We are thus able to check the integral

$$\int_{-\infty}^{\infty} e^{-x^2} e^{itx} H_r(x) H_s(x) dx = \sqrt{\pi} 2^s s! e^{-\frac{1}{4}t^2} (it)^{r-s} L_s^{r-s}(\frac{1}{2}t^2)$$

which arose as a solution of a differential equation (indeed, much time was spent checking the solution before examining the integral tables, as the error exists also in Erdélyi et al. See below). This latter integral may be decomposed into sine and cosine forms. The cosine form gives equivalent results when appropriate alterations to parameters are made. The sine form, however, is, I believe, incorrectly given on page 841, formula 7.388.6 of Gradshteyn and Ryzhik:

$$\begin{aligned} \int_0^{\infty} e^{-x^2} \sin bx H_n(x) H_{n+2m+1}(x) dx &= \\ &= 2^n (-1)^m \sqrt{\frac{\pi}{2}} n! b^{2m} e^{-\frac{1}{4}b^2} L_n^{2m+1}(\frac{1}{2}b^2) \quad (b > 0) \end{aligned}$$

It should read

$$\int_0^{\infty} \dots dx = \sqrt{\pi} 2^{n-1} (-1)^m n! b^{2m+1} e^{-\frac{1}{4}b^2} L_n^{2m+1}(\frac{1}{2}b^2)$$

or

$$= 2^{n-\frac{1}{2}} \sqrt{\frac{\pi}{2}} n! (-1)^m b^{2m+1} e^{-\frac{1}{4}b^2} L_n^{2m+1}(\frac{1}{2}b^2)$$

(b > 0)

which is symmetrical in appearance to the cosine form.

These errors appear in the corresponding integrals in A. Erdélyi, W. Magnus, F. Oberhettinger, and F.G. Tricomi, Tables of Integral Transforms, McGraw-Hill, New York, 1954.

Thus Volume 1, Page 95, Transform 11 of

$$e^{-\frac{1}{2}x^2} \text{He}_n(x) \text{He}_{n+2m+1}(x) \text{ is given as}$$

$$(-1)^m 2^{-\frac{1}{2}} \pi^{\frac{1}{2}} n! y^{2m} e^{-\frac{1}{2}y^2} L_n^{2m+1}(y^2)$$

It should be

$$(-1)^m 2^{-\frac{1}{2}} \pi^{\frac{1}{2}} n! y^{2m+1} e^{-\frac{1}{2}y^2} L_n^{2m+1}(y^2)$$

Again, in Volume 2, Page 289, Integral 13 is given as

$$\int_{-\infty}^{\infty} \exp(-\frac{1}{2}(x-y)^2) \text{He}_m(x) \text{He}_n(x) dx$$

$$= (2\pi)^{\frac{1}{2}} m! y^{n-m} L_n^{n-m}(-y^2) \quad (m \leq n)$$

whereas it should be

$$= (2\pi)^{\frac{1}{2}} m! y^{n-m} L_m^{n-m}(-y^2) \quad (m \leq n)$$

which is simply a misprinting of n for m.

These errors have apparently not been reported in Mathematics of Computation (or Mathematical Tables and Other Aids to Computation), nor, as far as I know, elsewhere.

Yours sincerely,

John C. Nash

Professor Eugene Isaacson
Mathematics of Computation

APPENDIX 2

A Theorem of Bradley and Hughes

Let H be a self-adjoint operator on an inner-product vector space V over the complex field, and suppose that $|A\rangle$ is an eigenvector of H with eigenvalue E_a . Further let $\{W_m \mid m=1,2,\dots,N\}$ be a set of linear operators on V and U the subspace of V spanned by the set of vectors $\{|W_m A\rangle \mid m=1,2,\dots,N\}$. Suppose that the vector $|B\rangle$ and the real number E_b satisfy the hypervirial relations

$$(2.1) \quad \langle B | [H, W_m] | A \rangle = (E_b - E_a) \langle B | W_m | A \rangle$$

for $m=1,2,\dots,N$. Then $(H - E_b) |B\rangle$ is orthogonal to the subspace U .

Proof

As H is self-adjoint and W_m is linear Equation(2.1) reduces to

$$(2.2) \quad \langle (H - E_b)B | W_m A \rangle = \langle B | W_m (H - E_a)A \rangle$$

But from the conditions of the theorem

$$(2.3) \quad H |A\rangle = E_a |A\rangle$$

so that the right hand side of Equation (2.2) vanishes.

Hence

$$(2.4) \quad \langle (H - E_b)B | W_m A \rangle = 0$$

And, if this holds for $m=1,2,\dots,N$, then $(H - E_b) |B\rangle$ is orthogonal to $|W_m A\rangle$ for these values of m . That is, $(H - E_b) |B\rangle$ is orthogonal to the subspace U .

Corollary 1

Suppose that the set $\{W_m\}$ is such that $U=V$, that is the vectors $\{|W_m A\rangle \mid m=1,2,\dots,N\}$ span V (if V is an infinite dimensional Hilbert space this of course requires N to be a complete countable infinity). Then $(H-E_b) |B\rangle$ is orthogonal to V , and this implies that

$$(2.5) \quad (H-E_b) |B\rangle = 0$$

This then

Corollary 2

Suppose that the subspace U is invariant under H and that $|B\rangle \in U$. Then $(H-E_b) |B\rangle \in U$, but by the theorem it is also orthogonal to U . Hence, once again,

$$(2.6) \quad (H-E_b) |B\rangle = 0$$

The theorem shows that if N hypervirial relations are satisfied then in general $(H-E_b) |B\rangle$ is not necessarily zero but rather is constrained to be orthogonal to a subspace U of dimension N .

The above section is copied from Bradley and Hughes, 1969.

APPENDIX 3

Some Matrix Elements for Harmonic Oscillator Eigenfunctions

Following are the non-zero, lower triangle matrix elements for the operator

$$h = 2 \mu H = \hat{p}^2 + t x^2 + u x^4 + v x^6 \quad (A3.1)$$

(this operator is symmetric) in terms of the eigenfunctions ϕ_j of the harmonic oscillator Hamiltonian

$$H_0 = \frac{1}{2} (\hat{p}^2 + a^2 x^2) \quad (A3.2)$$

which has eigenvalues

$$(\phi_j, H \phi_j) = (j + \frac{1}{2}) a \quad (A3.3)$$

$$h_{r,r} = (2r+1)a + (2r+1)(t-a^2)/2a + 3u(2r^2+2r+1)/4a^2 + \\ + 5v(4r^3+6r^2+8r+3)/8a^3 \quad (A3.4)$$

$$h_{r,r-2} = (t-a^2)/2a + 2u(2r-1)/4a^2 + \\ + 15v(r^2-r+1)/8a^3 \quad (r(r-1))^{\frac{1}{2}} \quad (A3.5)$$

$$h_{r,r-4} = u/4a^2 + 3v(2r-3)/8a^3 \quad (r(r-1)(r-2)(r-3))^{\frac{1}{2}} \quad (A3.6)$$

$$h_{r,r-6} = v(r(r-1)(r-2)(r-3)(r-4)(r-5))^{\frac{1}{2}}/8a^3 \quad (A3.7)$$

APPENDIX 4

Earlier Research on Applications of Quantum Mechanical
Hypervirial Theorems

(References are given in the abbreviated form of J.Chem.Phys.)

General papers on the derivation of hypervirial theorems, special forms for particular operators, and the sufficiency theorems which state that if a wave function obeys "enough" hypervirial relations, then it is an eigenfunction of the relevant Hamiltonian:

- Björnå N., *Physica Norvegica*, 3, 151 (1968)
 Bradley C.J. & D.E.Hughes, *Int.J.Quantum Chem.*, 3, 699 (1969)
 Chen J.C.Y., *J.Chem.Phys.*, 39, 3167 (1963)
 Chen J.C.Y., *J.Chem.Phys.*, 40, 615 (1964)
 Coulson C.A., *Quart.J.Math.(Oxford)*, 16, 279 (1965)
 Epstein J.H. & S.T.Epstein, *Am.J.Phys.*, 30, 266 (1962)
 Epstein S.T., *J.Applied Physics*, 33, 235 (1962)
 Epstein S.T., *J.Chem.Phys.*, 42, 3813 (1965)
 Epstein S.T. & P.D.Robinson, *Phys.Rev.*, 129, 1396 (1963)
 Gordon S.L., *J.Chem.Phys.*, 42, 4184 (1965)
 Hirschfelder J.O., *J.Chem.Phys.*, 33, 1462 (1960)
 Hirschfelder J.O. & C.A.Coulson, *J.Chem.Phys.*, 36, 941 (1962)
 Pandres D., Jr., *Phys.Rev.*, 131, 886 (1963)
 Sullivan W.J., *J.Phys. A (Proc.Phys.Soc.)*, 1, 11 (1968)
 Sullivan W.J., *Quart.J.Math.(Oxford)*, 18, 285 (1967)
 Swenson R.J., *J.Chem.Phys.*, 52, 1012 (1970)
 Vetchinkin S.I., *J.Chem.Phys.*, 41, 1991 (1964)

Papers on the Off-Diagonal Hypervirial Method:

- Bradley C.J. & D.E.Hughes, *Int.J.Quantum Chem.*, 15, 687 (1967)
 Bradley C.J. & D.E.Hughes, *Int.J.Quantum Chem.*, 3, 699 (1970)
 Coulson C.A., *Quart.J.Math.(Oxford)*, 16, 279 (1965)
 Oppenheim I., *Proc.Int.Symposium on Molecular Structure
and Spectroscopy, Tokyo, 1962, Abstract B106-1*

Papers using hypervirial theorems to derive relationships between quantum mechanical matrix elements of various operators:

- Boyle L.L. & J.Murray, *J.Phys. B (Atom.Mol.Phys.)*, 2, 433 (1969)
 Boyle L.L. & L.Crawford-Jones, *J.Phys. B (Atom.Mol.Phys.)*
4, 28 (1971) and 4, 224 (1971)
 Epstein S.T., J.H.Epstein & B.Kennedy, *J.Math.Phys.*, 8,
 1747 (1967)
 Vetchinkin S.I., *Opt.Spectry.*, 14, 169 (1963)
 Vetchinkin S.I., *Soviet Physics 'Doklady'*, 7, 1132 (1963)
 Morgan D.J. & P.T.Landsberg, *Proc.Phys.Soc.*, 86, 261 (1965)

Papers concerning hypervirial theorems and the variational method and the use of hypervirial theorems as constraints on wave functions in variational calculations:

- Benston M.L. & D.P.Chong, Molec.Phys., 12, 487 (1967)
 Bjørnå N., J.Phys. B (Atom.Mol.Phys.), 4, 424 (1971)
 Bjørnå N., Physica Norvegica, 4, 223 (1970)
 Byers Brown W., J.Chem.Phys., 44, 567 (1966)
 Chong D.P., J.Chem.Phys., 48, 1413 (1968)
 Chong D.P., Theor.Chim.Acta, 11, 205 (1968)
 Chong D.P. & M.L.Benston, J.Chem.Phys., 49, 1302 (1968)
 Epstein S.T. & J.O.Hirschfelder, Phys.Rev., 123, 1495 (1961)

Papers on hypervirial theorems in perturbation theory:

- Arthurs A.M. & P.D.Robinson, Proc.Roy.Soc.(London),
A301, 507 (1967)
 Bangudu E.A. & P.D.Robinson, Proc.Phys.Soc., 86, 1259 (1965)
 Bjørnå N., Physica Norvegica, 3, 151 (1968)
 Byers Brown W., J.Chem.Phys., 44, 567 (1966)
 Carr W.J., Jr., Phys.Rev., 106, 414 (1957)
 Hirschfelder J.O., W. Byers Brown & Epstein, Advances in
 Quantum Chem., 1, 255 (1964)
 Robinson P.D., Proc.Roy.Soc., A283, 229 (1965)
 Robinson P.D., J.Chem.Phys., 47, 2319 (1967)

Papers concerning anharmonic oscillators:

- Bell S., R.Davidson & P.A.Warsop, J.Phys.B (Atom.Mol.Phys.)
3, 113 & 123 (1970)
 Bender C.M., J.Math.Phys., 11, 796 (1970)
 Bender C.M. & T.T.Wu, Phys.Rev., 184, 1231 (1969)
 Chan S.I., D.Stelman & L.E.Thompson, J.Chem.Phys., 41, 2828 (1964)
 Laane, J., J.Chem.Phys., 50, 776 (1969)
 Robinson P.D., J.Chem.Phys., 49, 2874 (1968)
 see also entries under Chan & Stelman and Heilbronner in
 Bibliography.

Papers concerning hypervirials in scattering theory:

- Epstein S.T. & P.D.Robinson, Phys.Rev., 129, 1396 (1963)
 Heaton M. & B.L.Moisewitsch, J.Phys.B (Atom.Mol.Phys.),
4, 332 (1971)
 McElroy M.B. & J.O.Hirschfelder, Phys.Rev., 131, 1589 (1963)
 Robinson P.D. & J.O.Hirschfelder, Phys.Rev., 129, 1391 (1963)

APPENDIX 5

A Catalogue of Computer Programs Used

Following is a brief catalogue of computer programs which were developed in connection with the research project. ~~It was hoped to provide listings of many of these but the recent power crisis has hindered the production of such listings.~~

- Solution of the eigenproblem $(P - \lambda Q)\underline{x} = 0$:

Subroutines ADJ, REDN and ITN are used.

ADJ- forms $P - zQ$ to adjust the inverse iteration to converge to the desired eigensolution.

REDN- performs the Gaussian elimination reduction of P to give a triangular form ready for iteration. Q is also adjusted appropriately, but in general remains square.

ITN- carries out the iteration and tests for convergence using a form of the eigenvalue.

- Solution of the symmetric matrix eigenvalue problem:

Subroutine JNH performs Householder tridiagonalization of the original matrix in TRY, finds eigenvalues by a Sturm sequence bisection method in VAL, and vectors by inverse iteration in VEC. JNH checks for degeneracy and orthogonalizes vectors for close eigenvalues. Some loss of accuracy occurs for degeneracy of high multiplicity. Only selected eigensolutions need be found.

- Off-Diagonal Hypervirial Method for oscillators:

I. Hypervirial operators x and $d/dx = D$

OSC3- allows operators $W = D^k x^j$, k, j integers

OS3A- allows operators $W = x^j D^k$, k, j integers

OSC5- described in Section 4.5. OSC1, OSC2, OSC4, and OS4A were development variants. These programs all work with a power of x type basis, and require (some) of the following subroutines:

EKS- performs operation of x on wave function as represented by a vector of coefficients.

PEA- same for D .

VSUM- forms $\underline{a} + g\underline{b}$ in $\underline{a}$, i.e. sum of vector coefficients.

MAKE or BUILD- form the P and Q matrices given initial wave function and set of hypervirial operators. P and Q need not be square so additional tests may be performed at the same time e.g. extra row values.

HOME- forms function which results from Hamiltonian H acting on a wave function.

ENER- calculates energy of a wave function given in power of x type basis as well as Euclidean norm.

QUER- tests wave function by forming matrix elements $(A, H, W A)$ for W of type as in OSC5.

PLUG- calculates plotting data for wave functions.

INTL and Z- calculate integrals $x^s \exp(-ax^2) dx$

The programs allow "up-down" iterative calculations.

II. Hypervirial operator $\exp(itx)$

OSC6- carries out the calculations almost without extra subroutines.

- Transformations between power of x and harmonic oscillator bases as described in Section 4.6: HP

- Testing Wave functions by Hypervirial Theorems:

OSTEST- uses subroutines from OSC5 to see how well wave functions obey diagonal and off-diagonal hypervirial theorems. Also allows energies and plotting data to be found.

- Finding non-linear parameters - inverse interpolation:

INVINT- inverse Everett interpolation at equal intervals.

- Wave functions for Anharmonic Oscillators:

ANOSC- based on matrix elements in Appendix 3.

- Matrix elements ($\chi_j(N), x^n \chi_k(N)$):

MOMENTS- two versions as described in Section 2.5.

PROF- recurrence relation method described in Section 2.5.

- Overlaps between wave functions:

OVERLAPS- calculates $t_j = \sum_{k=1}^N V_{kj} b_k$ for vector of coefficients $\underline{b}$ with "known" vectors stored in matrix V .

- Listing programs in suitable format:

GOODLIST- lists programs in paginated form with a specified number of lines per page together with identifier and line sequence numbers.

BIBLIOGRAPHY

- Abramowitz M. & I.A.Stegun, 1965, Handbook of Mathematical Functions, Dover, New York.
- Bradley C.J, & D.E.Hughes, 1967, Int.J.Quantum Chem., 1S, 687-701.
- Bradley C.J. & D.E.Hughes, 1969, Int.J.Quantum Chem., 3, 699-710.
- Brand J.C.D. & J.C.Speakman, 1960, Molecular Structure, Edward Arnold, London.
- Carrier G.F., M.Krook, & C.E.Pearson, 1966, Functions of a Complex Variable, McGraw-Hill, New York.
- Chan S.I. & D.Stelman, 1963, J.Mol.Spectry., 10, 278-299
- Coulson C.A., 1965, Quart.J.Maths.(Oxford), Second Series, 16, 279-287.
- Dennerly P. & A.Krzywicki, 1967, Mathematics for Physicists, Harper & Row, New York.
- Erdelyi A., W.Magnus, F.Oberhettinger & F.G.Tricomi, 1953, Higher Transcendental Functions, McGraw-Hill, New York.
- Finkbeiner D.T., 1966, Introduction to Matrices and Linear Transformations, W.H.Freeman, San Francisco.
- Fox L., 1964, An Introduction to Numerical Linear Algebra, Clarendon Press, Oxford.
- Fox L. & D.F.Mayers, 1968, Computing Methods for Scientists and Engineers, Clarendon Press, Oxford.

- Gottfried K., 1966, Quantum Mechanics, Volume 1: Fundamentals, Benjamin, New York.
- Gradshteyn I.S. & I.M. Ryzhik, 1965, Tables of Integrals, Series and Products, Translated by Yu.V. Geronimus & M.Yu. Tseytlin, Academic Press, New York & London.
- Heilbronner E., H. Günthard & R. Gerdil, 1956, *Helv. Chim. Acta*, 39, 1171-1181.
- Heilbronner E., H. Rutishauser & F. Gerson, 1959a & 1959b, *Helv. Chim. Acta*, 42, 2289-2303 & 2304-2314.
- Herzberg, 1944, Atomic Spectra & Atomic Structure, Translated by J.W.T. Spinks, Dover, New York.
- Hirschfelder J.O., 1960, *J. Chem. Phys.*, 33, 1462-66.
- Isaacson E. & H.B. Keller, 1966, Analysis of Numerical Methods, Wiley, New York.
- Katriel J., 1970, *J. Phys. B, (Atom. & Mol. Phys.)*, 3, 1315-20.
- Katriel J. & G. Adam, 1970, *J. Phys. B, (Atom. & Mol. Phys.)*, 3, 13-15.
- Kemble E.C., 1937 or 1958, The Fundamental Principles of Quantum Mechanics, Dover, New York.
- McMillan C., Jr., 1970, Mathematical Programming, Wiley, New York.
- McWeeny R. & C.A. Coulson, 1948, *Proc. Camb. Phil. Soc.*, 44, 413-422.
- Messiah A., 1961, Quantum Mechanics, Translated by G.M. Temmer, North-Holland, Amsterdam.

- Mostow G.D. & J.H.Sampson, 1969, Linear Algebra, McGraw-Hill, New York.
- Nash J.C., 1969, Dissertation submitted for the requirements of the Diploma in Advanced Mathematics, Oxford.
- Newing R.A. & J.Cunningham, 1967, Quantum Mechanics, Oliver & Boyd, Edinburgh & London.
- Oppenheim I., 1962, Proceedings of the International Symposium on Molecular Structure and Spectroscopy, Tokyo, Abstract B106-1.
- Ralston A., 1965, A First Course in Numerical Analysis, McGraw-Hill, New York.
- Sneddon I.N., 1961, Special Functions of Mathematical Physics and Chemistry, Oliver & Boyd, Edinburgh & London.
- Sokolnikoff I.S. & R.M.Redheffer, 1966, Mathematics of Physics and Modern Engineering, Second Edition, McGraw-Hill, New York.
- Szegö G., 1959, Orthogonal Polynomials, Revised Edition, American Mathematical Society, New York.
- Widder D.V., 1961, Advanced Calculus, Second Edition, Prentice-Hall, Englewood Cliffs, N.J.
- Wilkinson J.H., 1963, Rounding Errors in Algebraic Processes, Notes on Applied Science No. 32, Her Majesty's Stationery Office, London.
- Wilkinson J.H., 1965, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford.

Wilkinson J.H. & C.Reinsch, 1971, Linear Algebra, Handbook for Automatic Computation, Volume II, Springer Verlag, Berlin.

Witschel W., 1970, J.Phys. B,(Atom. & Mol.Phys.), 3, L120.

Addendum

Coulson C.A. & J.C.Nash, paper entitled "Transition Moment Integrals and the Hypervirial Theorem", accepted for publication in the Journal of Physics, Part B

Nash J.C., 1972, Mathematical Table Erratum in Mathematics of Computation, to be published in April, 1972.

Carriera L.A., I.M. Mills & W.B. Person, J.Chem.Phys, 56, 1444
(1972)

```

SUBROUTINE REDN(M,N,P,Q,IFAIL)
C THIS SUBROUTINE PERFORMS GAUSSIAN ELIMINATION ON MATRIX P WITH PROPER REDN 10
C ADJUSTMENT OF Q, WHICH MAY BE A MATRIX OF RIGHT HAND SIDE VECTORS REDN 20
C IN LINEAR EQUATION OR INVERSION CALCULATIONS OR THE Q MATRIX IN REDN 30
C THE SOLUTION OF THE  $(P-E \cdot Q) \cdot X = 0$  EIGENPROBLEM REDN 40
C J.C. NASH SEPTEMBER 1971 REDN 50
C IFAIL SET TO 1 IF P SINGULAR, 0 OTHERWISE REDN 60
C MULTIPLIERS ARE LEFT IN LOWER TRIANGLE OF P REDN 70
C M IS DIMENSIONALITY, N IS ORDER OF P & Q REDN 80
C DIMENSION P(M,M),Q(M,M) REDN 90
C IFAIL=0 REDN 100
C IF(N.LE.1)RETURN REDN 110
C N1=N-1 REDN 120
C DO 20 I=1,N1 REDN 130
C SEARCH FOR PIVOT REDN 140
C IT=1 REDN 150
C BIG=ABS(P(I,I)) REDN 160
C I1=I+1 REDN 170
C DO 1 K=I1,N REDN 180
C IF (ABS(P(K,I)).LE.BIG) GO TO 1 REDN 190
C IT=K REDN 200
C BIG=ABS(P(K,I)) REDN 210
C CONTINUE REDN 220
C IF(BIG.EQ.0.0)GO TO 9 REDN 230
C IF(IT.EQ.I) GO TO 3 REDN 240
C INTERCHANGE ROWS IF PIVOT NOT IN POSITION REDN 250
C DO 2 J=1,N REDN 260
C SAVE=P(IT,J) REDN 270
C P(IT,J)=P(I,J) REDN 280
C P(I,J)=SAVE REDN 290
C SAVE=Q(IT,J) REDN 300
C Q(IT,J)=Q(I,J) REDN 310
C Q(I,J)=SAVE REDN 320
C FORM MULTIPLIERS REDN 330
C REDN 340

```

```

3      DO 6 K=1,N
        P(K,I)=P(K,I)/P(I,I)
        DO 4 J=1,N
          C ADJUST ELEMENTS OF P LEAVING MULTIPLIERS
            4 P(K,J)=P(K,J)-P(K,I)*P(I,J)
              DO 5 J=1,N
                C ADJUST ELEMENTS OF Q
                  5 Q(K,J)=Q(K,J)-P(K,I)*Q(I,J)
                    6 P(K,I)=0.0
                      20 CONTINUE
                        IF(P(N,N))0,9,0
                          C OUTPUT REDUCED MATRICES
                            WRITE(6,205)
                              205 FORMAT('OP OUTPUT FROM REDN')
                                WRITE(6,200)((P(I,J),J=1,N),I=1,N)
                                  200 FORMAT(1H,SE16.8)
                                    WRITE(6,206)
                                      206 FORMAT('OQ OUTPUT FROM REDN')
                                        WRITE(6,200)((Q(I,J),J=1,N),I=1,N)
                                          C DIAGONAL PRODUCT AT THIS STAGE GIVES DET(P)
                                            RETURN
                                              9 IFAIL=1
                                                RETURN
                                                  END

```

```

REDN 350
REDN 360
REDN 370
REDN 380
REDN 390
REDN 400
REDN 410
REDN 420
REDN 430
REDN 440
REDN 450
REDN 460
REDN 470
REDN 480
REDN 490
REDN 500
REDN 510
REDN 520
REDN 530
REDN 540
REDN 550
REDN 560
REDN 570
REDN 580

```

NAME OF PROGRAM OR SEGMENT ADJ

```

SUBROUTINE ADJ(M,N,P,Q,G)
C THIS SUBROUTINE FORMS P=G*Q IN P
C Q UNCHANGED
C J.C. NASH SEPTEMBER 1971
C M IS DIMENSIONALITY, N IS ORDER OF P & Q
  DIMENSION P(M,M),Q(M,M)
  DO 4 I=1,N
    DO 4 J=1,N
      P(I,J)=P(I,J)-G*Q(I,J)
  4 RETURN
END

```

ADJ	10
ADJ	20
ADJ	30
ADJ	40
ADJ	50
ADJ	60
ADJ	70
ADJ	80
ADJ	90
ADJ	100
ADJ	110

NAME OF PROGRAM OR SEGMENT ITN

```

SUBROUTINE ITN(M,N,TOL,ITM,P,Q,X,G,Y)
C THIS SUBROUTINE PERFORMS THE ITERATION TO SOLVE EIGENPROBLEM
C (P-E*Q)*X=0 AFTER REDUCTION IN SUBROUTINE REDN
C TOL IS TOLERANCE SUCH THAT WHEN RELATIVE DIFFERENCE BETWEEN
C SUCCESSIVE EIGENVALUE APPROXIMATIONS IS LESS THAN TOL AND
C MORE THAN 4 ITERATIONS HAVE PASSED, ITERATION STOPS
C M IS DIMENSIONALITY, N IS ORDER OF P & Q
C LIM IS A LIMIT ON THE NUMBER OF ITERATIONS (GREATER THAN 4)
C X IS OUTPUT VECTOR, G OUTPUT EIGENVALUE
C ON INPUT G IS REAL NUMBER NEAREST TO WHICH WE WANT ROOT
C Y IS TEMPORARY STORAGE
DOUBLE PRECISION S,SAVE
DIMENSION P(M,M),Q(M,M),X(M),Y(M)
NT=0
E=1000.
1 NT=NT+1
S=0.0
DO 310 J=1,N
310 S=S+DBLE(Q(N,J))*DBLE(X(J))
Y(N)=S/P(N,N)
C IF N=1, ONLY ONE ROOT SO EXIT
IF(N.IT.2)GO TO 90
DO 2 I=2,N
IT=N-I+1
IS=IT+1
SAVE=0.0
DO 3 J=1,N
3 SAVE=SAVE+DBLE(Q(IT,J))*DBLE(X(J))
DO 4 J=IS,N
4 SAVE=SAVE-DBLE(P(IT,J))*DBLE(Y(J))
Y(IT)=SAVE/P(IT,IT)
C FIND MAX Y
IT=1
BTG=ABS(Y(1))
ITN 10
ITN 20
ITN 30
ITN 40
ITN 50
ITN 60
ITN 70
ITN 80
ITN 90
ITN 100
ITN 110
ITN 120
ITN 130
ITN 140
ITN 150
ITN 160
ITN 170
ITN 180
ITN 190
ITN 200
ITN 210
ITN 220
ITN 230
ITN 240
ITN 250
ITN 260
ITN 270
ITN 280
ITN 290
ITN 300
ITN 310
ITN 320
ITN 330
ITN 340

```

NAME OF PROGRAM OR SEGMENT ITN

```

DO 5 J=2,N
IF(ABS(Y(J)).LE.BIG)GO TO 5
IT =J
BIG=ABS(Y(J))
CONTINUE
D=1.0/Y(IT)
DO 7 J=1,N
X(J)=Y(J)*D
WRITE(6,106) D
FORMAT(3H E=,E16.8)
IF(BIG*ABS(D-E).LE.TOL.AND.NIT.GT.4)GO TO 9
E=D
IF(NIT.EQ.LIM)GO TO 10
GO TO 1
WRITE(6,107)
FORMAT(20H NIT IS IMPROPTIOUS)
GO TO 9
D=Y(N)**(-1)
X(N)=1.0
LTM=NIT
WRITE(6,109)(X(J),J=1,N)
FORMAT(' VECTOR',5E16.8)
G=G+D
RETURN
END

```

ITN	350
ITN	360
ITN	370
ITN	380
ITN	390
ITN	400
ITN	410
ITN	420
ITN	430
ITN	440
ITN	450
ITN	460
ITN	470
ITN	480
ITN	490
ITN	500
ITN	510
ITN	520
ITN	530
ITN	540
ITN	550
ITN	560
ITN	570
ITN	580
ITN	590

NAME OF PROGRAM OR SEGMENT PROF

```

MASTER PROF
C TRANSITION MOMENT INTEGRALS BY RECURRENCE RELATION
DOUBLE PRECISION C(100),D(100),G(100),H(100),CUMP,DUMP,Z
DOUBLE PRECISION Q,P
READ(5,105)AK
FORMAT(F10.5)
105 READ(5,100)EA,EB
1 READ(5,100)EA,EB
100 FORMAT(2E20.10)
IF(EA.EQ.0.0.AND.EB.EQ.0.0)STOP
READ(5,101)P
101 FORMAT(D20.10)
READ(5,103)IPAR
103 FORMAT(I4)
IF(IPAR)2,3,2
3 C(1)=-0.5*P
DUMP=0.0D0
GO TO 4
2 DUMP=P
C(1)=0.0D0
4 CONTINUE
WRITE(6,200)EA,EB
200 FORMAT('0INPUT ENERGIES',2E16.8)
WRITE(6,201)AK
201 FORMAT('0COEFFICIENT OF X**4',F10.5)
P=(EA-EB)**2
Q=EA+EB
300 FORMAT(' P,Q',2D20.10)
WRITE(6,300)P,Q
CUMP=0.0
IF(EA.EQ.EB)CUMP=1.0
U=-0.5/AK
C(2)=U*(P*C(1)-Q*CUMP)/(3.0*Z(2,3))

```

PROF 10
 PROF 20
 PROF 30
 PROF 40
 PROF 50
 PROF 60
 PROF 70
 PROF 80
 PROF 90
 PROF 100
 PROF 110
 PROF 120
 PROF 130
 PROF 140
 PROF 150
 PROF 160
 PROF 170
 PROF 180
 PROF 190
 PROF 200
 PROF 210
 PROF 220
 PROF 230
 PROF 240
 PROF 250
 PROF 260
 PROF 270
 PROF 280
 PROF 290
 PROF 300
 PROF 310
 PROF 320

NAME OF PROGRAM OR SEGMENT PROF

```

C(3)=U*(P*C(2)-Q*C(1)+0.25*CUMP)/(5.0*Z(4,3))      PROF 330
D(1)=U*P*DUMP/(2.0*Z(1,3))                          PROF 340
D(2)=U*(P*D(1)-Q*DUMP)/(4.0*Z(3,3))                 PROF 350
D(3)=U*(P*D(2)-Q*D(1)+0.25*DUMP)/(6.0*Z(5,3))       PROF 360
H(2)=120.*D(2)                                         PROF 370
G(2)=24.*C(2)                                          PROF 380
G(1)=-2*C(1)                                           PROF 390
G(3)=-720.*C(3)                                        PROF 400
H(1)=-6.*D(1)                                          PROF 410
H(3)=-5040.*D(3)                                       PROF 420
WRITE(6,244)                                           PROF 430
244  FORMAT('0 J',15X,'C',15X,'D',15X,'G',15X,'H',1X)  PROF 440
J=0                                                     PROF 450
WRITE(6,245)J,CUMP,DUMP,CUMP,DUMP                    PROF 460
245  WRITE(6,245)(J,C(J),D(J),G(J),H(J),J=1,2)         PROF 470
      FORMAT(1H,14,4D20.10)                            PROF 480
DO 10 J=3,99                                           PROF 490
D(J+1)=U*(P*D(J)-Q*D(J-1)+0.25*D(J-2))/(FLOAT(2*J+2)*Z(2*J+1,3))  PROF 500
C(J+1)=U*(P*C(J)-Q*C(J-1)+0.25*C(J-2))/(FLOAT(2*J+1)*Z(2*J,3))    PROF 510
G(J+1)=U*(P*G(J)+Q*G(J-1)+Z(2*J-1,2)+0.25*G(J-2)*Z(2*J-3,4))/Z(2*J,4)  PROF 520
1,2)**(-1.0)                                           PROF 530
H(J+1)=U*(P*H(J)+Q*H(J-1)*Z(2*J,2)+0.25*H(J-2)*Z(2*J-2,4))/Z(2*J+1,3)  PROF 540
1,2)**(-1.0)                                           PROF 550
WRITE(6,245)J,C(J),D(J),G(J),H(J)                   PROF 560
IF(DABS(H(J+1)).GT.1.0E+60)GO TO 1                    PROF 570
IF(DABS(G(J+1)).GT.1.0E+60)GO TO 1                    PROF 580
CONTINUE                                              PROF 590
10  J=100                                              PROF 600
WRITE(6,245)J,C(J),D(J),G(J),H(J)                   PROF 610
GO TO 1                                               PROF 620
END                                                  PROF 630
DOUBLE PRECISION FUNCTION Z(L,M)                     PROF 640

```

NAME OF PROGRAM OR SEGMENT PROF

LT=L	PROF 650
M1=M-1	PROF 660
Z=FLOAT(LT)	PROF 670
IF(M1.LT.1)RETURN	PROF 680
DO 11 J=1,M1	PROF 690
LT=LT+1	PROF 700
Z=Z*FLOAT(LT)	PROF 710
11 RETURN	PROF 720
END	PROF 730
FINISH	PROF 740

NAME OF PROGRAM OR SEGMENT MOM1

```

MASTER MOMENTS
C MOMENTS OF X**N BETWEEN OSCILLATOR WAVE FUNCTIONS
C USING POWER OF X TYPE BASIS
DOUBLE PRECISION Z,P,T(250)
DIMENSION X(100),Y(100),CI(100),W1(100),W2(100),W3(100),IM(20)
2 IREP=0
1 READ(5,102)IV,NM,A0,A2,A4,A6,EG
102 FORMAT(2I4,4F10.5,E16.8)
IF(NM.EQ.0)STOP
READ(5,103)(CI(J),J=1,NM)
103 FORMAT(5E16.8)
101 FORMAT(14)
WRITE(6,201)IV,NM,A0
201 FORMAT('0STATE',14,' LENGTH OF VECTOR=',14,' EXPONENT=',F10.5)
WRITE(6,202)A2,A4,A6
202 FORMAT('0HAMILTONIAN=P**2+',F10.5,'*X**2+',F10.5,'*X**4+',F10.5,'*X**6')
1X**6')
WRITE(6,203)EG
203 FORMAT('0EIGENVALUE=2*ENERGY=',E16.8)
WRITE(6,204)
204 FORMAT('0VECTOR')
WRITE(6,205)(CI(J),J=1,NM)
205 WRITE(6,212)
212 FORMAT('0VECTOR IN POLYNOMIAL BASIS')
205 FORMAT(1H,5E16.8)
CALL HP(NM,CI,X,A0,0,W1,W2,W3)
WRITE(6,205)(X(J),J=1,NM)
READ(5,150)IM
150 FORMAT(20I4)
MX=IM(1)
A=A0
DO 15 K=2,20
MOM1 10
MOM1 20
MOM1 30
MOM1 40
MOM1 50
MOM1 60
MOM1 70
MOM1 80
MOM1 90
MOM1 100
MOM1 110
MOM1 120
MOM1 130
MOM1 140
MOM1 150
MOM1 160
MOM1 170
MOM1 180
MOM1 190
MOM1 200
MOM1 210
MOM1 220
MOM1 230
MOM1 240
MOM1 250
MOM1 260
MOM1 270
MOM1 280
MOM1 290
MOM1 300
MOM1 310
MOM1 320

```

NAME OF PROGRAM OR SEGMENT MOM1

15	IF(IM(K).GT.MX)MX=IM(K)	MOM1 330
	CONTINUE	MOM1 340
	MX=MX+NM+NM	MOM1 350
16	DO 16 K=1,MX	MOM1 360
	T(K)=Z(K-2,A)	MOM1 370
	WRITE(6,500)	MOM1 380
500	FORMAT('0INTEGRALS')	MOM1 390
	WRITE(6,501)(T(IK),IK=1,MX)	MOM1 400
501	FORMAT(1H ,5D20.10)	MOM1 410
	K=1	MOM1 420
10	IF(IM(K))6,12,12	MOM1 430
12	P=0.0D0	MOM1 440
	KAP=IM(K)	MOM1 450
	DO 13 I=1,NM	MOM1 460
	DO 13 J=1,NM	MOM1 470
	KISS=I+J+KAP	MOM1 480
13	P=P+DBLE(X(I))*DBLE(X(J))*T(KISS)	MOM1 490
	WRITE(6,250)KAP,P	MOM1 500
250	FORMAT('0EXPECTATION OF X**',I4,' ',D20.10)	MOM1 510
	K=K+1	MOM1 520
	IF(K=20)10,10,6	MOM1 530
6	IF(IREP-1)8,9,8	MOM1 540
8	READ(5,101)IOD	MOM1 550
	IF(IOD)7,1,7	MOM1 560
7	IREP=1	MOM1 570
	DO 21 J=1,NM	MOM1 580
21	Y(J)=X(J)	MOM1 590
	NM1=NM	MOM1 600
	EG1=EG	MOM1 610
	B2=A2	MOM1 620
	B4=A4	MOM1 630
	B6=A6	MOM1 640

NAME OF PROGRAM OR SEGMENT MOM1

```

IV1=IV
AS=A0
GO TO 1
9 WRITE(6,207)IV,IV1
207 FORMAT('00FF-DIAGONAL HYPERVIRIAL TEST BETWEEN STATES',I4,' AND ',MOM1
114)
IF(A2.NE.B2.OR.A4.NE.B4.OR.A6.NE.B6)GO TO 109
READ(5,150)IM
A=0.5*(A0+AS)
MX=IM(1)
DO 40 K=2,20
IF(IM(K).GT.MX)MX=IM(K)
40 CONTINUE
MX=MX+NM+NM1
DO 41 K=1,MX
41 T(K)=Z(K-2,A)
WRITE(6,500)
WRITE(6,501)(T(IK),IK=1,MX)
K=1
30 KAP=IM(K)
IF(KAP)32,33,33
33 P=0.0D0
DO 34 I=1,NM
DO 34 J=1,NM1
KISS=I+J+KAP
34 P=P+DBLE(X(I))*DBLE(Y(J))*T(KISS)
WRITE(6,251)KAP,P
251 FORMAT('OMATRIX ELEMENT OF X**',I4,' = ',D20.10)
K=K+1
IF(K=20)30,30,32
32 READ(5,101)IOD
IF(IOD)7,2,1

```

MOM1 650
MOM1 660
MOM1 670
MOM1 680
MOM1 690
MOM1 700
MOM1 710
MOM1 720
MOM1 730
MOM1 740
MOM1 750
MOM1 760
MOM1 770
MOM1 780
MOM1 790
MOM1 800
MOM1 810
MOM1 820
MOM1 830
MOM1 840
MOM1 850
MOM1 860
MOM1 870
MOM1 880
MOM1 890
MOM1 900
MOM1 910
MOM1 920
MOM1 930
MOM1 940
MOM1 950
MOM1 960

NAME OF PROGRAM OR SEGMENT MOM1

109 WRITE(6,211)
 211 FORMAT('OERROR = DIFFERENT HAMILTONIANS')
 GO TO 2
 END

MOM1 970
 MOM1 980
 MOM1 990
 MOM11000

10
 20
 30
 40
 50
 60
 70
 80
 90
 100
 110
 120
 130
 140
 150
 160
 170
 180
 190
 200
 210
 220
 230
 240
 250
 260
 270
 280
 290
 300
 310
 320
 330
 340
 350
 360
 370
 380
 390
 400
 410
 420
 430
 440
 450
 460
 470
 480
 490
 500
 510
 520
 530
 540
 550
 560
 570
 580
 590
 600
 610
 620
 630
 640
 650
 660
 670
 680
 690
 700
 710
 720
 730
 740
 750
 760
 770
 780
 790
 800
 810
 820
 830
 840
 850
 860
 870
 880
 890
 900
 910
 920
 930
 940
 950
 960
 970
 980
 990
 1000
 1010
 1020
 1030
 1040
 1050
 1060
 1070
 1080
 1090
 1100
 1110
 1120
 1130
 1140
 1150
 1160
 1170
 1180
 1190
 1200
 1210
 1220
 1230
 1240
 1250
 1260
 1270
 1280
 1290
 1300
 1310
 1320
 1330
 1340
 1350
 1360
 1370
 1380
 1390
 1400
 1410
 1420
 1430
 1440
 1450
 1460
 1470
 1480
 1490
 1500
 1510
 1520
 1530
 1540
 1550
 1560
 1570
 1580
 1590
 1600
 1610
 1620
 1630
 1640
 1650
 1660
 1670
 1680
 1690
 1700
 1710
 1720
 1730
 1740
 1750
 1760
 1770
 1780
 1790
 1800
 1810
 1820
 1830
 1840
 1850
 1860
 1870
 1880
 1890
 1900
 1910
 1920
 1930
 1940
 1950
 1960
 1970
 1980
 1990
 2000
 2010
 2020
 2030
 2040
 2050
 2060
 2070
 2080
 2090
 2100
 2110
 2120
 2130
 2140
 2150
 2160
 2170
 2180
 2190
 2200
 2210
 2220
 2230
 2240
 2250
 2260
 2270
 2280
 2290
 2300
 2310
 2320
 2330
 2340
 2350
 2360
 2370
 2380
 2390
 2400
 2410
 2420
 2430
 2440
 2450
 2460
 2470
 2480
 2490
 2500
 2510
 2520
 2530
 2540
 2550
 2560
 2570
 2580
 2590
 2600
 2610
 2620
 2630
 2640
 2650
 2660
 2670
 2680
 2690
 2700
 2710
 2720
 2730
 2740
 2750
 2760
 2770
 2780
 2790
 2800
 2810
 2820
 2830
 2840
 2850
 2860
 2870
 2880
 2890
 2900
 2910
 2920
 2930
 2940
 2950
 2960
 2970
 2980
 2990
 3000
 3010
 3020
 3030
 3040
 3050
 3060
 3070
 3080
 3090
 3100
 3110
 3120
 3130
 3140
 3150
 3160
 3170
 3180
 3190
 3200
 3210
 3220
 3230
 3240
 3250
 3260
 3270
 3280
 3290
 3300
 3310
 3320
 3330
 3340
 3350
 3360
 3370
 3380
 3390
 3400
 3410
 3420
 3430
 3440
 3450
 3460
 3470
 3480
 3490
 3500
 3510
 3520
 3530
 3540
 3550
 3560
 3570
 3580
 3590
 3600
 3610
 3620
 3630
 3640
 3650
 3660
 3670
 3680
 3690
 3700
 3710
 3720
 3730
 3740
 3750
 3760
 3770
 3780
 3790
 3800
 3810
 3820
 3830
 3840
 3850
 3860
 3870
 3880
 3890
 3900
 3910
 3920
 3930
 3940
 3950
 3960
 3970
 3980
 3990
 4000
 4010
 4020
 4030
 4040
 4050
 4060
 4070
 4080
 4090
 4100
 4110
 4120
 4130
 4140
 4150
 4160
 4170
 4180
 4190
 4200
 4210
 4220
 4230
 4240
 4250
 4260
 4270
 4280
 4290
 4300
 4310
 4320
 4330
 4340
 4350
 4360
 4370
 4380
 4390
 4400
 4410
 4420
 4430
 4440
 4450
 4460
 4470
 4480
 4490
 4500
 4510
 4520
 4530
 4540
 4550
 4560
 4570
 4580
 4590
 4600
 4610
 4620
 4630
 4640
 4650
 4660
 4670
 4680
 4690
 4700
 4710
 4720
 4730
 4740
 4750
 4760
 4770
 4780
 4790
 4800
 4810
 4820
 4830
 4840
 4850
 4860
 4870
 4880
 4890
 4900
 4910
 4920
 4930
 4940
 4950
 4960
 4970
 4980
 4990
 5000
 5010
 5020
 5030
 5040
 5050
 5060
 5070
 5080
 5090
 5100
 5110
 5120
 5130
 5140
 5150
 5160
 5170
 5180
 5190
 5200
 5210
 5220
 5230
 5240
 5250
 5260
 5270
 5280
 5290
 5300
 5310
 5320
 5330
 5340
 5350
 5360
 5370
 5380
 5390
 5400
 5410
 5420
 5430
 5440
 5450
 5460
 5470
 5480
 5490
 5500
 5510
 5520
 5530
 5540
 5550
 5560
 5570
 5580
 5590
 5600
 5610
 5620
 5630
 5640
 5650
 5660
 5670
 5680
 5690
 5700
 5710
 5720
 5730
 5740
 5750
 5760
 5770
 5780
 5790
 5800
 5810
 5820
 5830
 5840
 5850
 5860
 5870
 5880
 5890
 5900
 5910
 5920
 5930
 5940
 5950
 5960
 5970
 5980
 5990
 6000
 6010
 6020
 6030
 6040
 6050
 6060
 6070
 6080
 6090
 6100
 6110
 6120
 6130
 6140
 6150
 6160
 6170
 6180
 6190
 6200
 6210
 6220
 6230
 6240
 6250
 6260
 6270
 6280
 6290
 6300
 6310
 6320
 6330
 6340
 6350
 6360
 6370
 6380
 6390
 6400
 6410
 6420
 6430
 6440
 6450
 6460
 6470
 6480
 6490
 6500
 6510
 6520
 6530
 6540
 6550
 6560
 6570
 6580
 6590
 6600
 6610
 6620
 6630
 6640
 6650
 6660
 6670
 6680
 6690
 6700
 6710
 6720
 6730
 6740
 6750
 6760
 6770
 6780
 6790
 6800
 6810
 6820
 6830
 6840
 6850
 6860
 6870
 6880
 6890
 6900
 6910
 6920
 6930
 6940
 6950
 6960
 6970
 6980
 6990
 7000
 7010
 7020
 7030
 7040
 7050
 7060
 7070
 7080
 7090
 7100
 7110
 7120
 7130
 7140
 7150
 7160
 7170
 7180
 7190
 7200
 7210
 7220
 7230
 7240
 7250
 7260
 7270
 7280
 7290
 7300
 7310
 7320
 7330
 7340
 7350
 7360
 7370
 7380
 7390
 7400
 7410
 7420
 7430
 7440
 7450
 7460
 7470
 7480
 7490
 7500
 7510
 7520
 7530
 7540
 7550
 7560
 7570
 7580
 7590
 7600
 7610
 7620
 7630
 7640
 7650
 7660
 7670
 7680
 7690
 7700
 7710
 7720
 7730
 7740
 7750
 7760
 7770
 7780
 7790
 7800
 7810
 7820
 7830
 7840
 7850
 7860
 7870
 7880
 7890
 7900
 7910
 7920
 7930
 7940
 7950
 7960
 7970
 7980
 7990
 8000
 8010
 8020
 8030
 8040
 8050
 8060
 8070
 8080
 8090
 8100
 8110
 8120
 8130
 8140
 8150
 8160
 8170
 8180
 8190
 8200
 8210
 8220
 8230
 8240
 8250
 8260
 8270
 8280
 8290
 8300
 8310
 8320
 8330
 8340
 8350
 8360
 8370
 8380
 8390
 8400
 8410
 8420
 8430
 8440
 8450
 8460
 8470
 8480
 8490
 8500
 8510
 8520
 8530
 8540
 8550
 8560
 8570
 8580
 8590
 8600
 8610
 8620
 8630
 8640
 8650
 8660
 8670
 8680
 8690
 8700
 8710
 8720
 8730
 8740
 8750
 8760
 8770
 8780
 8790
 8800
 8810
 8820
 8830
 8840
 8850
 8860
 8870
 8880
 8890
 8900
 8910
 8920
 8930
 8940
 8950
 8960
 8970
 8980
 8990
 9000
 9010
 9020
 9030
 9040
 9050
 9060
 9070
 9080
 9090
 9100
 9110
 9120
 9130
 9140
 9150
 9160
 9170
 9180
 9190
 9200
 9210
 9220
 9230
 9240
 9250
 9260
 9270
 9280
 9290
 9300
 9310
 9320
 9330
 9340
 9350
 9360
 9370
 9380
 9390
 9400
 9410
 9420
 9430
 9440
 9450
 9460
 9470
 9480
 9490
 9500
 9510
 9520
 9530
 9540
 9550
 9560
 9570
 9580
 9590
 9600
 9610
 9620
 9630
 9640
 9650
 9660
 9670
 9680
 9690
 9700
 9710
 9720
 9730
 9740
 9750
 9760
 9770
 9780
 9790
 9800
 9810
 9820
 9830
 9840
 9850
 9860
 9870
 9880
 9890
 9900
 9910
 9920
 9930
 9940
 9950
 9960
 9970
 9980
 9990
 10000

NAME OF PROGRAM OR SEGMENT MOM2

```

MASTER MOMENTS
C MOMENTS OF X**N BETWEEN OSCILLATOR WAVE FUNCTIONS
C USING HARMONIC OSCILLATOR BASIS
DOUBLE PRECISION C,W,U,ANUM,P,Q,Z
COMMON /SS/ C(150),W(150),ANUM(150),Q(5),V(6,100),TITLE(4),IEN(6),
1EN(6),P,1,NSETS,NM
ND=150
1 READ(5,100)NSETS,NMOM,TITLE
100 FORMAT(214,4A8)
IF(NSETS.LE.0)STOP
WRITE(6,200)TITLE
200 FORMAT(1H0,4A8)
DO 35 J=1,ND
TEMP=FLOAT(J)
U=DBLE(TEMP)
ANUM(J)=DSQRT(U)
35 READ(5,102)IEN(1),NM,A0,A2,A4,A6,EN(1)
102 FORMAT(214,4F10.5,E16.8)
IF(NM.EQ.0)STOP
READ(5,103)(V(1,J),J=1,NM)
103 FORMAT(5E16.8)
WRITE(6,201)IEN(1),NM,A0
201 FORMAT('0STATE',I4,' LENGTH OF VECTOR=',I4,' EXPONENT=',F10.5)
WRITE(6,202)A2,A4,A6
202 FORMAT('0HAMILTONIAN=P**2+',F10.5,'*X**2+',F10.5,'*X**4+',F10.5,'*
1X**6')
WRITE(6,203)IEN(1),EN(1)
203 FORMAT('0STATE',I4,' ENERGY*2=EIGENVALUE=',E16.8)
WRITE(6,204)
204 FORMAT('0VECTOR')
WRITE(6,205)(V(1,K),K=1,NM)
205 FORMAT(1H ,5E16.8)

```

MOM2	10
MOM2	20
MOM2	30
MOM2	40
MOM2	50
MOM2	60
MOM2	70
MOM2	80
MOM2	90
MOM2	100
MOM2	110
MOM2	120
MOM2	130
MOM2	140
MOM2	150
MOM2	160
MOM2	170
MOM2	180
MOM2	190
MOM2	200
MOM2	210
MOM2	220
MOM2	230
MOM2	240
MOM2	250
MOM2	260
MOM2	270
MOM2	280
MOM2	290
MOM2	300
MOM2	310
MOM2	320

NAME OF PROGRAM OR SEGMENT MOM2

```

51      IF(NSETS.LT.2)GO TO 51
52      DO 50 J=2,NSETS
53      READ(5,102)IEN(J),NL,B0,B2,B4,B6,EN(J)
54      IF(NL-NM)56,55,57
55      NL1=NL+1
56      DO 61 K=NL1,NM
57      V(J,K)=0.0
58      NL=NK
59      GO TO 55
60      NM1=NM+1
61      DO 62 K=NM1,NL
62      V(1,K)=0.0
63      NM=NL
64      IF(A0.NE.B0.OR.A2.NE.B2.OR.A4.NE.B4.OR.A6.NE.B6)GOTO 9MOM2 460
65      READ(5,103)(V(J,K),K=1,NM)
66      WRITE(6,203)IEN(J),EN(J)
67      WRITE(6,204)
68      WRITE(6,205)(V(J,K),K=1,NM)
69      CONTINUE
70      TEMP=0.5/A0
71      U=DBLE(TEMP)
72      U=DSQRT(U)
73      DO 52 J=1,NSETS
74      NC=NM
75      DO 53 K=1,NM
76      C(K)=DBLE(V(I,K))
77      WRITE(6,210)IEN(I)
78      FORMAT('OMATRIX ELEMENTS OF STATE',I4,' WITH STATES')
79      WRITE(6,214)(IEN(J),J=1,NSETS)
80      FORMAT('0POWER',10X,5(I4,16X))
81      NTOI=NSETS-I+1
82      DO 54 L=1,NMOM
83      MOM2 330
84      MOM2 340
85      MOM2 350
86      MOM2 360
87      MOM2 370
88      MOM2 380
89      MOM2 390
90      MOM2 400
91      MOM2 410
92      MOM2 420
93      MOM2 430
94      MOM2 440
95      MOM2 450
96      MOM2 460
97      MOM2 470
98      MOM2 480
99      MOM2 490
100     MOM2 500
101     MOM2 510
102     MOM2 520
103     MOM2 530
104     MOM2 540
105     MOM2 550
106     MOM2 560
107     MOM2 570
108     MOM2 580
109     MOM2 590
110     MOM2 600
111     MOM2 610
112     MOM2 620
113     MOM2 630
114     MOM2 640

```

NAME OF PROGRAM OR SEGMENT MOM2

	CALL SUM	MOM2 650
	L1=L-1	MOM2 660
212	WRITE(6,212)L1,(Q(1T),1T=1,NTOT)	MOM2 670
	FORMAT(1H ,14,5D20.10)	MOM2 680
54	CALL HERX(ND,NC,C,W,U,ANUM)	MOM2 690
	CONTINUE	MOM2 700
	CALL SUM	MOM2 710
	L1=NMOM	MOM2 720
52	WRITE(6,212)L1,(Q(1T),1T=1,NTOT)	MOM2 730
	CONTINUE	MOM2 740
	GO TO 1	MOM2 750
9	WRITE(6,211)	MOM2 760
211	FORMAT('OERROR - DIFFERENT HAMILTONIANS')	MOM2 770
	GO TO 1	MOM2 780
	END	MOM2 790

NAME OF PROGRAM OR SEGMENT HERX

```

SUBROUTINE HERX (ND,N,C,W,U,ANUM)
C X TIMES WAVE FUNCTION REPRESENTED IN HARMONIC OSCILLATOR BASIS
DOUBLE PRECISION C(ND),W(ND),ANUM(ND),U
IF(ND-N-1) 11,11,10
NP1=N+1
NM1=N-1
IF(N=2)9,8,7
DO 6 J=2,NM1
W(J)=U*(C(J+1)*ANUM(J)+C(J-1)*ANUM(J-1))
W(1)=C(2)*U
W(N)=ANUM(NM1)*U+C(NM1)
W(NP1)=ANUM(N)*U+C(N)
DO 5 J=1,NP1
C(J)=W(J)
N=N+1
RETURN
11 WRITE (6,100)
100 FORMAT('0DIMENSIONS TOO SMALL')
STOP
END

```

HERX 10
HERX 20
HERX 30
HERX 40
HERX 50
HERX 60
HERX 70
HERX 80
HERX 90
HERX 100
HERX 110
HERX 120
HERX 130
HERX 140
HERX 150
HERX 160
HERX 170
HERX 180
HERX 190
HERX 200

NAME OF PROGRAM OR SEGMENT SUM

	SUBROUTINE SUM		SUM	
C SUMMATIONS = ESSENTIALLY OVERLAPS OF VECTORS			SUM	10
DOUBLE PRECISION C,W,U,ANUM,P,Q,Z			SUM	20
COMMON /SS/ C(150),W(150),ANUM(150),Q(5),V(6,100),TITLE(4),IEN(6),SUM			SUM	30
IEN(6),P,I,NSETS,NM			SUM	40
DO 10 J=1,NSETS			SUM	50
P=0.0D0			SUM	60
DO 13 K=1,NM			SUM	70
P=P+DBLE(V(J,K))*C(K)			SUM	80
Q(J)=P			SUM	90
CONTINUE			SUM	100
RETURN			SUM	110
END			SUM	120
FINISH			SUM	130
			SUM	140

NAME OF PROGRAM OR SEGMENT ANOS

```

MASTER ANOSC
C WAVE FUNCTIONS FOR THE ANHARMONIC OSCILLATOR
DOUBLE PRECISION T,U
DIMENSION X(101),H(50,50),V(50,50),A(50),B(50),E(50),W1(50),W2(50)
DIMENSION W3(50),W4(50),VA(50),ZY(50)
1 READ(5,100)NMAX,A0,A2,A4,A6
100 FORMAT(14,4F10.5)
IF(NMAX.LE.0)STOP
WRITE(6,200)NMAX
200 FORMAT('0MAXIMUM VECTOR SIZE=',14)
A02=A0*A0
WRITE(6,201)A02
201 FORMAT('0EIGENVECTORS IN TERMS OF EVECS OF H=P*2+',F10.5,'*X**2')
202 WRITE(6,202)A2,A4,A6
FORMAT('0HAMILTONIAN=P*2+',F10.5,'*X**2+',F10.5,'*X**4+',F10.5,'*X**6')
1X**6')
N=NMAX/2
SA=A0
CR=0.5*(A2-A02)/SA
CC=0.25*A4/A02
CD=0.125*A6/(A02*SA)
204 FORMAT(1H,4E20.10)
READ(5,101)L,M
101 FORMAT(214)
C IP ODD, ODD PARITY WAVE FUNCTIONS , IP EVEN, EVEN FUNCTIONS
DO 10 IP=1,2
P=FLOAT(IP)
DO 11 I=1,N
R=2.*FLOAT(I)-P
H(I,I)=(2.*R+1.)*(SA+CB+3.*CC)+6.*R*R*(CC+5.*CD)+5.*CD*((4.*R*R+8.
1)*R+3.)
IF(N-I)11,11,21
11

```

ANOS 10

ANOS 20

ANOS 30

ANOS 40

ANOS 50

ANOS 60

ANOS 70

ANOS 80

ANOS 90

ANOS 100

ANOS 110

ANOS 120

ANOS 130

ANOS 140

ANOS 150

ANOS 160

ANOS 170

ANOS 180

ANOS 190

ANOS 200

ANOS 210

ANOS 220

ANOS 230

ANOS 240

ANOS 250

ANOS 260

ANOS 270

ANOS 280

ANOS 290

ANOS 300

ANOS 310

ANOS 320

NAME OF PROGRAM OR SEGMENT ANOS

21	T=DBLE(R+1.)*DBLE(R+2.)	ANOS 330
	U=DSQRT(T)	ANOS 340
	H(I,I+1)=U*(CB+2.*CC*(2.*R+3.))+15.*CD*((R+3.)*R+3.)	ANOS 350
	H(I+1,I)=H(I,I+1)	ANOS 360
	IF(N-I-1)11,11,22	ANOS 370
22	T=T*DBLE(R+3.)*DBLE(R+4.)	ANOS 380
	U=DSQRT(T)	ANOS 390
	H(I,I+2)=U*(CC+3.*CD*(2.*R+5.))	ANOS 400
	H(I+2,I)=H(I,I+2)	ANOS 410
	IF(N-I-2)11,11,23	ANOS 420
23	T=T*DBLE(R+5.)*DBLE(R+6.)	ANOS 430
	U=DSQRT(T)	ANOS 440
	H(I,I+3)=CD*U	ANOS 450
	H(I+3,I)=CD*U	ANOS 460
	IF(N-I-3)11,11,24	ANOS 470
24	I4=I+4	ANOS 480
	DO 14 J=I4,N	ANOS 490
	H(I,J)=0.0	ANOS 500
14	H(J,I)=0.0	ANOS 510
11	CONTINUE	ANOS 520
	NT=2	ANOS 530
	P1=0.5E-10	ANOS 540
	P2=1.0E-06	ANOS 550
	RL=1.0E-10	ANOS 560
4445	CALL JNH(N,50,L,M,H,E,V,A,B,VA,W1,W2,W3,W4,BG,P1,P2,KZ,NT,RL)	ANOS 570
	IF(FAIL.EQ.1)WRITE(6,4445)	ANOS 580
	FORMAT('FAILUREI?I?I?I?I?I?')	ANOS 590
	DO 12 I=L,M	ANOS 600
	IQ=1-IP	ANOS 610
	DO 13 J=1,N	ANOS 620
	K=2+J+IQ	ANOS 630
13	X(K)=V(J,I)	ANOS 640

NAME OF PROGRAM OR SEGMENT ANOS

```

10=IP-2
DO 15 J=1,N
K=2*J+1Q
X(K)=0.0
EGG=E(I)
WRITE(6,203)I,EGG,A0
FORMAT('OVECTOR',I4,' EIGENVALUE',E20.10,'EXPONENT',F10.5)
203 EGH=0.5*EGG
WRITE(6,4444)EGH
FORMAT(' HALF EIGENVALUE=',E20.10)
4444 NMAX=2*N+1-IP
WRITE(6,204)(X(J),J=1,NMAX)
IV=2*1-IP
WRITE(7,209 )IV,NMAX,A0,A2,A4,A6,EGG
WRITE(7,210)(X(J),J=1,NMAX)
210 FORMAT(5E16.8)
209 FORMAT(2I4,4F10.5,E16.8)
12 CONTINUE
10 CONTINUE
GO TO 1
END

```

ANOS 650
ANOS 660
ANOS 670
ANOS 680
ANOS 690
ANOS 700
ANOS 710
ANOS 720
ANOS 730
ANOS 740
ANOS 750
ANOS 760
ANOS 770
ANOS 780
ANOS 790
ANOS 800
ANOS 810
ANOS 820
ANOS 830
ANOS 840
ANOS 850

NAME OF PROGRAM OR SEGMENT OVER

	MASTER OVERLAPS	OVER 10
C	OVERLAP CALCULATIONS	OVER 20
	DOUBLE PRECISION S	OVER 30
	DIMENSION V(20,20),IS(20),X(20)	OVER 40
7	I=0	OVER 50
6	I=I+1	OVER 60
1	READ(5,100)IS(I),N,EX	OVER 70
100	FORMAT(2I4,F10.5)	OVER 80
	IF(I=1)3,3,4	OVER 90
3	N1=N	OVER 100
	EX1=EX	OVER 110
	GO TO 9	OVER 120
4	IF(N,NE,N1)WRITE(6,202)	OVER 130
202	FORMAT('0WARNING - DIFFERENT LENGTHS')	OVER 140
	IF(EX,NE,EX1)WRITE(6,200)	OVER 150
200	FORMAT('0DIFFERENT EXPONENT PARAMETERS')	OVER 160
9	IF(N,LT,0)STOP	OVER 170
	IF(N)5,5,8	OVER 180
8	READ(5,101)(V(I,J),J=1,N)	OVER 190
101	FORMAT(5E16.8)	OVER 200
	IF(N,GE,N1)GO TO 6	OVER 210
	L=N+1	OVER 220
	DO 16 J=L,N1	OVER 230
16	V(I,J)=0.0	OVER 240
	GO TO 6	OVER 250
5	IMAX=I-1	OVER 260
	N=N1	OVER 270
	EX=EX1	OVER 280
	WRITE(6,203)N	OVER 290
203	FORMAT('0VECTORS OF LENGTH ',I4)	OVER 300
	DO 10 K=1,IMAX	OVER 310
	WRITE(6,204)IS(K)	OVER 320

NAME OF PROGRAM OR SEGMENT OVER

```

204  FORMAT('OVECTOR FOR STATE',I4)          OVER 330
10   WRITE(6,205)(V(K,J),J=1,N)              OVER 340
205  FORMAT(1H,5E16.8)                       OVER 350
11   READ(5,100)JS,M,EX1                    OVER 360
      IF(M.EQ.0)GO TO 7                      OVER 370
      IF(EX.NE.EX1)WRITE(6,200)              OVER 380
      READ(5,101)(X(J),J=1,M)                OVER 390
      WRITE(6,211)JS,M                      OVER 400
211  FORMAT('OVECTOR APPROXIMATING STATE',I4,' LENGTH',I4) OVER 410
      WRITE(6,205)(X(J),J=1,M)              OVER 420
      L=MIN0(N,M)                           OVER 430
      DO 20 K=1,IMAX                        OVER 440
        S=0.0D0                             OVER 450
        DO 21 J=1,L                         OVER 460
          S=S+DBLE(V(K,J))*DBLE(X(J))        OVER 470
21   WRITE(6,212)IS(K),S                   OVER 480
20   FORMAT(' OVERLAP WITH STATE',I4,' = ',D20.10) OVER 490
212  GO TO 11                              OVER 500
      END                                  OVER 510
      FINISH                              OVER 520

```

NAME OF PROGRAM OR SEGMENT INVE

```

MASTER INVE
C THIS PROGRAM PERFORMS AN INVERSE INTERPOLATION ON DATA OUTPUTTED
C FROM HYPERVERSION 3
C REFERENCE ABRAMOWITZ AND STEGAN PG. 882
C JCNASH MATHS INSTITUTE FEBRUARY 1970
C METHOD INVERSION OF EVERETS INTERPOLATION SERIES
C DIMENSION TOP(5),C(5),D3(10)
C DIMENSION P(15),F(15),D1(14),D2(13),D4(11),D5(10),D6(9)
1 READ(5,100)(TOP(I),I=1,5),NSETS
100 FORMAT(5A8,14)
WRITE(6,101)(TOP(I),I=1,5),NSETS
101 FORMAT(1H0,5A8,14)
IF(NSETS.EQ.0)GO TO 1001
READ(5,102)(P(I),F(I),I=1,NSETS)
102 FORMAT(2E16.8)
WRITE(6,104)
104 FORMAT(9H0 P(J),10X,4HF(J))
WRITE(6,105)(P(J),F(J),J=1,NSETS)
105 FORMAT(1H ,2E16.8)
H=P(2)-P(1)
WRITE(6,103)H
103 FORMAT(10H0INTERVAL=,E16.8)
NO=NSETS-1
IF(NO.EQ.0)GO TO 30
DO 20 J=1,NO
20 D1(J)=F(J+1)-F(J)
WRITE(6,106)
106 FORMAT(18H0FIRST DIFFERENCES)
WRITE(6,107)(D1(J),J=1,NO)
107 FORMAT(1H ,5E16.8)
NO=NO-1
IF(NO.EQ.0)GO TO 30
INVE 10
INVE 20
INVE 30
INVE 40
INVE 50
INVE 60
INVE 70
INVE 80
INVE 90
INVE 100
INVE 110
INVE 120
INVE 130
INVE 140
INVE 150
INVE 160
INVE 170
INVE 180
INVE 190
INVE 200
INVE 210
INVE 220
INVE 230
INVE 240
INVE 250
INVE 260
INVE 270
INVE 280
INVE 290
INVE 300
INVE 310
INVE 320

```

NAME OF PROGRAM OR SEGMENT INVE

```

21      DO 21 J=1,NO
          D2(J)=D1(J+1)-D1(J)
          WRITE(6,108)
108      FORMAT(19H0SECOND DIFFERENCES)
          WRITE(6,107)(D2(J),J=1,NO)
          NO=NO-1
          IF(NO.EQ.0)GO TO 30
          DO 22 J=1,NO
22      D3(J)=D2(J+1)-D2(J)
          WRITE(6,109)
          WRITE(6,107)(D3(J),J=1,NO)
          FORMAT(18H0THIRD DIFFERENCES)
109      NO=NO-1
          IF(NO.EQ.0)GO TO 30
          DO 23 J=1,NO
23      D4(J)=D3(J+1)-D3(J)
          FORMAT(19H0FOURTH DIFFERENCES)
110      WRITE(6,110)
          WRITE(6,107)(D4(J),J=1,NO)
          NO=NO-1
          IF(NO.EQ.0)GO TO 30
          DO 24 J=1,NO
24      D5(J)=D4(J+1)-D4(J)
          WRITE(6,111)
          FORMAT(18H0FIFTH DIFFERENCES)
111      WRITE(6,107)(D5(J),J=1,NO)
          NO=NO-1
          IF(NO.EQ.0)GO TO 30
          DO 25 J=1,NO
25      D6(J)=D5(J+1)-D5(J)
          FORMAT(18H0SIXTH DIFFERENCES)
112      WRITE(6,112)

```

```

INVE 330
INVE 340
INVE 350
INVE 360
INVE 370
INVE 380
INVE 390
INVE 400
INVE 410
INVE 420
INVE 430
INVE 440
INVE 450
INVE 460
INVE 470
INVE 480
INVE 490
INVE 500
INVE 510
INVE 520
INVE 530
INVE 540
INVE 550
INVE 560
INVE 570
INVE 580
INVE 590
INVE 600
INVE 610
INVE 620
INVE 630
INVE 640

```

NAME OF PROGRAM OR SEGMENT INVE

```

30      WRITE(6,107)(D6(J),J=1,NO)
        IF(NSETS.LT.6)GO TO 1
        NUT=NSETS/2
        SH=D1(NUT)
        SIS=D2(NUT)
        SOS=D2(NUT-1)
        SIF=D4(NUT-1)
        SOF=D4(NUT-2)
        A0=F(NUT)
        A1=SH-SOS/3.-SIS/6.+SOF/20.+SIF/30.
        A2=SOS/2.-SOF/24.
        A3=(SIS-SOS)/6.-(SOF+SIF)/24.
        A4=SOF/24.
        A5=(SIF-SOF)/120.
        WRITE(6,113)
        WRITE(6,107)A1,A2,A3,A4,A5
113     FORMAT(15H0A1 A2 A3 A4 A5)
        C(1)=-A0/A1
        C(2)=-A2/A1
        C(3)=(2.0*A2*A2-A3*A1)/(A1*A1)
        C(4)=(5.0*A2*(A3*A1-A2*A2)-A4*A1*A1)/(A1*A1*A1)
        C(5)=6.0*A2*A4/(A1*A1)-A5/A1+3.0*A3*A3/(A1*A1)-21.0*A2*A2*A3/(A1*A1*A1)
114     FORMAT(9H0C VALUES)
        WRITE(6,114)
        WRITE(6,107)(C(I),I=1,5)
115     FORMAT(26H0SUCCESSIVE APPROXIMATIONS)
        WRITE(6,115)
        SIF=P(NUT)
        SUM=0.0
        QUIP=C(1)
        DO 35 J=1,5

```

INVE 650

INVE 660

INVE 670

INVE 680

INVE 690

INVE 700

INVE 710

INVE 720

INVE 730

INVE 740

INVE 750

INVE 760

INVE 770

INVE 780

INVE 790

INVE 800

INVE 810

INVE 820

INVE 830

INVE 840

INVE 850

INVE 860

INVE 870

INVE 880

INVE 890

INVE 900

INVE 910

INVE 920

INVE 930

INVE 940

INVE 950

INVE 960

NAME OF PROGRAM OR SEGMENT INVE

	DO 36 I=2,5	INVE 970
36	C(I)=C(I)*QUIP	INVE 980
	SUM=SUM+C(J)	INVE 990
	SAM=SUM*H	INVE1000
	TRY=STIFF+SAM	INVE1010
35	WRITE(6,116)SUM,SAM,TRY	INVE1020
116	FORMAT(12H0PROPORTION=,F16.8,5HCORR=,F16.8,6HVALUE=,F16.8)	INVE1030
	GO TO 1	INVE1040
1001	STOP	INVE1050
	END	INVE1060

NAME OF PROGRAM OR SEGMENT OSC5

```

MASTER OSC5
DOUBLE PRECISION P,Q
REAL KCA,KCB,KCS,KCX,KCP
DIMENSION A(40,40),B(40,40),X(40),Y(40),CI(40),W1(40),W2(40)
DIMENSION W3(40),W4(40),W5(40),AP(40,40),BP(40,40)
COMMON KCA,KCS,KCP,KCX,KCB
NA=40
NOUER=40
READ(5,99)KCA,KCS,KCP,KCX,KCB
99  FORMAT(5A1)
1  READ(5,100)N,NIN,A0,A2,A4,A6,AU
100  FORMAT(2I4,5F10.5)
      READ(5,214)
      WRITE(6,214)
214  FORMAT('
IF(N.EQ.0)STOP
WRITE(6,201)A0,A2,A4,A6
201  FORMAT('OEXPONENT=',F10.5,' HAMILTONIAN=0.5*P**2+',F10.5,'*X**2+',
1F10.5,'*X**4+',F10.5,'*X**6')
      IF(AU.EQ.0.0)AU=A0
2  READ(5,102)ITYP,LUP,LREP,PORE
102  FORMAT(3I4,F10.5)
      READ(5,101)(CI(J),J=1,NIN)
101  FORMAT(5E16.8)
      WRITE(6,203)
203  FORMAT('OINPUT VECTOR')
      WRITE(6,204)(CI(J),J=1,NIN)
204  FORMAT(1H,5E16.8)
      IF(ITYP)10,11,10
11  DO 111 J=1,NIN
111  X(J)=CI(J)
      GO TO 12
      OSC5 10
      OSC5 20
      OSC5 30
      OSC5 40
      OSC5 50
      OSC5 60
      OSC5 70
      OSC5 80
      OSC5 90
      OSC5 100
      OSC5 110
      OSC5 120
      OSC5 130
      OSC5 140
      OSC5 150
      OSC5 160
      OSC5 170
      OSC5 180
      OSC5 190
      OSC5 200
      OSC5 210
      OSC5 220
      OSC5 230
      OSC5 240
      OSC5 250
      OSC5 260
      OSC5 270
      OSC5 280
      OSC5 290
      OSC5 300
      OSC5 310
      OSC5 320

```

NAME OF PROGRAM OR SEGMENT OSC5

```

C TRANSFORMATION OF BASIS - HARMONIC OSCILLATOR TO POWER
10 CALL HP(NIN,CI,X,A0,0,W1,W2,W3)
12 WRITE(6,205)
205 FORMAT('OPOLYNOMIAL BASIS INPUT VECTOR')
WRITE(6,204)(X(J),J=1,NIN)
IF(NIN=N)3,28,28
3 READ(5,103)M,MH,SUSP
103 FORMAT(2I4,F10.5)
IF(MH.LT.M)STOP
IMP=0

C CALCULATE ENERGY - INITIAL STATE
CALL ENER(NA,NIN,A0,A2,A4,A6,AN,EN,X)
6 WRITE(6,209)SUSP
209 FORMAT('OSUSPECTED ROOT AT ',E16.8)
C FORM MATRICES P AND Q
4 CALL MAKE(NA,M,MH,X,A,B,A0,AU,AZ,A4,A6,NIN,CI,W1,W2,W3,W4,Y)
C SUSP=GUESS FOR EIGENVALUE
C SOLUTION OF MATRIX EIGENVALUE PROBLEM
IF(SUSP.NE.0.0)CALL ADJ(NA,M,A,B,SUSP)
CALL REDN(NA,M,A,B,IFAIL)
IF(IFAIL.EQ.1)GO TO 30
C INITIALIZE VECTOR
ANM=1.0/FLOAT(M)
DO 14 J=1,M
14 X(J)=ANM
TOL=1.0E-10
LIM=100
EV=SUSP
CALL ITN(NA,M,TOL,LIM,A,B,X,EV,Y)
32 WRITE(6,212)EV,LIM
212 FORMAT('OFINAL VALUE ',E16.8,' AT ',I4,' ITERATIONS')
C CALCULATE ENERGY -- FINAL STATE

```

```

OSC5 330
OSC5 340
OSC5 350
OSC5 360
OSC5 370
OSC5 380
OSC5 390
OSC5 400
OSC5 410
OSC5 420
OSC5 430
OSC5 440
OSC5 450
OSC5 460
OSC5 470
OSC5 480
OSC5 490
OSC5 500
OSC5 510
OSC5 520
OSC5 530
OSC5 540
OSC5 550
OSC5 560
OSC5 570
OSC5 580
OSC5 590
OSC5 600
OSC5 610
OSC5 620
OSC5 630
OSC5 640

```

NAME OF PROGRAM OR SEGMENT OSCS

```

CALL ENER(NA,M,AU,A2,A4,A6,AM,EM,X)
EP=EV+EN
216 WRITE(6,216)EP,EM
    FORMAT('CALCULATED ENERGY=',E16.8,' VECTOR ENERGY=',E16.8)
    Q=1.0/SQRT(AM)
    DO 18 K=1,M
18   X(K)=X(K)*Q
    WRITE(6,210)
210  FORMAT('FINAL VECTOR')
    WRITE(6,204)(X(I),I=1,M)
    IF(MH.EQ.M)GO TO 40
    M1=M+1
    DO 41 I=M1,MH
    P=0.0D0
    Q=0.0
    DO 42 J=1,M
    Q=Q+A(I,J)*X(J)
    P=P+B(I,J)*X(J)
42   WRITE(6,333)I,Q,P
333  FORMAT('MATRIX ELEMENTS FOR OPERATOR ',I4,' COMMUTATOR',D16.8,'
      1 ELEMENT',D16.8)
    Q=Q-EV*P
41   WRITE(6,223)I,Q
223  FORMAT(' EXTRA ROW VALUE FOR ROW',I4,' = ',D16.8)
    CALL QUER(NQUER,M,MH,X,AP,BP,AQ,AU,A2,A4,A6,NIN,C1,W1,W2,W3,W4,W5,OSCS 890
      1Y,EV)
C TRANSFORMATION OF BASIS - POWER TO HARMONIC OSCILLATOR
    CALL HP(M,Y,X,AU,1,W1,W2,W3)
    Q=0.0
    DO 15 J=1,M
15   Q=Q+Y(J)**2
    WRITE(6,222)Q
OSCS 650
OSCS 660
OSCS 670
OSCS 680
OSCS 690
OSCS 700
OSCS 710
OSCS 720
OSCS 730
OSCS 740
OSCS 750
OSCS 760
OSCS 770
OSCS 780
OSCS 790
OSCS 800
OSCS 810
OSCS 820
OSCS 830
OSCS 840
OSCS 850
OSCS 860
OSCS 870
OSCS 880
OSCS 890
OSCS 900
OSCS 910
OSCS 920
OSCS 930
OSCS 940
OSCS 950
OSCS 960

```

NAME OF PROGRAM OR SEGMENT OSC5

```

222  FORMAT('ONORM CHECK',E16.8)
40   WRITE(6,211)IMP,AU
211  FORMAT('0EUCCLIDEAN NORM VECTOR AT JUMP',I4,' EXPONENT=',F10.5)
      WRITE(6,204)(Y(J),J=1,M)
      IF(LREP)21,0,21
      IMP=IMP+1
      IF(LUP-IMP)1,1,0
      SUSP=- (EV*PORE+(1.0-PORE)*SUSP)
      Q=AU
      AO=AU
      AU=Q
      GO TO 4
C ANOTHER ROOT?
21   READ(5,104)SUSY,ISTOP
104  FORMAT(F10.5,I2)
      IF(ISTOP)0,1,0
      WRITE(6,220)SUSY
220  FORMAT(' ANOTHER ROOT SUSPECTED A ',E16.8)
      DELT=SUSY-SUSP
      WRITE(6,221)DELT
221  FORMAT(' P=P-DELT+Q   DELT=',E16.8)
      CALL ADJ(NA,M,A,B,DELT)
      CALL REDN(NA,M,A,B,IFAIL)
      IF(IFAIL.EQ.1)GO TO 21
      LIM=50
      EV=SUSY
      SUSP=SUSY
      DO 33 J=1,M
33   X(J)=ANM
      CALL ITN(NA,M,TOL,LIM,A,B,X,EV,Y)
      GO TO 32
30   IF(LREP)21,1,21

```

OSC5 970
 OSC5 980
 OSC5 990
 OSC51000
 OSC51010
 OSC51020
 OSC51030
 OSC51040
 OSC51050
 OSC51060
 OSC51070
 OSC51080
 OSC51090
 OSC51100
 OSC51110
 OSC51120
 OSC51130
 OSC51140
 OSC51150
 OSC51160
 OSC51170
 OSC51180
 OSC51190
 OSC51200
 OSC51210
 OSC51220
 OSC51230
 OSC51240
 OSC51250
 OSC51260
 OSC51270
 OSC51280

NAME OF PROGRAM OR SEGMENT OSC5

28 WRITE(6,215)
215 FORMAT('INPUT VECTOR TOO BIG')

STOP
END

OSC51290
OSC51300
OSC51310
OSC51320

NAME OF PROGRAM OR SEGMENT HOME

```

SUBROUTINE HOME(N,M,A0,A2,A4,A6,X,T,U,V)
C FORMS HAMILTONIAN * WAVE FUNCTION REPRESENTED BY POWER OF X BASIS
C FUNCTION COEFFICIENTS IN X, AND PUT RESULT IN V
C X IS INPUT, V IS OUTPUT, T,U, TEMP RARY
  DIMENSION X(N),T(N),U(N),V(N)
  DO 12 J=1,N
12  V(J)=0.0
    IF(N=7,LT,M)GO TO 110
    CALL FMOVE(X,T,M)
    L=M
    T(L+1)=0.0
    CALL PEA(N,L,T,U,A0)
    CALL PEA(N,L,T,U,A0)
    DO 10 J=1,L
10  V(J)=-0.5*T(J)
      DO 17 KK=1,M
17  T(KK)=X(KK)
        L=M
        T(L+1)=0.0
        CALL EKS(N,L,T)
        CALL EKS(N,L,T)
        DO 13 J=1,L
13  V(J)=V(J)+A2*T(J)
          CALL EKS(N,L,T)
          CALL EKS(N,L,T)
          DO 14 J=1,L
14  V(J)=V(J)+A4*T(J)
            CALL EKS(N,L,T)
            CALL EKS(N,L,T)
            DO 16 J=1,L
16  V(J)=V(J)+A6*T(J)
              M=L

```

HOME	10
HOME	20
HOME	30
HOME	40
HOME	50
HOME	60
HOME	70
HOME	80
HOME	90
HOME	100
HOME	110
HOME	120
HOME	130
HOME	140
HOME	150
HOME	160
HOME	170
HOME	180
HOME	190
HOME	200
HOME	210
HOME	220
HOME	230
HOME	240
HOME	250
HOME	260
HOME	270
HOME	280
HOME	290
HOME	300
HOME	310
HOME	320

NAME OF PROGRAM OR SEGMENT HOME

```
110 RETURN  
111 WRITE(6,111)M  
111 FORMAT('0M TOO BIG=',I4)  
RETURN  
END
```

```
HOME 330  
HOME 340  
HOME 350  
HOME 360  
HOME 370
```

NAME OF PROGRAM OR SEGMENT EKS

	SUBROUTINE EKS(N,L,C)	EKS 10
C	FORMS X * WAVE FUNCTION	EKS 20
	DIMENSION C(N)	EKS 30
1	IF(L+1-N)1,22,22	EKS 40
1	IF(C(L+1))20,2,20	EKS 50
2	DO 10 J=1,L	EKS 60
	K=L-J+1	EKS 70
10	C(K+1)=C(K)	EKS 80
	C(1)=0.0	EKS 90
	L=L+1	EKS 100
	C(L+1)=0.0	EKS 110
	RETURN	EKS 120
20	WRITE(6,101)L	EKS 130
101	FORMAT('OERRORI SEGMENT EKS - ELEMENT C('',14,''+1) NOT ZERO')	EKS 140
	STOP 20	EKS 150
22	WRITE(6,102)L,N	EKS 160
102	FORMAT('ODIMENSIONING ERROR IN EKS L='',14,' ' N='',14)	EKS 170
	STOP 21	EKS 180
	END	EKS 190

NAME OF PROGRAM OR SEGMENT PEA

```

SUBROUTINE PEA(N,L,C,D,A0)
C FORMS D/DX * WAVE FUNCTION
DIMENSION C(N),D(N)
1 IF(L+1=N)1,22,22
2 IF(C(L+1))20,2,20
13 IF(L=1)11,11,13
DO 10 J=2,L
D(J)=C(J+1)*FLOAT(J)-A0+C(J-1)
10 CONTINUE
D(1)=C(2)
11 D(L+1)=-A0+C(L)
L=L+1
CALL FMOVE(D,C,L)
C(L+1)=0.0
RETURN
20 WRITE(6,101)L
101 FORMAT('OERRORI SEGMENT PEA - ELEMENT C('',I4,''+1) NOT ZERO')
1000 WRITE(6,1000)C
FORMAT(1H,5E16.8)
STOP 30
22 WRITE(6,102)L,N
102 FORMAT('ODIMENSIONING ERROR IN PEA L='',I4,' ' N='',I4)
STOP 31
END

```

Line	Statement	PEA	Count
1	IF(L+1=N)1,22,22	PEA	10
2	IF(C(L+1))20,2,20	PEA	20
13	IF(L=1)11,11,13	PEA	30
DO 10 J=2,L		PEA	40
D(J)=C(J+1)*FLOAT(J)-A0+C(J-1)		PEA	50
10 CONTINUE		PEA	60
D(1)=C(2)		PEA	70
11 D(L+1)=-A0+C(L)		PEA	80
L=L+1		PEA	90
CALL FMOVE(D,C,L)		PEA	100
C(L+1)=0.0		PEA	110
RETURN		PEA	120
20 WRITE(6,101)L		PEA	130
101 FORMAT('OERRORI SEGMENT PEA - ELEMENT C('',I4,''+1) NOT ZERO')		PEA	140
1000 WRITE(6,1000)C		PEA	150
FORMAT(1H,5E16.8)		PEA	160
STOP 30		PEA	170
22 WRITE(6,102)L,N		PEA	180
102 FORMAT('ODIMENSIONING ERROR IN PEA L='',I4,' ' N='',I4)		PEA	190
STOP 31		PEA	200
END		PEA	210

NAME OF PROGRAM OR SEGMENT QUER

```

SUBROUTINE QUER(N,NIN,MH,X,A,B,A0,A2,A4,A6,C1,W1,W2,W3,W4,Y)
C DIAGONAL HYPERVIRIAL TESTS
REAL KCA,KCB,KCS,KCX,KCP
DIMENSION A(N,N),B(N,N),X(N),CI(N),W1(N),W2(N),W3(N),W4(N),Y(N)
COMMON KCA,KCS,KCP,KCX,KCB
IF(MH)30,30,10
10 CALL MAKE(N,NIN,MH,X,A,B,A0,A0,A2,A4,A6,NIN,C1,W1,W2,W3,W4,Y)
DO 11 I=1,MH
P=0.0
Q=0.0
DO 12 J=1,NIN
P=P+A(I,J)*X(J)
Q=Q+B(I,J)*X(J)
12 WRITE(6,151)I,P
151 FORMAT('0DIAGONAL HYPERVIRIAL OF OPERATOR ',I4,' = ',E16.8)
11 WRITE(6,152)Q
152 FORMAT(' EXPECTATION VALUE=',E16.8)
30 CALL PLUG(N,NIN,A0,X,33,0.25,0.0)
RETURN
END

```

QUER 10
 QUER 20
 QUER 30
 QUER 40
 QUER 50
 QUER 60
 QUER 70
 QUER 80
 QUER 90
 QUER 100
 QUER 110
 QUER 120
 QUER 130
 QUER 140
 QUER 150
 QUER 160
 QUER 170
 QUER 180
 QUER 190
 QUER 200

NAME OF PROGRAM OR SEGMENT PLUG

```

SUBROUTINE PLUG(N,M,AU,X,NP,HI,A)
C PLOTTING DATA FOR WAVE FUNCTION REPRESENTED IN POWER TYPE BASIS
DIMENSION X(N)
V=A
IP=0
WRITE(6,100)AU
100 FORMAT('O PLOT FOR FUNCTION WITH EXPONENT=' ,F10.5)
WRITE(6,101)(X(J),J=1,M)
101 FORMAT(' VECTOR ',5E16.8)
WRITE(6,103)
103 FORMAT('O X',7X,'EXP0',5X,'POLY',4X,'FN')
G=-0.5*AU
1 E=EXP(G*V*V)
P=X(M)
L=M-1
2 IF(L)3,3,4
4 P=P*V+X(L)
L=L-1
GO TO 2
3 F=P*E
WRITE(6,102)V,E,P,F
102 FORMAT(1H ,2X,F6.3,2X,F6.3,2X,F9.3,2X,F6.3)
IP=IP+1
V=V+HI
IF(IP=NP)1,1,5
5 RETURN
END

```

```

PLUG 10
PLUG 20
PLUG 30
PLUG 40
PLUG 50
PLUG 60
PLUG 70
PLUG 80
PLUG 90
PLUG 100
PLUG 110
PLUG 120
PLUG 130
PLUG 140
PLUG 150
PLUG 160
PLUG 170
PLUG 180
PLUG 190
PLUG 200
PLUG 210
PLUG 220
PLUG 230
PLUG 240
PLUG 250
PLUG 260
PLUG 270

```

NAME OF PROGRAM OR SEGMENT FN Z AND MAKE

DOUBLE PRECISION FUNCTION Z(M,A)
 C CALCULATES INTEGRAL OF $X**M * \exp(-A * X**2)$
 C RANGE MINUS INFINITY TO PLUS INFINITY
 C VALUE RETURNED IN Z (DOUBLE PRECISION)

Z=0.0D0
 IF(M.LT.0)RETURN
 IF(M=2*(M/2).NE.0)RETURN
 Z=1.772453850905516D0/SQRT(A)
 L=2
 1 IF(L.GT.M)RETURN
 Z=0.5D0*FLOAT(L-1)*Z/A
 L=L+2
 GO TO 1
 END

MAKE
 32

SUBROUTINE MAKE(N,M,MH,X,A,B,A0,AU,A2,A4,A6,NIN,S,T,U,V,W,Y)
 REAL KCA,KCB,KCS,KCX,KCP
 DOUBLE PRECISION Z
 DIMENSION X(N),A(N,N),B(N,N),S(N),T(N),U(N),V(N),W(N),Y(N),CA(40)
 COMMON KCA,KCS,KCP,KCX,KCB
 LD=NIN

C INITIAL VECTOR X REPRESENTING WAVE FUNCTION IN POWER TYPE BASIS
 C FORM H X IN Y

CALL HOME(N,LD,A0,A2,A4,A6,X,T,U,Y)
 AZ=0.5*(A0+AU)

DO 2 I=1,MH
 C W WILL STORE STORE OPERATOR * WAVE FUNCTION
 C V WILL STORE OPERATOR * HAMILTONIAN * WAVE FUNCTION
 DO 1 J=1,N

V(J)=0.0
 W(J)=0.0
 1

FN Z	10
FN Z	20
FN Z	30
FN Z	40
FN Z	50
FN Z	60
FN Z	70
FN Z	80
FN Z	90
FN Z	100
FN Z	110
FN Z	120
FN Z	130
FN Z	140
FN Z	150
FN Z	160
FN Z	170
FN Z	180
FN Z	190
FN Z	200
FN Z	210
FN Z	220
FN Z	230
FN Z	240
FN Z	250
FN Z	260
FN Z	270
FN Z	280
FN Z	290
FN Z	300
FN Z	310
FN Z	320

NAME OF PROGRAM OR SEGMENT ~~FN-2~~ MAKE

C MOVE COEFFICIENT VECTORS INTO WORK AREA AND INITIALIZE SUITABLY	FN Z 330
CALL FMOVE(X,T,NIN)	FN Z 340
CALL FMOVE(Y,U,LD)	FN Z 350
LP=LD	FN Z 360
L=NIN	FN Z 370
U(LD+1)=0.0	FN Z 380
T(L+1)=0.0	FN Z 390
C READ IN HYPERVIRIAL OPERATOR AS STRING OF CHARACTERS	FN Z 400
READ(5,300)CA	FN Z 410
FORMAT(40A1)	FN Z 420
WRITE(6,400)CA	FN Z 430
FORMAT('0HYPERVIRIAL OPERATOR ',40A1)	FN Z 440
IK=41	FN Z 450
C PROCEED BACKWARDS THROUGH CHARACTER STRING	FN Z 460
IK=IK-1	FN Z 470
C TERMINATE WHEN WE RUN OUT OF CHARACTERS?	FN Z 480
IF(IK)14,14,5	FN Z 490
C IGNORE RIGHT HAND BLANKS	FN Z 500
IF(CA(IK)-KCB)6,3,6	FN Z 510
C IS CHARACTER INSTRUCTING A D/DX OPERATION?	FN Z 520
IF(CA(IK)-KCP)7,4,7	FN Z 530
C D/DX ON X VECTOR I.E. WAVE FUNCTION	FN Z 540
CALL PEAC(N,L,T,S,A0)	FN Z 550
C D/DX ON H X I.E. ON H * WAVE FUNCTION	FN Z 560
CALL PEAC(N,LP,U,S,A0)	FN Z 570
GO TO 3	FN Z 580
C IS CHARACTER INSTRUCTING A MULTIPLICATION BY X OPERATION?	FN Z 590
IF(CA(IK)-KCX)8,9,8	FN Z 600
C MULTIPLY WAVE FUNCTION AND H* WAVE FUNCTION BY X	FN Z 610
CALL EKS(N,L,T)	FN Z 620
CALL EKS(N,LP,U)	FN Z 630
GO TO 3	FN Z 640

NAME OF PROGRAM OR SEGMENT ~~FN-2~~ MAKE

```

C IS CHARACTER A MINUS SIGN INSTRUCTION?
8   IF(CA(IK)-KCS)10,11,10
C PERFORM SUBTRACTIONS FROM V AND W
11  CALL VSUM(N,L,W,T,-1.0)
    CALL VSUM(N,LP,V,U,-1.0)
    GO TO 3
C IS CHARACTER A PLUS SIGN INSTRUCTION ?
10  IF(CA(IK)-KCA)12,14,12
C STOP IF OPERATOR NOT YET IDENTIFIED
12  WRITE(6,303)
303  FORMAT('UNIDENTIFIED OPERATOR')
    STOP
C PERFORM ADDITIONS INTO V AND W
14  CALL VSUM(N,L,W,T,1.0)
    CALL VSUM(N,LP,V,U,1.0)
C TERMINATE IF WE RUN OUT OF CHARACTERS?
    IF(IK)15,15,3
C FORM HAMILTONIAN * OPERATOR * WAVE FUNCTION IN S
15  CALL HOME(N,L,A0,A2,A4,A6,W,T,U,S)
C FORM H * OPERATOR * FN - OPERATOR * H * FN IN S
    CALL VSUM(N,L,S,V,-1.0)
C FORM MATRICES P AND Q
    DO 16 J=1,M
        F=0.0
        G=0.0
        DO 17 K=1,L
            KJ=K+J-2
            Z1=Z(KJ,AZ)
            F=F+S(K)*Z1
            G=G+W(K)*Z1
17      A(I,J)=F
16      B(I,J)=G
    FN Z 650
    FN Z 660
    FN Z 670
    FN Z 680
    FN Z 690
    FN Z 700
    FN Z 710
    FN Z 720
    FN Z 730
    FN Z 740
    FN Z 750
    FN Z 760
    FN Z 770
    FN Z 780
    FN Z 790
    FN Z 800
    FN Z 810
    FN Z 820
    FN Z 830
    FN Z 840
    FN Z 850
    FN Z 860
    FN Z 870
    FN Z 880
    FN Z 890
    FN Z 900
    FN Z 910
    FN Z 920
    FN Z 930
    FN Z 940
    FN Z 950
    FN Z 960

```

NAME OF PROGRAM OR SEGMENT ~~FN-2~~ MAKE

2	CONTINUE	FN Z 970
	WRITE(6,101)MH,M	FN Z 980
101	FORMAT('0A BY ROWS ',I4,' ROWS',I4,' COLUMNS')	FN Z 990
	WRITE(6,120)((A(I,J),J=1,M),I=1,MH)	FN Z1000
	WRITE(6,102)	FN Z1010
102	FORMAT('0B BY ROWS')	FN Z1020
	WRITE(6,120)((B(I,J),J=1,M),I=1,MH)	FN Z1030
120	FORMAT(1H,5E16.8)	FN Z1040
	RETURN	FN Z1050
	END	FN Z1060

NAME OF PROGRAM OR SEGMENT VSUM

SUBROUTINE VSUM(N,L,X,Y,G)

C VECTOR SUM

C FORMS $X = G * Y$ IN X

DIMENSION X(N),Y(N)

DO 10 J=1,L

10 X(J)=X(J)+G*Y(J)

RETURN

END

VSUM 10

VSUM 20

VSUM 30

VSUM 40

VSUM 50

VSUM 60

VSUM 70

VSUM 80

NAME OF PROGRAM OR SEGMENT HP

```

SUBROUTINE HP(N,X,Z,AL,KAP,F,D,Q)
  DIMENSION X(N),Z(N),F(N),D(N),Q(N)
C TRANSFORMATION OF BASIS
C POWER OF X == X**J + EXP(- A0*X**2)
C HARMONIC OSCILLATOR == 2H = P**2 + A0**2 * X**4
C AL=A0
C FORM POWERS AND FACTORIALS NEEDED
  AT=SQRT(AL)
  F(1)=1.0
  Q(1)=1.0
  C=0.7511255445
  D(1)=C*SQRT(AT)
  QD=2.0*AT
  DO 10 I=2,N
    F(I)=F(I-1)*FLOAT(I-1)
    D(I)=0.7071067867*D(I-1)/SQRT(FLOAT(I-1))
  10 Q(I)=QD*Q(I-1)
C KAP = 0 HARMONIC OSCILLATOR TO POWER OF X , KAP = 1 REVERSE
  IF(KAP.EQ.1) GO TO 100
  DO 20 I=1,N
    A=-Q(I)/F(I)
    S=0.0
    NT=(N-I)/2+1
    DO 21 J=1,NI
      L=I+2*J-2
      A=-A
  21 S=S+A*D(L)*F(L)*X(L)/F(J)
  20 Z(I)=S
C Z HAS COEFFICIENTS OF POWER OF X BASIS FUNCTIONS
C Z(K) IS COEFFICIENT OF X**(K-1) M EXP(-AL + X**2)
  RETURN
100 DO 25 I=1,N

```

HP	10
HP	20
HP	30
HP	40
HP	50
HP	60
HP	70
HP	80
HP	90
HP	100
HP	110
HP	120
HP	130
HP	140
HP	150
HP	160
HP	170
HP	180
HP	190
HP	200
HP	210
HP	220
HP	230
HP	240
HP	250
HP	260
HP	270
HP	280
HP	290
HP	300
HP	310
HP	320

NAME OF PROGRAM OR SEGMENT HP

S=0.0	HP	330
NI=(N-1)/2+1	HP	340
DO 26 J=1,NI	HP	350
L=1+2*J-2	HP	360
26 S=S+Z(L)*F(L)/((F(J)*Q(L))	HP	370
25 X(I)=S/((F(I)*D(I))	HP	380
C X(K) IS COEFFICIENT OF (K-1) ST HARMONIC OSCILLATOR EIGENFUNCTION	HP	390
RETURN	HP	400
END	HP	410

NAME OF PROGRAM OR SEGMENT ENER

```

SUBROUTINE ENER(ND,N,AU,A2,A4,A6,AN,EN,C)
C CALCULATES ENERGY ASSOCIATED WITH WAVE FUNCTION DEFINED BY A VECTOR
C OF COEFFICIENTS C AND EXPONENT PARAMETER AU (LENGTH OF VECTOR N,
C MAXIMUM OF N IS ND), WHERE HAMILTONIAN IS
C H = 0.5 * P**2 + A2 * X**2 + A4 * X**4 + A6 * X**6
C ENERGY RETURNED IN EN, NORM OF VECTOR (WAVE FUNCTION) IN AN
DOUBLE PRECISION P,Q,W,Z
DIMENSION C(ND)
Q=0.0D0
P=0.0D0
R=A2=0.5*AU**2
DO 10 J=1,N
Y=0.5*AU*FLOAT(2*J-1)
X=0.5*FLOAT((J-1)*(J-2))
DO 10 I=1,N
K=I+J
W=C(I)*DBLE(C(J))
Q=Q+W*Z(K-2,AU)
P=P+W*(A6*Z(K+4,AU)+A4*Z(K+2,AU)+R*Z(K,AU)+Y*Z(K-2,AU)-X*Z(K-4,AU)
1)
1) CONTINUE
AN=SNGL(Q)
EN=SNGL(P)/AN
WRITE(6,200)N,AU
200 FORMAT('LENGTH OF VECTOR=',I4,' EXPONENT=',F10.5)
WRITE(6,204)AN,EN
204 FORMAT('ONORM=',E16.8,' ENERGY=',E16.8)
RETURN
END
ENER 10
ENER 20
ENER 30
ENER 40
ENER 50
ENER 60
ENER 70
ENER 80
ENER 90
ENER 100
ENER 110
ENER 120
ENER 130
ENER 140
ENER 150
ENER 160
ENER 170
ENER 180
ENER 190
ENER 200
ENER 210
ENER 220
ENER 230
ENER 240
ENER 250
ENER 260
ENER 270
ENER 280
ENER 290

```

NAME OF PROGRAM OR SEGMENT OSST

```

MASTER OSTEEST
C TESTING WAVE FUNCTIONS TO SEE IF THEY OBEY HYPERVIRIAL THEOREMS
REAL KCA,KCS, KCP,KCX,KCB
DIMENSION A(30,30),B(30,30),X(30),Y(30),CI(30),W1(30),W2(30),W3(30)
1),W4(30),W5(30)
COMMON KCA,KCS,KCP,KCX,KCB
ND=30
READ(5,100)KCA,KCS,KCP,KCX,KCB
100 FORMAT(5A1)
READ(5,101)IEN
101 FORMAT(I4)
2 IREP=0
1 READ(5,102)IV,NM,A0,A2,A4,A6,EG
102 FORMAT(2I4,4F10.5,E16.8)
IF(NM.EQ.0)STOP
READ(5,103)(CI(J),J=1,NM)
103 FORMAT(5E16.8)
WRITE(6,201)IV,NM,A0
201 FORMAT('0STATE',I4,' LENGTH OF VECTOR=',I4,' EXPONENT=',F10.5)
202 WRITE(6,202)A2,A4,A6
FORMAT('0HAMILTONIAN=P**2+',F10.5,'*X**2+',F10.5,'*X**4+',F10.5,'*OSST
1X**6')
A6=0.5*A6
A2=0.5*A2
A4=0.5*A4
WRITE(6,203)EG
203 FORMAT('0EIGENVALUE=2*ENERGY=',E16.8)
WRITE(6,204)
204 FORMAT('0VECTOR')
WRITE(6,205)(CI(J),J=1,NM)
212 WRITE(6,212)
FORMAT('0VECTOR IN POLYNOMIAL BASIS')
OSST 10
OSST 20
OSST 30
OSST 40
OSST 50
OSST 60
OSST 70
OSST 80
OSST 90
OSST 100
OSST 110
OSST 120
OSST 130
OSST 140
OSST 150
OSST 160
OSST 170
OSST 180
OSST 190
OSST 200
OSST 210
OSST 220
OSST 230
OSST 240
OSST 250
OSST 260
OSST 270
OSST 280
OSST 290
OSST 300
OSST 310
OSST 320

```

NAME OF PROGRAM OR SEGMENT OSST

```

205  FORMAT(1H ,5E16.8)
C  TRANSFORMATION OF BASIS -  HARMONIC OSCILLATOR TO POWER
      CALL HP(NM,CI,X,A0,0,W1,W2,W3)
      WRITE(6,205)(X(J),J=1,NM)
      IF(IEN=1)3,4,3
C  CALCULATE ENERGY -
4    CALL ENER(ND,NM,A0,A2,A4,A6,AN,EN,X)
      EN=2.0*EN
      WRITE(6,206)AN,EN
206  FORMAT('ONORM=',F12.8,' TWICE ENERGY=',E16.8)
      IF(ABS(AN-1.0).LT.1.0E-07)GO TO 3
      AN=1.0/SQRT(AN)
      DO 20 J=1,NM
20    X(J)=AN*X(J)
3    READ(5,101)MH
C  DIAGONAL HYPERVIRIALS
5    CALL QUER(ND,NM,MH,X,A,B,A0,A2,A4,A6,CI,W1,W2,W3,W4,W5)
6    IF(IREP=1)8,9,8
8    READ(5,101)IOD
      IF(10D)7,1,7
7    IREP=1
      DO 21 J=1,NM
21    Y(J)=X(J)
      NM1=NM
      EG1=EG
      B2=A2
      B4=A4
      B6=A6
      IV1=IV
      AS=A0
      GO TO 1
9    WRITE(6,207)IV,IV1

```

```

OSST 330
OSST 340
OSST 350
OSST 360
OSST 370
OSST 380
OSST 390
OSST 400
OSST 410
OSST 420
OSST 430
OSST 440
OSST 450
OSST 460
OSST 470
OSST 480
OSST 490
OSST 500
OSST 510
OSST 520
OSST 530
OSST 540
OSST 550
OSST 560
OSST 570
OSST 580
OSST 590
OSST 600
OSST 610
OSST 620
OSST 630
OSST 640

```

NAME OF PROGRAM OR SEGMENT OSST

```

207  FORMAT('00FF=DIAGONAL HYPERVIRIAL TEST BETWEEN STATES',I4,' AND ',OSST 650
      1I4)
      IF(A2.NE.B2.OR.A4.NE.B4.OR.A6.NE.B6)GO TO 10
      READ(5,101)MH
      IF(MH)2,2,11
      EV=0.5*(EG-EG1)
11   EV=0.5*(EG-EG1)
C FORM MATRICES P AND Q
      CALL MAKE(ND,NM,MH,Y,A,B,AS,A0,A2,A4,A6,NM1,W1,W2,W3,W4,W5,C1)
      DO 22 I=1,MH
      S1=0.0
      S2=0.0
      DO 23 J=1,NM
      S1=S1+A(I,J)*X(J)
      S2=S2+B(I,J)*X(J)
23   WRITE(6,208)I
      FORMAT('OMATRIX ELEMENTS FOR OPERATOR ',I4)
208  FORMAT('OMATRIX ELEMENTS FOR OPERATOR ',I4)
      WRITE(6,209)S2,S1
209  FORMAT(' EXPECTATION =',E16.8,' EXPECTATION OF COMMUTATOR=',E16.8)OSST 820
      S1=S1-EV*S2
      WRITE(6,210)S1
      FORMAT(' OFF-DIAGONAL RESIDUAL=',E16.8)
210  CONTINUE
      GO TO 2
22   GO TO 2
      WRITE(6,211)
211  FORMAT('OERROR = DIFFERENT HAMILTONIANS')
      GO TO 2
      END
OSST 660
OSST 670
OSST 680
OSST 690
OSST 700
OSST 710
OSST 720
OSST 730
OSST 740
OSST 750
OSST 760
OSST 770
OSST 780
OSST 790
OSST 800
OSST 810
OSST 830
OSST 840
OSST 850
OSST 860
OSST 870
OSST 880
OSST 890
OSST 900
OSST 910

```

NAME OF PROGRAM OR SEGMENT SHOK

```

SUBROUTINE SHOCK(N,N,L,M,TOL,R,E,V)
C LIST EIGENSOLUTIONS L TO M FOR MATRIX R OF ORDER N
C TEST OVERLAPS OF VECTORS AND SOLUTIONS E,X BY (R-E)X=?
C ? SHOULD BE ZERO
C ALSO DIVIDES J TH ELEMENT OF RX BY JTH ELEMENTS OF X TO SEE IF THIS
C RATIO CORRESPONDS TO E
C VECTORS X STORED COLUMNWISE IN COLUMNS L TO M OF N BY N MATRIX V
  DIMENSION R(N,N),E(N,N),V(N,N)
  DO 10 I=L,M
    WRITE(6,101) I,E(I)
101  FORMAT('OEIGENVALUE',I4,' = ',E20.10)
10   WRITE(6,102)(V(J,I),J=1,N)
102  FORMAT(1H ,5E20.10)
    DO 11 I=L,M
      DO 11 J=I,M
        S=0.0
        DO 12 K=1,N
          S=S+V(K,I)*V(K,J)
12   WRITE(6,103)I,J,S
11   FORMAT(' VECTOR PRODUCT',I4,I4,' = ',E20.10)
103  DO 13 I=L,M
      T=E(I)
      WRITE(6,105) I
      DO 14 J=1,N
        S=0.0
        DO 15 K=1,N
          S=S+R(J,K)*V(K,I)
15   W=V(J,I)
          IF(W)16,17,16
17   WRITE(6,107)S
107  FORMAT(' ZERO VECTOR ELEMENT -- NUMERATOR=',E16.8)
      GO TO 18

```

SHOK	10
SHOK	20
SHOK	30
SHOK	40
SHOK	50
SHOK	60
SHOK	70
SHOK	80
SHOK	90
SHOK	100
SHOK	110
SHOK	120
SHOK	130
SHOK	140
SHOK	150
SHOK	160
SHOK	170
SHOK	180
SHOK	190
SHOK	200
SHOK	210
SHOK	220
SHOK	230
SHOK	240
SHOK	250
SHOK	260
SHOK	270
SHOK	280
SHOK	290
SHOK	300
SHOK	310
SHOK	320

NAME OF PROGRAM OR SEGMENT SHOK

16	U=S/W	SHOK 330
	IF(U)20,18,20	SHOK 340
20	IF(ABS((E(I)-U)/U).GT.TOL)WRITE(6,108)E(I),U	SHOK 350
108	FORMAT(' DISCREPANCY -- E=',E16.8,' TEST=',E16.8)	SHOK 360
18	S=S-T*U	SHOK 370
14	WRITE(6,106) S	SHOK 380
105	FORMAT('TEST ON EIGENSOLUTION',I4)	SHOK 390
106	FORMAT(' ELEMENT OF (MATRIX-1*EV)*VECTOR',E16.8)	SHOK 400
13	CONTINUE	SHOK 410
	RETURN	SHOK 420
	END	SHOK 430

NAME OF PROGRAM OR SEGMENT JNH

```

SUBROUTINE JNH(N,NA,M1,M2,R,E,V,A,B,S,T,U,W,X,BG,P1,P2,KZ,NT,RI)
C Awaiting full documentation
C BRIEF DESCRIPTION IN APPENDIX 5
DOUBLE PRECISION O,Q1
DIMENSION R(NA,N),V(NA,N),E(N),A(N),B(N),S(N),T(N),U(N),W(N),X(N)
C NT = 0 SPECIFIES NO VECTORS
C N=ORDER OF R
C BIG IS INFINITY NORM OF TRIDIAGONAL MATRIX
S2 IF(N-1) 50,51,52
IF(M1.GT.N)M1=N
IF(M2.GT.N)M2=N
IF(M1.LT.1)M1=1
IF(M2.LT.1)M2=1
TOLL=P2
WRITE(6,5000)((R(I,J),J=1,N),I=1,N)
5000 FORMAT(' R',5E16.8)
CALL TRY(N,NA,P,A,B)
WRITE(6,5001)(A(I),I=1,N)
5001 FORMAT(' A',5E16.8)
WRITE(6,5002)(B(I),I=1,N)
5002 FORMAT(' B',5E16.8)
CALL VAL(N,M1,M2,A,B,S,E,T,BG,P1,P2,KZ,RL)
IF(NT.FQ.0)RETURN
DEL=RL*BG
I=M1
I1=I+1
18 LDEG=I
IF(I1.GT.M2)GO TO 121
DO 12 J=I1,M2
IF(E(J).EQ.E(J-1))E(J)=E(J)+DEL
IF((E(J)-E(J-1))/(ABS(E(J))+P2).GE.TOLL)GO TO 121
12 LDEG=LDEG+1
JNH 10
JNH 20
JNH 30
JNH 40
JNH 50
JNH 60
JNH 70
JNH 80
JNH 90
JNH 100
JNH 110
JNH 120
JNH 130
JNH 140
JNH 150
JNH 160
JNH 170
JNH 180
JNH 190
JNH 200
JNH 210
JNH 220
JNH 230
JNH 240
JNH 250
JNH 260
JNH 270
JNH 280
JNH 290
JNH 300
JNH 310
JNH 320

```

NAME OF PROGRAM OR SEGMENT JNH

121	CALL VEC(N,HA,R,A,B,E(I),S,T,U,W,X,BG,NT,RL,DEL)	JNH	330
	DO 21 K=1,N	JNH	340
21	V(K,I)=W(K)	JNH	350
	IF(LDEG.EQ.I)GO TO 20	JNH	360
	DO 13 J=1,LDEG	JNH	370
	CALL VEC(N,NA,R,A,B,E(J),S,T,U,W,X,BG,NT,RL,DEL)	JNH	380
	L1=LDEG-1	JNH	390
	DO 14 L=1,L1	JNH	400
	Q=0.0	JNH	410
	DO 15 K=1,N	JNH	420
15	Q=Q+DBLE(W(K))*DBLE(V(K,L))	JNH	430
	WRITE(6,1121)J,L,Q	JNH	440
1121	FORMAT(' J,L,Q',2I4,D20.10)	JNH	450
14	X(L)=Q	JNH	460
	Q1=0.0	JNH	470
	DO 16 K=1,N	JNH	480
	Q=0.0	JNH	490
	DO 17 L=1,L1	JNH	500
17	Q=Q+DBLE(X(L))*DBLE(V(K,L))	JNH	510
	Q=DBLE(W(K))-Q	JNH	520
	Q1=Q+Q+Q1	JNH	530
16	W(K)=Q	JNH	540
	Q1=1.0/DSQRT(Q1)	JNH	550
	DO 13 K=1,N	JNH	560
13	V(K,J)=W(K)*Q1	JNH	570
	L1=LDEG-1+1	JNH	580
	WRITE(6,1115)L1	JNH	590
1115	FORMAT(' DEGENERACY OF ORDER',I4)	JNH	600
20	I=LDEG+1	JNH	610
	IF(I-M2)18,18,50	JNH	620
50	RETURN	JNH	630
51	E(1)=R(1,1)	JNH	640

NAME OF PROGRAM OR SEGMENT JNH

V(1,1)=1.0
A(1)=E(1)
B(1)=0.0
RETURN
END

JNH 650
JNH 660
JNH 670
JNH 680
JNH 690

NAME OF PROGRAM OR SEGMENT TRY

	SUBROUTINE TRY(N,NA,R,A,B)	TRY	10
	DOUBLE PRECISION S,RS	TRY	20
	DIMENSION R(NA,N),A(N),B(N)	TRY	30
C	HOUSEHOLDER TRIDIAGONALIZATION	TRY	40
	N2=N-2	TRY	50
	IF(N2.EQ.0)GO TO 8	TRY	60
	DO 100 I=1,N2	TRY	70
	A(I)=R(I,I)	TRY	80
	S=0.0	TRY	90
	I1=I+1	TRY	100
	DO 1 J=I1,N	TRY	110
1	S=S+DBLE(R(I,J))*2	TRY	120
	IF(S.NE.0.0)GO TO 9	TRY	130
	B(I)=R(I,I1)	TRY	140
	GO TO 100	TRY	150
9	RS=R(I,I1)	TRY	160
	S=DSQRT(S)	TRY	170
	IF(RS.LT.0.0)S=-S	TRY	180
	B(I)=-S	TRY	190
	RS=RS+S	TRY	200
	R(I,I1)=RS	TRY	210
	C=1.0/(S*RS)	TRY	220
	S=0.0	TRY	230
	DO 3 J=I1,N	TRY	240
	RS=0.0	TRY	250
4	DO 4 K=I1,J	TRY	260
	RS=RS+DBLE(R(K,J))*DBLE(R(I,K))	TRY	270
	K=J	TRY	280
42	IF(K.EQ.N)GO TO 41	TRY	290
	K=K+1	TRY	300
	RS=RS+DBLE(R(J,K))*DBLE(R(I,K))	TRY	310
	GO TO 42	TRY	320

NAME OF PROGRAM OR SEGMENT TRY

41	B(J)=PS*C	TRY	350
3	S=S+DBLE(R(I,J))*DBLE(B(J))	TRY	340
	S=-0.5*C*S	TRY	350
	DO 6 J=I1,N	TRY	360
6	B(J)=B(J)+S*R(I,J)	TRY	370
	DO 7 J=I1,N	TRY	380
	DO 7 K=J,N	TRY	390
7	R(J,K)=R(J,K)-R(I,J)*B(K)-B(J)*R(I,K)	TRY	400
100	CONTINUE	TRY	410
8	B(N)=1.0	TRY	420
	L=N-1	TRY	430
	B(L)=R(L,N)	TRY	440
	A(L)=R(L,L)	TRY	450
	A(N)=R(N,N)	TRY	460
	RETURN	TRY	470
	END	TRY	480

NAME OF PROGRAM OR SEGMENT VAL

	SUBROUTINE VAL(N,M1,M2,C,B,BETA,E,W,XMAX,EPS1,EPS2,K2,RELF)	VAL	10
C	TRIDIRISECTION	VAL	20
	DIMENSION C(N),B(N),BETA(N),E(N),W(N)	VAL	30
	N1=N-1	VAL	40
C	CALCULATE NORMS	VAL	50
	S=ABS(B(1))	VAL	60
	XMIN=C(1)-S	VAL	70
	XMAX=C(1)+S	VAL	80
	BETA(1)=0.0	VAL	90
	BETA(2)=S*S	VAL	100
	IF(M1.EQ.1)GO TO 11	VAL	110
	DO 1 I=2,N1	VAL	120
	BETA(I+1)=B(I)**2	VAL	130
	T=ABS(B(I))	VAL	140
	P=T+S	VAL	150
	Q=C(I)	VAL	160
	IF(Q+P.GT.XMAX)XMAX=Q+P	VAL	170
	IF(Q-P.LT.XMIN)XMIN=Q-P	VAL	180
1	S=T	VAL	190
11	Q=C(N)	VAL	200
	IF(Q-S.LT.XMIN)XMIN=Q-S	VAL	210
	IF(Q+S.GT.XMAX)XMAX=Q+S	VAL	220
	S=XMAX	VAL	230
	IF(S+XMIN.LT.0.0)S=-XMIN	VAL	240
	EPS2=RELF*S	VAL	250
	IF(EPS1.LE.0.0)EPS1=EPS2	VAL	260
	EPS2=0.5*EPS1+7.0*EPS2	VAL	270
	P=2.0*RELF	VAL	280
C	INNER BLOCK	VAL	290
	K2=0	VAL	300
	DO 101 K=M1,M2	VAL	310
	E(K)=XMIN	VAL	320

NAME OF PROGRAM OR SEGMENT VAL

101	W(K)=XMAX	VAL 330
C	LOOP FOR KTH EIGENVALUE	VAL 340
	XU=XMIN	VAL 350
	DO 100 K=M1,M2	VAL 360
	X0=XMAX	VAL 370
	DO 2 I=K,M2	VAL 380
	IF(X0.LE.W(I))GO TO 2	VAL 390
	X0=W(I)	VAL 400
	GO TO 3	VAL 410
2	CONTINUE	VAL 420
3	IF(XU.LT.E(K))XU=E(K)	VAL 430
4	X1=(XU+X0)*0.5	VAL 440
	IF(X0-XU.LE.P*(ABS(XU)+ABS(X0))+EPS1)GO TO 100	VAL 450
	K2=K2+1	VAL 460
	L=0	VAL 470
	Q=1.0	VAL 480
	DO 5 I=1,N	VAL 490
	Q=C(I)-X1-BETA(I)/Q	VAL 500
	IF(Q)6,7,5	VAL 510
7	Q=ABS(R(I))*RELF	VAL 520
	GO TO 5	VAL 530
6	L=L+1	VAL 540
5	CONTINUE	VAL 550
	IF(L.LT.K)GO TO 8	VAL 560
	X0=X1	VAL 570
	IF(L.GE.M2)GO TO 9	VAL 580
	L1=L+1	VAL 590
	W(L)=X1	VAL 600
	IF(E(L1).LT.X1)E(L1)=X1	VAL 610
	GO TO 4	VAL 620
9	W(M2)=X1	VAL 630
	GO TO 4	VAL 640

NAME OF PROGRAM OR SEGMENT VAL

8	XU=X1	VAL 650
	GO TO 4	VAL 660
100	E(K)=(XU+X0)*0.5	VAL 670
	XMAX=S	VAL 680
	RETURN	VAL 690
	END	VAL 700

NAME OF PROGRAM OR SEGMENT VEC

SUBROUTINE VEC(N,NA,R,A,B,E,S,T,U,W,X,RIG,NLIM,RELF,EPS)
 DIMENSION R(NA,N),A(N),B(N),S(N),T(N),U(N),W(N),X(N)
 C VECTORS BY INVERSE ITERATION

N1=N-1	VEC	10
ETA=1.0/FLOAT(N)	VEC	20
P=A(1)-E	VEC	30
Q=B(1)	VEC	40
IF(Q.EQ.0.0)Q=EPS	VEC	50
DO 2 I=1,N1	VEC	60
L=I+1	VEC	70
BI=B(I)	VEC	80
IF(BI.EQ.0.0)BI=EPS	VEC	90
BL=B(L)	VEC	100
IF(BL.EQ.0.0)BL=EPS	VEC	110
IF(ABS(P).GE.ABS(BI))GO TO 3	VEC	120
X(I)=P/BI	VEC	130
IF(X(I).EQ.0.0.AND.BI.LE.EPS)X(I)=1.0	VEC	140
S(I)=BI	VEC	150
T(I)=A(L)-E	VEC	160
U(I)=BL	VEC	170
P=Q-X(I)*T(I)	VEC	180
Q=-X(I)*BL	VEC	190
GO TO 4	VEC	200
X(I)=BI/P	VEC	210
U(I)=0.0	VEC	220
S(I)=P	VEC	230
T(I)=Q	VEC	240
P=A(L)-E-X(I)*Q	VEC	250
Q=BL	VEC	260
IF(S(I).EQ.0.0)S(I)=EPS	VEC	270
CONTINUE	VEC	280
S(N)=P	VEC	290
	VEC	300
	VEC	310
	VEC	320

NAME OF PROGRAM OR SEGMENT VEC

```

      T(N)=0.0
      U(N)=0.0
      IF(S(N).EQ.0.0)S(N)=EPS
      W(N)=ETA/S(N)
      W(N1)=(ETA-T(N1)*W(N))/S(N1)
      Q=W(N)**2+W(N1)**2
      IF(N.EQ.2)GO TO 12
      DO 5 I=2,N1
      L=N-I
      W(L)=(ETA-T(L)*W(L+1)-U(L)*W(L+2))/S(L)
      Q=Q+W(L)**2
      IF(NLIM.EQ.1)GO TO 12
      NIT=1
      NIT=NIT+1
      Q=1.0/SQRT(Q)
      W(1)=W(1)*Q
      DO 7 I=2,N
      L=I-1
      IF(U(L).NE.0.0)GO TO 8
      W(I)=W(I)*Q-W(L)*X(L)
      GO TO 7
      BI=W(I)*Q
      W(I)=W(L)-BI*X(L)
      W(L)=BI
      CONTINUE
      W(N)=W(N)/S(N)
      W(N1)=(W(N1)-T(N1)*W(N))/S(N1)
      Q=W(N)**2+W(N1)**2
      DO 9 I=2,N1
      L=N-I
      W(L)=(W(L)-T(L)*W(L+1)-U(L)*W(L+2))/S(L)
      Q=Q+W(L)**2

```

VEC	330
VEC	340
VEC	350
VEC	360
VEC	370
VEC	380
VEC	390
VEC	400
VEC	410
VEC	420
VEC	430
VEC	440
VEC	450
VEC	460
VEC	470
VEC	480
VEC	490
VEC	500
VEC	510
VEC	520
VEC	530
VEC	540
VEC	550
VEC	560
VEC	570
VEC	580
VEC	590
VEC	600
VEC	610
VEC	620
VEC	630
VEC	640

NAME OF PROGRAM OR SEGMENT VEC

12	IF(NIT.LT.NLIM) GO TO 6	VEC 650
	Q=1.0/SQRT(Q)	VEC 660
	DO 10 I=1,N	VEC 670
10	W(I)=W(I)*Q	VEC 680
	IF(N1.LT.2)RETURN	VEC 690
	DO 11 I=2,N1	VEC 700
	L=N-I	VEC 710
	L1=1+L	VEC 720
	BT=0.0	VEC 730
	DO 18 J=L1,N	VEC 740
18	BT=BT+P(L,J)*W(J)	VEC 750
	IF(R(L,L+1).EQ.0.0.OR.B(L).EQ.0.0)GO TO 20	VEC 760
	BT=-BT/(R(L,L+1)*B(L))	VEC 770
20	DO 19 J=L1,N	VEC 780
19	W(J)=W(J)-BT*P(L,J)	VEC 790
11	CONTINUE	VEC 800
	RETURN	VEC 810
	END	VEC 820

NAME OF PROGRAM OR SEGMENT OSC6

```

MASTER OSC6
DOUBLE PRECISION R1,R2,R3,R4,A1(100),H(100)
INTEGER P,Q,PQ
DIMENSION AS(10,10),BS(10,10)
DIMENSION A(10,10),B(10,10),AP(10,10),BP(10,10)
DIMENSION X(21),CI(21),V(21),T(21),Z(21)
ND=10
ND1=21
NS=100
1 READ(5,102)IV,NM,A0,A2,A4,A6,EG
C NM IS LENGTH INITIAL STATE
C IV IS STATE NUMBER
102 FORMAT(2I4,4F10.5,E16.8)
IF(NM.EQ.0)STOP
READ(5,103)(CI(J),J=1,NM)
103 FORMAT(5E16.8)
WRITE(6,201)IV,NM,A0
201 FORMAT('0STATE',I4,' LENGTH OF VECTOR=',I4,' EXPONENT=',F10.5)
202 FORMAT('0HAMILTONIAN=P**2+',F10.5,'*X**2+',F10.5,'*X**4+',F10.5,'*X**6')
1X**6')
A6=0.5*A6
A2=0.5*A2
A4=0.5*A4
WRITE(6,203)EG
203 FORMAT('0EIGENVALUE=2*ENERGY=',E16.8)
WRITE(6,204)
204 FORMAT('0VECTOR')
WRITE(6,205)(CI(J),J=1,NM)
205 FORMAT('0VECTOR IN POLYNOMIAL BASIS')
WRITE(6,212)
212 FORMAT('0VECTOR IN POLYNOMIAL BASIS')
205 FORMAT(1H ,5E16.8)

```

OSC6 10

OSC6 20

OSC6 30

OSC6 40

OSC6 50

OSC6 60

OSC6 70

OSC6 80

OSC6 90

OSC6 100

OSC6 110

OSC6 120

OSC6 130

OSC6 140

OSC6 150

OSC6 160

OSC6 170

OSC6 180

OSC6 190

OSC6 200

OSC6 210

OSC6 220

OSC6 230

OSC6 240

OSC6 250

OSC6 260

OSC6 270

OSC6 280

OSC6 290

OSC6 300

OSC6 310

OSC6 320

NAME OF PROGRAM OR SEGMENT OSC6

```

C TRANSFORMATION OF BASIS - HARMONIC OSCILLATOR TO POWER
      CALL HP(NM,CI,X,A0,0,T,Z,V)
      WRITE(6,205)(X(J),J=1,NM)
C CALCULATE ENERGY - INITIAL STATE
4     CALL ENER(ND,NM,A0,A2,A4,A6,AN,EN,X)
      EN=2.0*EN
      WRITE(6,206)AN,EN
206    FORMAT('ONORM=1,F12.8,' TWICE ENERGY=1,E16.8)
C NORMALIZE IF NECESSARY
      IF(ABS(AN-1.0).LT.1.0E-07)GO TO 3
      AN=1.0/SORT(AN)
      DO 20 J=1,NM
20     X(J)=AN*X(J)
3     READ(5,151)M,MH,IU,SUSP
C SUSP= GUESS FOR EIGENVALUE
C IU= EXPECTED NUMBER OF FINAL STATE I.E. STATE 4
C M= NUMBER OF BASIS FUNCTIONS IN FINAL STATE
C MH= NUMBER OF HYPERVIRIALS
151    FORMAT(3I4,F10.5)
      READ(5,152)(T(K),K=1,MH)
152    FORMAT(10F8.5)
      WRITE(6,159)MH
159    FORMAT(1H0,I4,' HYPERVIRIALS FOR T VALUES:')
      WRITE(6,153)(T(K),K=1,MH)
153    FORMAT(1H,10F10.5)
      NAP=(M+1)/2
      NIP=(NM+1)/2
      Q=M+1-2*NAP
      P=IV-2*(IV/2)
      PQ=P+Q
      NUF=NIP+NAP+1
      NOG=2*NUF+PQ+4

```

OSC6 330

OSC6 340

OSC6 350

OSC6 360

OSC6 370

OSC6 380

OSC6 390

OSC6 400

OSC6 410

OSC6 420

OSC6 430

OSC6 440

OSC6 450

OSC6 460

OSC6 470

OSC6 480

OSC6 490

OSC6 500

OSC6 510

OSC6 520

OSC6 530

OSC6 540

OSC6 550

OSC6 560

OSC6 570

OSC6 580

OSC6 590

OSC6 600

OSC6 610

OSC6 620

OSC6 630

OSC6 640

NAME OF PROGRAM OR SEGMENT OSC6

```

171 WRITE(6,171)P,Q
    FORMAT(' PARITIES LOWER=',I4,' UPPER=',I4)
    IF(MH-NAP)5,6,6
5    WRITE(6,150)MH,NAP
150  FORMAT('NUMBER OF CONDITIONS',I4,'.LT.',I4,' , NUMBER OF BASIS FNS',
    1.'')
    STOP
C CALCULATE PARAMETERS NEEDED
6    AL(1)=1.0D0
    S=-0.25/A0
    A02=SQRT(A0)
    DO 11 I=2,NUP
11    AL(I)=S*AL(I-1)
    WRITE(6,1000)(AL(LJ),LJ=1,NUP)
1000  FORMAT(' AL',5D16.8)
    DO 10 K=1,MH
    U=-0.5*I(K)/A02
C HERMITE FUNCTION VALUES
    CALL HERM(NOG,U,H)
    V=I(K)
    DO 12 J=1,NAP
    R1=0.0D0
    R2=0.0D0
    R3=0.0D0
C FORM MATRICES P AND Q
    DO 13 I=1,NIP
    LAP=2*I+P-2
    FLAP=FLOAT(LAP)
    KR=I+J-1
    L=2*KR+PQ+1
    RA=X(LAP+1)*AL(KR)
    R1=R1+R4*0.5*V*(V*H(L)+A02*(4.0*FLOAT(LAP)+H(L-1)+H(L+1)))

```

OSC6 650

OSC6 660

OSC6 670

OSC6 680

OSC6 690

OSC6 700

OSC6 710

OSC6 720

OSC6 730

OSC6 740

OSC6 750

OSC6 760

OSC6 770

OSC6 780

OSC6 790

OSC6 800

OSC6 810

OSC6 820

OSC6 830

OSC6 840

OSC6 850

OSC6 860

OSC6 870

OSC6 880

OSC6 890

OSC6 900

OSC6 910

OSC6 920

OSC6 930

OSC6 940

OSC6 950

OSC6 960

NAME OF PROGRAM OR SEGMENT OSC6

```

      R2=R2+R4*(0.25*V+V*(16.*A0*FLAP*(FLAP-1.)*H(L-2)+(4.*A0*(2.*FLAP+10SC6 970
1.)+V*V)*H(L)+A0*H(L+2)+2.*V*A02*(4.*FLAP*H(L-1)+H(L+1)))-U*(2.*A2*OSC6 980
2H(L+1)+(4.*A4*H(L+3)+6.*A6*H(L+5)*S)*S))
      R3=R3+R4*H(L)
13  CONTINUE
      A(K,J)=R1
      AP(K,J)=R2
      B(K,J)=R3
      BP(K,J)=R3
      BS(K,J)=R1
      AS(K,J)=R2
12  CONTINUE
10  CONTINUE
      IREP=0
C CASE 1
35  EV=SUSP*(IREP+1)
36  WRITE(6,163)MH,NAP
163  FORMAT('0A BY ROWS',I4,' ROWS',I4,' COLUMNS')
      WRITE(6,205)((A(LI,LJ),LJ=1,NAP),LI=1,MH)
      WRITE(6,162)
162  FORMAT('0B BY ROWS')
      WRITE(6,205)((B(LI,LJ),LJ=1,NAP),LI=1,MH)
      WRITE(6,161)EV
161  FORMAT('0SUSPECTED ROOT AT',F10.5)
C SOLUTION OF MATRIX EIGENVALUE PROBLEM
      IF(EV.NE.0.0)CALL ADJ(ND,NAP,A,B,EV)
      CALL REDN(ND,NAP,A,B,IFAIL)
      IF(IFAIL.EQ.1)GO TO 22
C INITIALIZE VECTOR
      ANM=1.0/SQRT(FLOAT(NAP))
      DO 14 J=1,NAP
14  X(J)=ANM

```

```

OSC61000
OSC61010
OSC61020
OSC61030
OSC61040
OSC61050
OSC61060
OSC61070
OSC61080
OSC61090
OSC61100
OSC61110
OSC61120
OSC61130
OSC61140
OSC61150
OSC61160
OSC61170
OSC61180
OSC61190
OSC61200
OSC61210
OSC61220
OSC61230
OSC61240
OSC61250
OSC61260
OSC61270
OSC61280

```

NAME OF PROGRAM OR SEGMENT OSC6

```

      LIM=100
      CALL ITN(ND,NAP,1.0E-10,LIM,A,B,X,EV,Y)
      WRITE(6,154)EV,LIM
154  FORMAT('OFINAL VALUE ',E16.8,' AT',I4,' ITERATIONS')
      IF(IREP.EQ.1)GO TO 27
28  DO 180 I=1,M
      R1=0.0D0
      C CALCULATE "RESIDUALS"
      DO 181 J=1,NAP
181  R1=R1+DBLE(X(J))*(DBLE(BS(I,J))-DBLE(EV)*DBLE(RP(I,J)))
180  WRITE(6,182)R1
182  FORMAT(' RESIDUAL ',D20.10)
      C EXPAND VECTOR
      DO 19 I=1,NAP
      L=NAP-I+1
      K=2*L+Q-1
      X(K)=X(L)
19  X(K+1)=0.0
      IF(Q.EQ.1)X(1)=0.0
      WRITE(6,172)(X(J),J=1,M)
172  FORMAT(' EXPANDED X',5E16.8)
      C CALCULATE ENERGY -- FINAL STATE
      CALL ENER(ND1,M ,A0,A2,A4,A6,AN,EM,X)
      EM=2.0*EM
      WRITE(6,206)AN,EM
      IF(ABS(AN-1.0).LT.1.0E-07)GO TO 16
      AN=1.0/SQRT(AN)
      DO 17 J=1,M
17  X(J)=AN*X(J)
      C TRANSFORMATION OF BASIS - POWER TO HARMONIC OSCILLATOR
16  CALL HP(M,C1,X,A0,1,Y,Z,T)
      WRITE(6,164)IU,A0

```

OSC61290

OSC61300

OSC61310

OSC61320

OSC61330

OSC61340

OSC61350

OSC61360

OSC61370

OSC61380

OSC61390

OSC61400

OSC61410

OSC61420

OSC61430

OSC61440

OSC61450

OSC61460

OSC61470

OSC61480

OSC61490

OSC61500

OSC61510

OSC61520

OSC61530

OSC61540

OSC61550

OSC61560

OSC61570

OSC61580

OSC61590

OSC61600

NAME OF PROGRAM OR SEGMENT OSC6

```

164  FORMAT('VECTOR FOR STATE',I4,' IN H.O. FNS WITH EXPONENT=',F10.5)OSC61610
      WRITE(6,205)(CI(J),J=1,M)
      IF(MH=NAP)5,22,23
23    M1=NAP+1
      DO 24 K=M1,MH
      R1=0.000
      DO 25 J=1,NAP
      L=2*J+1
25    R1=R1+X(L)*(A(K,J)-EV*B(K,J))
24    WRITE(6,155)K,R1
155  FORMAT('OFF DIAGONAL HYPERVIRIAL TEST FOR ROW',I4,' = ',D16.8)
22    IF(IREP-1)38,29,1
38    IREP=1
C CASE II
      DO 26 K=1,MH
      DO 26 J=1,NAP
      A(K,J)=AP(K,J)
26    B(K,J)=BP(K,J)
      GO TO 35
27    EV=SQRT(EV)
      IF(SUSP.LT.0.0)EV=-EV
      EV2=2.0*EV
165  WRITE(6,165)EV,EV2
      FORMAT('ENERGY DIFFERENCE AND DOUBLE',2E16.8)
      GO TO 28
29    IREP=IREP+1
      EV=SUSP
C CASE III
      DO 37 I=1,MH
      DO 37 J=1,NAP
      A(I,J)=AS(I,J)
37    B(I,J)=BS(I,J)

```

OSC61610

OSC61620

OSC61630

OSC61640

OSC61650

OSC61660

OSC61670

OSC61680

OSC61690

OSC61700

OSC61710

OSC61720

OSC61730

OSC61740

OSC61750

OSC61760

OSC61770

OSC61780

OSC61790

OSC61800

OSC61810

OSC61820

OSC61830

OSC61840

OSC61850

OSC61860

OSC61870

OSC61880

OSC61890

OSC61900

OSC61910

OSC61920

NAME OF PROGRAM OR SEGMENT OSC6

GO TO 36
END

OSC61930
OSC61940

NAME OF PROGRAM OR SEGMENT HERM

```

SUBROUTINE HERM(NOG,U,H)
C H(L)= HERMITE POLYNOMIAL OF ORDER (L-3) IN U
  DOUBLE PRECISION H(NOG)
  H(1)=0.0D0
  H(2)=0.0D0
  H(3)=1.0D0
  DO 10 I=4,NOG
    H(I)=2.0*(U*H(I-1))-FLOAT(I-4)*H(I-2))
  10 RETURN
END

```

HERM 10
 HERM 20
 HERM 30
 HERM 40
 HERM 50
 HERM 60
 HERM 70
 HERM 80
 HERM 90
 HERM 100

```

1 MASTER GOODLIST
2 INTEGER A(18),TERM,NAME
3 READ(5,100) TERM
4   FORMAT (A4)
5   TERM=TERM/4
6   1 READ (5,100) NAME
7   IF(NAME/4.EQ.TERM)GO TO 1000
8   READ (5,101) LINE
9   101 FORMAT (14)
10  IF(LINE.GT.100) GO TO 10
11  LINS=10+LINE
12  NUM=0
13  2 WRITE(6,201)NAME
14  201 FORMAT(1H1,/////////,10X,' NAME OF PROGRAM OR SEGMENT',2X,A4)
15  WRITE (6,202)
16  202 FORMAT (1H0)
17  3 READ (5,102) A
18  102 FORMAT(18A4)
19  IF(A(1)/4.EQ.TERM)GO TO 1
20  NUM=NUM+10
21  WRITE (6,203) A,NAME,NUM
22  203 FORMAT(1H ,10X,18A4,A4,14)
23  IF(MOD(NUM,LINS))3,2,3
24  10 NUM=0
25  11 READ(5,102) A
26  IF(A(1)/4.EQ.TERM)GO TO 1
27  NUM=NUM+10
28  WRITE(6,200) A,NAME,NUM
29  200 FORMAT(18A4,A4,14)
30  GO TO 11
31  1000 WRITE(6,205)
32  205 FORMAT(1H1)
33  STOP
34  END

```

FINISH

EXECUTION STARTED