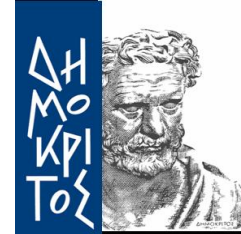




ΣΧΟΛΗ
ΕΦΑΡΜΟΣΜΕΝΩΝ
ΜΑΘΗΜΑΤΙΚΩΝ &
ΦΥΣΙΚΩΝ
ΕΠΙΣΤΗΜΩΝ
ΣΧΟΛΗ
ΜΗΧΑΝΟΛΟΓΩΝ
ΜΗΧΑΝΙΚΩΝ

ΙΝΣΤΙΤΟΥΤΟ
ΠΥΡΗΝΙΚΗΣ
& ΣΩΜΑΤΙΔΙΑΚΗΣ
ΦΥΣΙΚΗΣ
ΙΝΣΤΙΤΟΥΤΟ
ΝΑΝΟΕΠΙΣΤΗΜΗΣ
&
ΝΑΝΟΤΕΧΝΟΛΟΓΙΑΣ



**Αρχή Βελτιστοποίησης
Κυματοσυναρτήσεων σε Διηγερμένες
Καταστάσεις
(με εφαρμογή σε διατομικά μόρια)**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
Μ.Δ.Ε. «Φυσική και Τεχνολογικές Εφαρμογές»

ΤΟΥ

Δελλαπόρτα Π. Έκτορα

Ερευνητικός Επιβλέπων:	Ακαδημαϊκός Επιβλέπων:
Ναούμ Μπακάλης, Ινστιτούτο Θεωρητικής και Φυσικής Χημείας, Ε.Ι.Ε.	Κωστής Παρασκευαΐδης, Τομέας Φυσικής, ΣΕΜΦΕ, Ε.Μ.Π.

ΑΘΗΝΑ, ΜΑΙΟΣ 2014

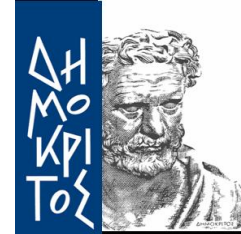


ΣΧΟΛΗ
ΕΦΑΡΜΟΣΜΕΝΩΝ
ΜΑΘΗΜΑΤΙΚΩΝ &
ΦΥΣΙΚΩΝ
ΕΠΙΣΤΗΜΩΝ

ΣΧΟΛΗ
ΜΗΧΑΝΟΛΟΓΩΝ
ΜΗΧΑΝΙΚΩΝ

ΙΝΣΤΙΤΟΥΤΟ
ΠΥΡΗΝΙΚΗΣ
& ΣΩΜΑΤΙΔΙΑΚΗΣ
ΦΥΣΙΚΗΣ

ΙΝΣΤΙΤΟΥΤΟ
ΝΑΝΟΕΠΙΣΤΗΜΗΣ
&
ΝΑΝΟΤΕΧΝΟΛΟΓΙΑΣ



Αρχή Βελτιστοποίησης Κυματοσυναρτήσεων σε Διηγερμένες Καταστάσεις (με εφαρμογή σε διατομικά μόρια)

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
Μ.Δ.Ε. «Φυσική και Τεχνολογικές Εφαρμογές»

του

Δελλαπόρτα Π. Έκτορα

Επιβλέπων : Ναούμ Μπακάλης
Κύριος Ερευνητής,
Ινστιτούτο Θεωρητικής και Φυσικής Χημείας,
Εθνικό Ίδρυμα Ερευνών.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 6 / 6 / 2014

.....
Ναούμ Μπακάλης
Ερευνητής Α'
Ι.Θ.Φ.Χ-Ε.Ι.Ε

.....
Κώστας Παρασκευαΐδης
Καθηγητής Ε.Μ.Π

.....
Κουτσούμπας Γεώργιος
Καθηγητής Ε.Μ.Π

.....
Έκτορας Π. Δελλαπόρτας

Διπλωματούχος Χημικός, Τμήματος Χημείας, Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών.

Διπλωματούχος Μ.Δ.Ε. «Φυσική & Τεχνολογικές Εφαρμογές», Σ.Ε.Μ.Φ.Ε.,
Ε.Μ.Π

© 2014, Εθνικό Μετσόβιο Πολυτεχνείο. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Ευχαριστίες

Η ολοκλήρωση της παρούσας διπλωματικής εργασίας σηματοδοτεί τη λήξη μιας προσπάθειας χρόνων. Νιώθω βαθιά δικαιωμένος για την επιλογή μου να ασχοληθώ με αυτό το καθαρά ερευνητικό έργο, καθώς τα κέρδη είναι πολλά σε μορφωτικό και σε προσωπικό επίπεδο. Το θέμα αυτό μου είχε κινήσει τη περιέργια ήδη σε προπτυχιακό επίπεδο, την οποία και ικανοποίησα σε μεγάλο βαθμό με την ενασχόλησή μου με αυτό. Πιστεύω ότι βελτιώθηκα πολύ ως επιστήμονας ενώ παράλληλα μου δόθηκαν πολλά κίνητρα για το μέλλον. Στη διαδρομή αυτή φυσικά δεν ήμουν μόνος· πολλοί είναι εκείνοι που με στήριξαν και με πίστεψαν, γι' αυτό και οι γραμμές αυτής της σελίδας μου φαίνονται απελπιστικά λίγες. Θέλω, λοιπόν, να ευχαριστήσω αρχικά τον επιβλέποντα καθηγητή μου, Ναούμ Μπακάλη, που με την απέραντη υπομονή του και τις καίριες παρεμβάσεις του με βοήθησε σε όλη αυτή την προσπάθεια παρά το βαρύ του πρόγραμμα. Δευτερευόντως, θα ήθελα να ευχαριστήσω τη μητέρα μου, Μαρία Π., χωρίς την υποστήριξη της οποίας, συναισθηματική και επαγγελματική, η εργασία αυτή δεν θα μπορούσε να περατωθεί. Τρίτον, ευχαριστώ το ίδιο το φαρμακείο, όπου η εργασία μου σε αυτό ως Χημικός εργαστηρίου, μου εξασφάλισε τους απαραίτητους οικονομικούς πόρους. Δε θα μπορούσα να παραλείψω φυσικά τους φίλους-συναδέλφους Γεράσιμο Β. (A.K.A gerry) για τη βοήθειά του σε σχέση με τη $\text{\LaTeX}2_{\epsilon}$ και τα διαδικαστικά του μεταπτυχιακού, Ολίνα Μ. για τη πολύτιμη στήριξη που μου παρείχε ως συμφοιτήριά μου στη Φαρμακευτική και στο Χημικό και φυσικά στην κα.

Αστέρω Π. οι συζητήσεις της οποίας με κατεύθηναν στο μεταπτυχιακό. Πολλά, επίσης, χρωστάω στους γονείς μου που με αμέριστη αγάπη είναι δίπλα μου σε κάθε στροφή της ζωής μου και στην Ανδριάνα Π. (Α.Κ.Α. μικρούλα) που με ανέχτηκε όλο αυτο το διάστημα, γεμίζοντάς με παράλληλα με αισιοδοξία. Κλείνοντας, θέλω να αφιερώσω την εργασία αυτή στην γάτα μου (Α.Κ.Α. *Bianca*), που κοιμόταν και ροχάλιζε δίπλα μου καθόλη τη διάρκεια της εργασίας.

Έκτορας

Περίληψη

Το περιεχόμενο της παρούσας διπλωματικής εργασίας μπορεί να χωριστεί σε τρεις θεματικές ενότητες :

1. τη μελέτη των μονοηλεκτρονιακών διατομικών τροχιακών (*OEDO*) καθώς και των διαφορικών εξισώσεων απο τις οποίες προκύπτουν,
2. την ανάπτυξη των κατάλληλων μητροστοιχείων και μεθοδολογίας, για την επίλυση του ηλεκτρονιακού προβλήματος σε πολυηλεκτρονιακά διατομικά μόρια και
3. την περιγραφή της αρχής μεταβολών για διεγερμένες καταστάσεις N.C. Bacalis, Z. Xiong, D. Karaoulanis [1] και την ανάπτυξη προγράμματος FORTRAN για την κωδικοποίηση όλων των παραπάνω.

Αναλυτικότερα

Για τη πληρέστερη κατανόηση του πρώτου σκέλους, γίνεται αναφορά στις διάφορες μεθόδους επίλυσης των γενικευμένων σφαιροειδών διαφορικών εξισώσεων (GSDE) (παρ. Β' και κεφ. 3, στίς σελ. 96 και 17, αντίστοιχα), που προκύπτουν απο την εξίσωση *Schrödinger* για τα μονοηλεκτρονιακά διατομικά μόρια (*OEDM*) (παρ. Α', σελ. 83), καθώς και των λύσεων που παράγονται.

Στη δεύτερη θεματική ενότητα αναλύουμε την οριζουσα *Slater* και τις μεθόδους συντελεστών *Lagrange* και *CI* (κεφ. 4, σελ. 33), ενώ

αναπτύσσουμε διεξοδικά το πολυπολικό ανάπτυγμα κατά *Neumann* και τα διάφορα μητροστοιχεία που είναι απαραίτητα στον υπολογισμό των ιδιοτήτων και καταστάσεων των OEDM's (κεφ. 5, σελ. 52).

Στη τρίτη θεματική ενότητα εισάγουμε το συναρτησοειδές \mathcal{F} [1] και περιγράφουμε τα σημαντικότερα σημεία της διαδικασίας υπολογισμού του (κεφ. 1, σελ. 3). Τέλος, παραθέτουμε τις κεντρικές subroutines του προγράμματος FORTRAN (παρ. Γ', σελ. 108) καθώς και αυτές που υπολογίζουν τα επιμέρους μητροστοιχεία (παρ. Δ', σελ. 203), τα οποία βασίζονται στην δεύτερη θεματική ενότητα.

Abstract

This master thesis is divided in three parts :

1. the study of one electron diatomic orbitals (*OEDO*), as well as the differential equations that produce them,
2. the development of the correct methodology and matrix elements, for solving the electronic problem of many electron diatomic molecules and
3. the description of the N.C. Bacalis, Z. Xiong, D. Karaoulanis [1] variation theory and finally, the development of FORTRAN code for the implementation of all of the above.

Namely

For better understanding of the first part, I make a reference on the various methods of solving the Generalised spheroidal differential equations (GSDE) (app. B' and ch. 3, at p. 96 and 17, respectively), that result from the *Schrödinger* equation for the one electron diatomic molecules (*OEDM*) (app. A', p. 83), and to the generated solutions.

In the second part I develop the *Slater* determinant, the *CI* and the *Lagrange* multipliers methods (ch. 4, p. 33) moreover, I introduce the *Neumann* multipole expansion and the necessary matrix elements, for the calculation of *OEDM*'s properties and corresponding states (ch. 5, p. 52).

In the third part I define the functional \mathcal{F} [1] and describe the highlights of the procedure of calculating it (ch. 1, p. 3). Finally, I display the main subroutines of the FORTRAN code (ap. Γ , p. 108), as well as those that calculate the matrix elements (ap. Δ , p. 203), which I have derived in the second part.

Πληροφορίες

Η εργασία αυτή πραγματοποιήθηκε ως επι τω πλείστον στο Εθνικό Ίδρυμα Ερευνών, υπό την επίβλεψη του κύριου ερευνητή, Ναούμ Μπακάλη. Η ερευνητική πορεία βασίστηκε κυρίως στις δημοσιεύσεις του κ. Ναούμ Μπακάλη [1], [2] και [3]. Ο προγραμματισμός έγινε σε “γλώσσα” FORTRAN (Compaq Visual Fortran), ο έλεγχος των υπολογισμών έγινε μέσω της MATHEMATICA 9 και η συγγραφή της εργασίας καθώς και της αντίστοιχης παρουσίασης, έγινε στη $\text{\LaTeX} 2_{\epsilon}$. Τα σχήματα πάρθηκαν απο διάφορα σχετικά βιβλία που συμπεριλαμβάνονται στη βιβλιογραφία καθώς και με χρήση της MATHEMATICA 9 και του Origin 8. Τέλος, ως πρότυπο για τη δομή της εργασίας χρησιμοποιήθηκαν οι προπτυχιακές σημειώσεις/βιβλίο απο το μάθημα “Φυσικοχημεία 5” [6] του τμήματος Χημείας ΕΚΠΑ.



Εθνικό Ίδρυμα Ερευνών,
Ινστιτούτο Θεωρητικής και Φυσικής Χημείας

Θεμελιώδεις Σταθερές

Ταχύτητα του φωτός	c	$299792458 \text{ ms}^{-1}$
Στοιχειώδες φορτίο	e	$1.602176462 \times 10^{-19} \text{ C}$
Μάζα ηλεκτρονίου	m_e	$9.10938188 \times 10^{-31} \text{ kg}$
Μάζα πρωτονίου	m_p	$1.67262158 \times 10^{-27} \text{ kg}$
Σταθερά Planck	h	$6.62606876 \times 10^{-39} \text{ Js}$
	$\hbar = \frac{h}{2\pi}$	1.054571596 Js
Ηλεκτρική διαπερατότητα	ϵ_0	$8.854187817 \times 10^{-12} \text{ Fm}^{-1}$
Μαγνητική διαπερατότητα	$\mu_0 = \frac{1}{\epsilon_0 c^2}$	$4\pi \times 10^{-7} \text{ NA}^{-2}$
Μαγνητόνι Bohr	$\mu_B = \frac{e\hbar}{2m_e}$	$927.400899 \times 10^{-26} \text{ JT}^{-1}$
Πυρηνική μαγνητόνι	$\mu_N = \frac{e\hbar}{2m_p}$	$5.05078317 \times 10^{-27} \text{ JT}^{-1}$
Σταθερά λεπτής υφής	$\alpha = \frac{e^2}{4\pi\epsilon_0\hbar c}$	$7.297352533 \times 10^{-3}$
Παράγοντας Lande ηλεκτρονίου	g_e	-2.0023193043737
Παράγοντας Lande πρωτονίου	g_p	5.585694675

Περιεχόμενα

Περιεχόμενα	xvi
I Βασική Θεωρία	1
1 Εισαγωγή	3
1.1 Εισαγωγή	3
1.2 Σκοπός της εργασίας	3
1.3 Περιγραφή του αλγόριθμου	4
1.3.1 Συνοπτικά	5
2 Μοριακή Χαμιλτονιανή	6
2.1 Εισαγωγή	6
2.2 Η εξίσωση Schrödinger πολλών σωματιδίων	6
2.3 Η προσέγγιση Born-Oppenheimer	9
2.3.1 Ηλεκτρονιακή Χαμιλτονιανή	10
2.3.2 Πυρηνική Χαμιλτονιανή	12
II Θεωρητικό Υπόβαθρο	15
3 Μονοηλεκτρονιακά Διατομικά Μόρια	17
3.1 Εισαγωγή	17
3.2 Το Μοριακό Ιόν Υδρογόνου H_2^+	17
3.2.1 Προσεγγιστική λύση μέσω Θεωρίας Μεταβολών . . .	19

3.2.2	Αναλυτική λύση	22
4	Πολυηλεκτρονιακά Διατομικά Μόρια	33
4.1	Εισαγωγή	33
4.2	Το Μόριο του Υδρογόνου H_2	33
4.2.1	Γινόμενα Hartree	35
4.2.2	Ορίζουσες Slater	37
4.2.3	Μητροστοιχεία	39
4.2.4	Συμπεριφορά στο όριο $R \rightarrow \infty$	41
4.3	Αλληλεπίδραση Απεικονίσεων	42
4.3.1	Πλήρης κυματοσυνάρτηση CI	43
4.3.2	Συντελεστές <i>Lagrange</i>	44
III	Κυρίως Μέρος	50
5	Ανάπτυξη Μητροστοιχείων	52
5.1	Εισαγωγή	52
5.2	Αναμενόμενη Τιμή	52
5.2.1	Διαγωνοποίηση της \mathcal{H}_{pq}	53
5.2.2	Ανάπτυξη των οριζουσών <i>Slater</i>	55
5.3	Επικάλυψη	57
5.4	Υπολογισμός Ολοκληρωμάτων	58
5.4.1	OEDO	58
5.4.2	Ολοκληρώματα Επικαλύψεως	61
5.4.3	Μονοηλεκτρονιακά Ολοκληρώματα	61
5.4.4	Διηλεκτρονιακά Ολοκληρώματα	65
5.5	Αντί Επιλόγου	80
IV	Παραρτήματα	81
A'	Σφαιροειδείς Συντεταγμένες και η Εξίσωση Schrödinger	83

A.1	Σφαιροειδείς συντεταγμένες	83
A.1.1	Γεωμετρία Έλλειψης και Υπερβολής	83
A.1.2	Επιμήκεις και Πεπλατυσμένες Σφαιροειδείς Συντεταγμένες	84
A.1.3	Σχέσεις Μετασχηματισμού	86
A.2	Η εξίσωση <i>Schrödinger</i> σε σφαιροειδείς συντεταγμένες . .	87
A.2.1	Διαχωρισμός των μεταβλητών	89
A.3	Μελέτη Διαφορικών Εξισώσεων <i>OEDM</i>	91
A.3.1	Αναγωγή στην διαφορική εξίσωση σφαιροειδούς . .	93
B'	Συναρτήσεις Γενικευμένου Σφαιροειδούς	96
B.1	Γενικευμένη Διαφορική Εξίσωση Σφαιροειδούς	96
B.1.1	Εισαγωγή	96
B.1.2	Επίλυση εξίσωσης σφαιροειδούς μέσω δυναμοσειρών	98
B.1.3	Οι λύσεις των Hylleraas και Jaffé	101
B.2	Ακτινική Εξίσωση	104
B.3	Γωνιακή Εξίσωση	104
V	FORTRAN Subroutines	107
Γ'	Κεντρικά Προγράμματα	108
Γ.1	Κύριο πρόγραμμα	108
Γ.2	Βελτιστοποίηση των z^*	115
Γ.2.1	Συνάρτηση που καλεί η Γ.2	123
Γ.3	CI	158
Γ.3.1	Αναμενόμενη τιμή και επικάλυψη	164
Γ.3.2	Διαγωνοποίηση	188
Γ.4	Συναρτησοειδές \mathcal{F}	201
Γ.5	Δημιουργία Κατώτερων Καταστάσεων	202

Δ' Matrix Elements	203
Δ'.1 Επικαλύψεως	203
Δ'.1.1 Γωνιακά Ολοκληρώματα	209
Δ'.1.2 Ακτινικά Ολοκληρώματα	215
Δ'.2 Μονοπλεκτρονιακά	217
Δ'.2.1 Γωνιακά Ολοκληρώματα	224
Δ'.2.2 Ακτινικά Ολοκληρώματα	231
Δ'.3 Διπλεκτρονιακά	235
Δ'.3.1 Ακτινικά Ολοκληρώματα	244
Δ'.3.2 Ακτινικά Ολοκληρώματα	253
Δ'.4 Προγράμματα Ολοκλήρωσης	288
 Κατάλογος σχημάτων	 313
 Βιβλιογραφία	 315

Μέρος Ι
Βασική Θεωρία

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγή

Σύμφωνα με το θεώρημα των *Hylleraas, Undheim* και *MacDonald* (HUM) [35], οι λύσεις της χρονοανεξάρτητης εξίσωσης *Schroedinger* για δέσμιες διηγερμένες καταστάσεις, λαμβάνονται ως ανώτερες ρίζες μίας χαρακτηριστικής εξίσωσης. Όμως, οι παραγόμενες αυτές λύσεις, παρόλο που δίνουν τη σωστή ενέργεια, δεν είναι κάθετες στην θεμελιώδη κατάσταση και γενικά στις κατώτερές τους καταστάσεις· δηλ. $|\langle \Psi_1 | \Psi_0^{HUM} \rangle|^2 \neq 0$. Αποτέλεσμα αυτού είναι ουσιαστικά να χειροτερεύει η περιγραφή όλων των υποκειμένων καταστάσεων όταν βρίσκουμε την ενέργεια της επόμενης.

1.2 Σκοπός της εργασίας

Το ζητούμενο σε αυτή την εργασία είναι να υπολογίσουμε τις σωστές ενέργειες των διηγερμένων καταστάσεων, ενώ συγχρόνως θα βρούμε τις σωστές καταστάσεις. Αυτό το επιτυγχάνουμε με τη μέθοδο μεταβολών για διηγερμένες κυματοσυναρτήσεις, όπου ελαχιστοποιούμε το συναρτησοειδές F [1], το οποίο έχει τοπικό ελάχιστο στις δέσμιες διηγερμένες καταστάσεις μιας μη εκφυλισμένης Χαμιλτονιανής.

Επιπλέον, σημαντικό πλεονέκτημα αυτής της μεθόδου είναι ότι η ακρίβεια του υπολογισμού δέν εξαρτάται από τη ποιότητα της προσέγγισης των κατωτέρων καταστάσεων.

Για τον υπολογισμό του συναρτησοειδούς αυτού, είναι απαραίτητες οι αναμενόμενες τιμές των κατώτερων καταστάσεων, της ζητούμενης καθώς και τα στοιχεία μήτρας μεταξύ κατώτερων καταστάσεων και της ζητούμενης:

$$\mathcal{F}_n[\phi_0, \phi_1, \dots; \phi_n] = E[\phi_n] + 2 \frac{\sum_{i < n} \frac{\left(E[\phi_n] \langle \phi_i | \phi_n \rangle - \langle \phi_i | \hat{H} | \phi_n \rangle \right)^2}{E[\phi_n] - E[\phi_i]}}{1 - \sum_{i < n} \langle \phi_i | \phi_n \rangle^2} \quad (1.1)$$

Βασικός στόχος μας είναι να αναπτύξουμε όλες τις παραπάνω ώστε να τις υπολογίσουμε και τελικά να τις αντικαταστήσουμε στο συναρτησοειδές.

1.3 Περιγραφή του αλγόριθμου

Αφού το συναρτησοειδές \mathcal{F}_n περιέχει πέραν του $\langle \phi_n | \hat{H} | \phi_n \rangle$ και την εξίσωση του Schroedinger πολλαπλασιασμένη αριστερά με ϕ_i , $i < n$, δηλ. $\langle \phi_i | \hat{H} - E_n | \phi_n \rangle > 0$, ως θετικό "διορθωτικό" όρο, άρα στο ελάχιστο της F , όπου $\phi_n = \psi_n$ (η ακριβής άγνωστη διεγερμένη κατάσταση), δηλ. η εξ. του Schroedinger ικανοποιείται, ο "διορθωτικός" όρος θα μηδενίζεται, και άρα αρκεί να βρίσκουμε σε κάθε μεταβολή των μη γραμμικών παραμέτρων z^* , τους γραμμικούς συντελεστές του αναπτύγματος CI που ελαχιστοποιούν τον θετικό "διορθωτικό" όρο, λύνοντας την χαρακτηριστική (secular) εξίσωση ως προς αυτούς για κάθε δοκιμαστικό z^* .

Άρα, για κάθε z^* , λύνουμε την χαρακτηριστική εξίσωση προς εύρεση των συντελεστών c_i , της επιθυμητής "ρίζας" και της αντίστοιχης κυματοσυνάρτησης, τα βάζουμε στο \mathcal{F}_n , χρησιμοποιώντας προϋπάρχουσες

ϕ_i , $i < n$, από προηγούμενους υπολογισμούς, και συνεχίζουμε να μεταβάλλουμε τα z^* μέχρις ελαχιστοποίησης του \mathcal{F}_n .

1.3.1 Συνοπτικά

Δεδομένης, της πολυπλοκότητας του αλγόριθμου, αναφέρω εδώ σταδιακά τα βήματα της διαδικασίας:

1. Εισάγουμε τα N_{el} , Z_A , Z_B , R , N_{lwr} , καθώς και τα αρχικά z_A^* , z_B^{*1} (initial guess) και την ηλεκτρονιακή απεικόνιση του διατομικού μορίου.
2. Από αυτά, μέσω της επίλυσης των διαφορικών εξισώσεων και των παραγόμενων αναδρομικών σχέσεων, εξάγονται οι διάφορες παράμετροι $\{f_j\}$, $\{g_j\}$, p , ...
3. Βάσει αυτών, υπολογίζονται τα ακτινικά και γωνιακά ολοκληρώματα, καθώς και τα
4. διάφορα μπροστοιχεία, στα οποία εμπεριέχονται.
5. Κατόπιν, από τα παραπάνω, υπολογίζονται οι αναμενόμενες τιμές και η επικάλυψη,
6. τα οποία αντικαθίστανται στο συναρτησοειδές \mathcal{F} .
7. Η όλη διαδικασία επαναλαμβάνεται για διαφορετικά πλέον z^* και κατ' επέκταση OEDM, μέχρι να βρεθεί η ελάχιστη τιμή του \mathcal{F} .
8. Οι καινούριες καταστάσεις, που βρίσκονται από κάθε υπολογισμό, αποθηκεύονται ώστε να χρησιμοποιηθούν την επόμενη φορά ως κατώτερες καταστάσεις, για την βελτιστοποίηση του \mathcal{F} .

¹Όπου N_{el} : ο αριθμός των ηλεκτρονίων, Z_I : το πυρηνικό φορτίο του πυρήνα I , z_I^* : το ενεργό πυρηνικό φορτίο του πυρήνα I , R : η διαπυρηνική απόσταση, N_{lwr} : το πλήθος των κατωτέρων καταστάσεων.

Κεφάλαιο 2

Μοριακή Χαμιλτονιανή

2.1 Εισαγωγή

Περιγράφοντας συνοπτικά τα βασικά σημεία της θεωρίας, εκκινούμε από την εξίσωση *Schrödinger* και την αντίστοιχη μοριακή Χαμιλτονιανή και καταλήγουμε, μέσω της προσέγγισης *Born-Oppenheimer*, στην ηλεκτρονιακή Χαμιλτονιανή.

2.2 Η εξίσωση *Schrödinger* πολλών σωματιδίων

Για τη περιγραφή ενός συστήματος που περιέχει N_e ηλεκτρόνια με συντεταγμένες $\mathbf{r} \equiv \{\mathbf{r}_i\}$ και N_i άτομα στα σημεία $\mathbf{R} \equiv \{\mathbf{R}_I\}$, αρχικά πρέπει να επιλύσουμε την χρονοανεξάρτητη εξίσωση *Schrödinger* :

$$\hat{H}\Psi(\mathbf{r}, \mathbf{R}) = E\Psi(\mathbf{r}, \mathbf{R}) \quad (2.1)$$

οπου Ψ είναι η κυματοσυνάρτηση και E η ολική ενέργεια. Η Χαμιλτονιανή \hat{H} (2.3) για ένα οποιοδήποτε μοριακό σύστημα γράφεται [37]

$$\hat{H} = \hat{H}_0 + \hat{H}_f + \hat{H}_{hf} \quad (2.2)$$

όπου \hat{H}_0 : η μη σχετικιστική Χαμιλτονιανή, στην οποία και θα επικεντρωθούμε, \hat{H}_f : η Χαμιλτονιανή λεπτής υφής και \hat{H}_{hf} : η Χαμιλτονιανή υπέρλεπτης υφής. Η \hat{H}_0 αποτελείται από τη κινητική ενέργεια των πυρήνων (\hat{T}_N) και των ηλεκτρονίων (\hat{T}_e), την ηλεκτροστατική αλληλεπίδραση μεταξύ των ηλεκτρονίων και των πυρήνων (\hat{V}_{eN}), καθώς και την ηλεκτροστατική άπωση μεταξύ των πυρήνων (\hat{V}_{NN}) και μεταξύ των ηλεκτρονίων (\hat{V}_{ee}).

$$\hat{H} = \hat{T}_N(\mathbf{R}) + \hat{T}_e(\mathbf{r}) + \hat{V}_{eN}(\mathbf{r}; \mathbf{R}) + \hat{V}_{NN}(\mathbf{R}) + \hat{V}_{ee}(\mathbf{r}), \quad (2.3)$$

με

$$\hat{T}_N = - \sum_I^{N_N} \frac{\hbar^2}{2M_I} \nabla_I^2, \quad \hat{T}_e = - \sum_i^{N_e} \frac{\hbar^2}{2m_e} \nabla_i^2 \quad (2.4)$$

$$\hat{V}_{NN} = \frac{1}{2} \sum_I^{N_N} \sum_{J \neq I}^{N_N} \frac{e^2}{4\pi\epsilon_0} \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|}, \quad \hat{V}_{ee} = \frac{1}{2} \sum_i^{N_e} \sum_{j \neq i}^{N_e} \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}, \quad (2.5)$$

$$\hat{V}_{eN} = - \sum_i^{N_e} \sum_I^{N_N} \frac{e^2}{4\pi\epsilon_0} \frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} \quad (2.6)$$

όπου M_I και m_e είναι οι μάζες των πυρήνων και των ηλεκτρονίων αντίστοιχα και Z_I τα πυρηνικά φορτία.

Οι εναπομείναντες όροι, που επιβάλλονται σε μεγάλους ατομικούς αριθμούς (Z) (με την εμφάνιση σχετικιστικών επιδράσεων), υποδιαιρούνται περαιτέρω: ο \hat{H}_f σε : \hat{V}_m , \hat{V}_{LS} ¹, \hat{V}_{SS} και \hat{V}_D , ενώ ο \hat{H}_{hf} σε : \hat{H}_{LL} ,

¹Στη πραγματικότητα, σε ένα σύστημα πολλών ηλεκτρονίων, όπως ένα διατομικό μόριο, αυτά θα αλληλεπιδρούν μεταξύ τους με όλους τους δυνατούς τρόπους: δηλ. θα υπάρχει και σύζευξη των τροχιακών στροφορμών διαφορετικών ηλεκτρονίων (\hat{H}_{LL}). Συνεπώς, συνδυάζοντας όλα τα παραπάνω καταλήγουμε στην παρακάτω Χαμιλτονιανή για τις αλληλεπιδράσεις μαγνητικού διπόλου των ηλεκτρονίων [37]:

$$\hat{H}_{el}^{MD} = \sum_i^{N_e} \sum_{j>i}^{N_e} a_{ij} \hat{\mathbf{l}}_i \cdot \hat{\mathbf{s}}_j + \sum_i^{N_e} \sum_{j>i}^{N_e} b_{ij} \hat{\mathbf{l}}_i \cdot \hat{\mathbf{l}}_j + \sum_i^{N_e} \sum_{j>i}^{N_e} c_{ij} \hat{\mathbf{s}}_i \cdot \hat{\mathbf{s}}_j \quad (2.7)$$

\hat{H}_{SI} και \hat{H}_{FC} ². Συγκεκριμένα, ο 1^{ος} αντιστοιχεί στην στην διόρθωση που επιβάλλει η σχετικιστική μεταβολή της μάζας με την ταχύτητα, ο 2^{ος} στη σύζευξη τροχιάς-*spin*, ο 3^{ος} στη σύζευξη *spin-spin*³, ο 4^{ος} δίνει τη μη μηδενική συνεισφορά των διορθώσεων για τη κατάσταση s ($l = 0$)⁴, ο 5^{ος} περιγράφει την αλληλεπίδραση μεταξύ τροχιακής στροφορμής των ηλεκτρονίων $\hat{\mathbf{l}}_i$ και του πυρηνικού *spin* $\hat{\mathbf{I}}_J$, ο 6^{ος} στην αλληλεπίδραση μεταξύ ηλεκτρονιακού $\hat{\mathbf{s}}_i$ και πυρηνικού *spin* και τέλος ο 7^{ος} στην αλληλεπίδραση επαφής-*Fermi*.

Έτσι, για τους όρους λεπτής υφής έχουμε :

$$\hat{V}_m = \sum_i^{N_e} \frac{\hat{\mathbf{p}}_i^4}{8m^3c^2} = - \sum_i^{N_e} \frac{\hbar^4}{8m^3c^2} \nabla_i^4, \quad (2.8)$$

$$\begin{aligned} \hat{V}_{LS} &= \sum_i^{N_e} \sum_J^{N_N} \frac{\hbar^2}{2m^2c^2} \frac{1}{r_{iJ}} \frac{dV_{iJ}}{dr_{iJ}} \hat{\mathbf{l}}_{iJ} \cdot \hat{\mathbf{s}}_i + \sum_i^{N_e} \sum_{j \neq i}^{N_e} \frac{\hbar^2}{2m^2c^2} \frac{1}{r_{ij}} \frac{dV_{ij}}{dr_{ij}} \hat{\mathbf{l}}_{ij} \cdot (\hat{\mathbf{s}}_i + 2\hat{\mathbf{s}}_j) \\ &= \sum_i^{N_e} \sum_J^{N_N} \frac{\hbar^2 e^2}{8\pi\epsilon_0 m^2 c^2} \frac{Z_N}{r_{iJ}^3} (\hat{\mathbf{r}}_{iJ} \times \hat{\mathbf{p}}_i) \cdot \hat{\mathbf{s}}_i - \sum_i^{N_e} \sum_{j \neq i}^{N_e} \frac{\hbar^2 e^2}{8\pi\epsilon_0 m^2 c^2} \frac{1}{r_{ij}^3} (\hat{\mathbf{r}}_{ij} \times \hat{\mathbf{p}}_i) \cdot (\hat{\mathbf{s}}_i + 2\hat{\mathbf{s}}_j), \end{aligned} \quad (2.9)$$

$$\hat{V}_{SS} = \sum_i^{N_e} \sum_{j \neq i}^{N_e} \frac{\hbar^2 e^2}{8\pi\epsilon_0 m^2 c^2} \left[\frac{\hat{\mathbf{s}}_i \cdot \hat{\mathbf{s}}_j}{r_{ij}^3} - \frac{3(\hat{\mathbf{s}}_i \cdot \hat{\mathbf{r}}_{ij})(\hat{\mathbf{s}}_j \cdot \hat{\mathbf{r}}_{ij})}{r_{ij}^5} \right], \quad (2.10)$$

$$\hat{V}_D = - \sum_i^{N_e} \frac{\hbar^2}{8m^2c^2} \nabla_i^2 V_i, \quad (2.11)$$

²Έχουμε παραλείψει εδώ κάποιους όρους, όπως της αλληλεπίδρασης των πυρήνων μεταξύ τους, της ηλεκτρικής τετραπολικής αλληλεπίδρασης μεταξύ των ηλεκτρονίων και των πυρήνων, καθώς και όλες της ανωτέρας τάξης αλληλεπιδράσεις [38].

³Βέβαια, ο όρος που οφείλεται στη σύζευξη *spin-spin*, στη πράξη είναι πολύ μικρότερος από τον όρο *τροχιάς-spin* για βαρέα άτομα. Αυτό οφείλεται στο γεγονός ότι η αλληλεπίδραση *τροχιάς-spin* αυξάνει γρήγορα με τον ατομικό αριθμό, ενώ η αλληλεπίδραση *spin-spin* είναι ουσιαστικά ανεξάρτητη του Z .

⁴Ο όρος Darwin \hat{V}_D προκύπτει από την τρεμώδη κίνηση του ηλεκτρονίου ("Zitterbewegung"), που μειώνει το ενεργό δυναμικό που αισθάνεται το ηλεκτρόνιο.

ενώ, για τους όρους υπέρλεπτης υφής έχουμε :

$$\hat{H}_{LI} = - \sum_i^{N_e} \sum_J^{N_N} 2g_{IJ}\mu_B\mu_N \frac{\mu_0}{4\pi} \frac{\hat{\mathbf{I}}_J \cdot \hat{\mathbf{l}}_{iJ}}{r_{iJ}^3}, \quad (2.12)$$

$$\hat{H}_{SI} = - \sum_i^{N_e} \sum_J^{N_N} g_s g_{IJ} \mu_B \mu_N \frac{\mu_0}{4\pi} \left[\frac{3(\hat{\mathbf{s}}_i \cdot \hat{\mathbf{r}}_{iJ})(\hat{\mathbf{I}}_J \cdot \hat{\mathbf{r}}_{iJ})}{r_{iJ}^3} - \frac{(\hat{\mathbf{s}}_i \cdot \hat{\mathbf{r}}_{iJ})(\hat{\mathbf{s}}_J \cdot \hat{\mathbf{I}}_J)}{r_{iJ}^5} \right] \quad (2.13)$$

$$\hat{H}_{FC} = - \sum_i^{N_e} \sum_J^{N_N} g_s g_{IJ} \mu_B \mu_N \frac{\mu_0}{4\pi} \frac{8\pi}{3} \hat{\mathbf{I}}_J \cdot \hat{\mathbf{s}}_i \delta(\hat{\mathbf{r}}_{iJ}). \quad (2.14)$$

όπου $\hat{\mathbf{l}}_{iJ}$: η στροφορμή του i ηλεκτρονίου γύρω από το πυρήνα J , $\hat{\mathbf{r}}_{iJ} = \hat{\mathbf{r}}_i - \hat{\mathbf{R}}_J$: η απόσταση του ηλεκτρονίου i από το πυρήνα J και $\hat{\mathbf{p}}_i$: η ορμή του ηλεκτρονίου i .

Στην εργασία αυτή δεν θα ληφθούν υπ' όψιν οι τελευταίοι όροι, επειδή θα επικεντρωθούμε σε διατομικά μόρια που αποτελούνται από άτομα μικρού ατομικού αριθμού.

Στο εξής θα χρησιμοποιούνται ατομικές μονάδες ($e = m_e = \hbar = 4\pi\epsilon_0 = 1$), ως πιο βολικές για κβαντομηχανικά προβλήματα. Το μήκος εκφράζεται σε Bohrs και η ενέργεια σε hartrees ($1 a_{Bohr} = 5.2917 \times 10^{-11} \text{m}, 1 E_h = 2 \text{ Ryd} = 27.2116 \text{eV}$).

Θα εισάγουμε τώρα τις απαραίτητες προσεγγίσεις προς απλοποίηση της αρχικής εξίσωσης Schrödinger (2.1).

2.3 Η προσέγγιση Born-Oppenheimer

Μία από τις πιο σημαντικές προσεγγίσεις είναι η Born-Oppenheimer, η οποία ουσιαστικά δηλώνει πως οι βαθμοί ελευθερίας των ηλεκτρονίων και των πυρήνων μπορούν να αποσυζευχθούν. Βασίζεται στο γεγονός ότι οι μάζες των πυρήνων είναι πολύ μεγαλύτερες από αυτές των ηλεκτρονίων, τα οποία και θα ανταποκρίνονται σχεδόν στιγμιαία σε οποιαδήποτε αλλαγή στο δυναμικό, που οφείλεται στην κίνηση των πυρήνων. Έτσι, μπορούμε να θεωρήσουμε ότι τα ηλεκτρόνια κινούνται

στο πεδίο των ακίνητων πυρήνων ⁵ και κατά συνέπεια η άπωση μεταξύ των πυρήνων είναι σταθερή. Η προσέγγιση αυτή ουσιαστικά είναι εφαρμογή του *Αδιαβατικού Θεωρήματος* [8]:

Θεώρημα 2.3.1. Έστω ότι η Χαμιλτονιανή μεταβάλλεται σταδιακά από μία αρχική μορφή \hat{H}^i σε μία τελική \hat{H}^f , τότε εάν το σωματίδιο βρισκόταν αρχικά στην n -οστή ιδιοκατάσταση της \hat{H}^i , θα μεταφερθεί (υπό την εξίσωση *Schrödinger*) στην n -οστή ιδιοκατάσταση της \hat{H}^f .

Η ολική κυματοσυνάρτηση μπορεί να εκφραστεί ως το γινόμενο του πυρηνικού και του ηλεκτρονιακού μέρους :

$$\Psi(\mathbf{r}, \mathbf{R}) = \Psi_{el}(\mathbf{r}; \mathbf{R})\Psi_{nuc}(\mathbf{R}) \quad (2.15)$$

οπου η ηλεκτρονιακή κυματοσυνάρτηση εξαρτάται από τις θέσεις των πυρήνων μόνο παραμετρικά, δηλ. για κάθε πυρηνική στερεοχημική διαμόρφωση ορίζεται μια διαφορετική κυματοσυνάρτηση.

Εφόσον χωρίσαμε την ολική κυματοσυνάρτηση σε ηλεκτρονιακό και πυρηνικό μέρος, το ίδιο κάνουμε και για την ολική Χαμιλτονιανή: αρχίζουμε εξετάζοντας το ηλεκτρονιακό μέρος.

2.3.1 Ηλεκτρονιακή Χαμιλτονιανή

Οι διάφορες κβαντικές επιδράσεις περιορίζονται στην ηλεκτρονιακή κυματοσυνάρτηση Ψ_{el} , η οποία υπακούει σε μια απλούστερη εξίσωση *Schrödinger* :

$$\hat{H}_{el}\Psi_{el}(\mathbf{r}; \mathbf{R}) = E_{el}^n(\mathbf{R})\Psi_{el}^n(\mathbf{r}; \mathbf{R}) \quad (2.16)$$

⁵ Εξού και η εναλλακτική ονομασία της προσέγγισης : των "παγωμένων πυρήνων"

με

$$\hat{H}_{el} = \hat{T}_e(\mathbf{r}) + \hat{V}_{eN}(\mathbf{r}; \mathbf{R}) + \hat{V}_{ee}(\mathbf{r}) \quad (2.17)$$

Βασιζόμενοι στα προηγούμενα υπολογίζουμε την εξ. (2.3) ως :

$$\hat{H}(\mathbf{r}; \mathbf{R}) = \hat{H}_{el}(\mathbf{r}; \mathbf{R}) + \hat{T}_N(\mathbf{R}) + \hat{V}_{NN}(\mathbf{R}) \quad (2.18)$$

Στα πλαίσια της προσέγγισης Born-Oppenheimer η κινητική ενέργεια των πυρήνων T_N μπορεί να διαχωριστεί από την κίνηση των ηλεκτρονίων, ενώ η άπωση μεταξύ των πυρήνων V_{NN} θα είναι σταθερή για μία συγκεκριμένη χωρική διαμόρφωση \mathbf{R}_I .

Η ολική ενέργεια του πολυσωματιδιακού συστήματος δίνεται από τη σχέση : $E_n(\mathbf{R}_I) = E_{el}^n(\mathbf{R}_I) + V_{NN}(\mathbf{R}_I)$, όπου $E_{el}^n(\mathbf{R}_I)$ είναι το ενεργειακό φάσμα της πολυηλεκτρονιακής Χαμιλτονιανής \hat{H}_{el} (2.17). Στην προσέγγιση Born-Oppenheimer υποθέτουμε ότι η ηλεκτρονιακή συνιστώσα παραμένει στη θεμελιώδη κατάσταση, $n = 0$, οπότε ο όρος ολική ενέργεια (της θεμελιώδους κατάστασης), για μία συγκεκριμένη διαμόρφωση \mathbf{R}_I , αναφέρεται στην :

$$E(\mathbf{R}_I) = E_{el}^0(\mathbf{R}_I) + V_{NN}(\mathbf{R}_I) \quad (2.19)$$

Η επανάληψη των υπολογισμού με διαφορετικές διατάξεις πυρήνων, μας επιτρέπει να σχηματίσουμε την επιφάνεια δυναμικής ενέργειας καθώς και να βρούμε την γεωμετρία ισορροπίας. Για να υπολογίσουμε την Ψ_{el} στην (2.16) υπάρχουν πολλές προσεγγίσεις, π.χ μία απλή είναι η Hartree-Fock, που λαμβάνει υπ' όψιν την αντισυμμετρικότητα της κυματοσυνάρτησης σε εναλλαγή δύο ηλεκτρονίων, αλλά όχι και την ηλεκτρονιακή συσχέτιση, δηλ. την παραμόρφωση των τροχιακών λόγω απώσεως των ηλεκτρονίων, ενώ η αντισυμμετρικοποιημένη πολυηλεκτρονιακή κυματοσυνάρτηση γράφεται ως μία και μοναδική ορίζουσα

Slater, η οποία λύνεται αυτοσυνεπώς (SCF).

Μελετούμε στη συνέχεια τη δεύτερη “συνιστώσα” της ολικής Χαμιλτονιανής, την Πυρηνική Χαμιλτονιανή.

2.3.2 Πυρηνική Χαμιλτονιανή

Εφόσον, έχουμε λύσει το ηλεκτρονιακό πρόβλημα, δηλαδή έχουμε σχηματίσει μια πλήρη ηλεκτρονιακή επιφάνεια δυναμικής ενέργειας για τους πυρήνες (σχ. 2.1, σελ. 13), μπορούμε να προχωρήσουμε και στην επίλυση του προβλήματος για τη κίνηση των πυρήνων, αντικαθιστώντας τις ηλεκτρονιακές συντεταγμένες με τις μέσες⁶ τιμές τους (ως προς την ηλεκτρονιακή κυματοσυνάρτηση Ψ_{el} ⁷). Έτσι, από την (2.18), έχουμε την πυρηνική Χαμιλτονιανή $\hat{H} \equiv \hat{H}_{nuc}$ για την κίνηση των πυρήνων στο μέσο πεδίο των ηλεκτρονίων (σε α.υ):

$$\begin{aligned}
 \hat{H}_{nuc} &= - \sum_I^{N_N} \frac{1}{2M_I} \nabla_I^2 + \left\langle - \sum_i^{N_e} \frac{1}{2} \nabla_i^2 - \sum_i^{N_e} \sum_I^{N_N} \frac{Z_I}{|r_i - \mathbf{R}_I|} \right. \\
 &\quad \left. + \sum_i^{N_e} \sum_{j>i}^{N_e} \frac{1}{|r_i - r_j|} \right\rangle + \sum_I^{N_N} \sum_{J>I}^{N_N} \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} \\
 &= - \sum_I^{N_N} \frac{1}{2M_I} \nabla_I^2 + E_{el}(\{\mathbf{R}_I\}) + \sum_I^{N_i} \sum_{J>I}^{N_i} \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} \\
 &= - \sum_I^{N_N} \frac{1}{2M_I} \nabla_I^2 + E(\{\mathbf{R}_I\}) \tag{2.20}
 \end{aligned}$$

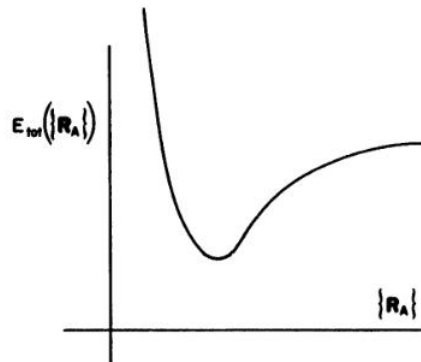
Η $E(\{\mathbf{R}_I\})$ μας δίνει την επιφάνεια δυναμικής ενέργειας στην οποία κινούνται οι πυρήνες (στα πλαίσια της προσέγγισης Born-Oppenheimer) (σχ. 2.2, σελ. 13).

Η επίλυση της πυρηνικής εξίσωσης Schrödinger:

$$\hat{H}_{nuc} \Psi_{nuc}(\{\mathbf{R}_I\}) = E(\{\mathbf{R}_I\}) \Psi_{nuc}(\{\mathbf{R}_I\}) \tag{2.21}$$

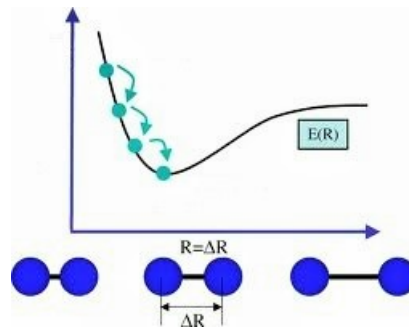
⁶Η προσέγγιση αυτή γίνεται για τους ίδιους λόγους με το ηλεκτρονιακό πρόβλημα: τα ηλεκτρόνια κινούνται πολύ πιο γρήγορα από τους πυρήνες.

⁷ που επιλέγεται αδιαβατική ή μί, ανάλογα με την ταχύτητα του υπό εξέταση φαινομένου



Σχήμα 2.1: Διαγραμματική απεικόνιση του δυναμικού επιφάνειας.

θα μας δώσει τις λύσεις Ψ_{nuc} που περιγράφουν την δονητική, περιστροφική και μεταφορική κίνηση του μορίου, ενώ η E είναι η ολική ενέργεια που περιλαμβάνει την ηλεκτρονιακή, δονητική, περιστροφική και μεταφορική ενέργεια.



Σχήμα 2.2: Σχηματική αναπαράσταση της επιφάνειας δυναμικής ενέργειας, με σταδιακές βελτιστοποιήσεις προς την θέση ισορροπίας.

Στη παρούσα εργασία θα ασχοληθούμε με την επίλυση του ηλεκτρονιακού προβλήματος (2.16) για διηγερμένες καταστάσεις διατομικών μορίων, βελτιστοποιώντας το συναρτησοειδές \mathcal{F}_n (1.1), ως προς τις μη γραμμικές (εκθετικές) παραμέτρους των κυματοσυναρτήσεων που εμφανίζονται ως “ενεργά” πυρηνικά φορτία.

Συνοψίζοντας, η προσέγγιση Born-Oppenheimer συνίσταται σε 3 επιμέρους στάδια :

1. Θεωρούμε ότι η ολική κυματοσυνάρτηση του μορίου μπορεί να γραφεί ως γινόμενο μιας ηλεκτρονιακής και μίας πυρηνικής κυματοσυναρτήσεως (2.15).
2. Λύνουμε το ηλεκτρονιακό πρόβλημα (2.16), θεωρώντας τους πυρήνες παγωμένους στις διάφορες θέσεις \mathbf{R}_I , και λαμβάνουμε την ηλεκτρονιακή ενέργεια $E_{el}^n(\mathbf{R}_I)$ ως συναρτηση των \mathbf{R}_I .
3. Λύνουμε την πυρηνική εξίσωση (2.21) Schrödinger με τη Χαμιλτονιανή (2.18) θεωρώντας την $E_{el}^n(\mathbf{R}_I)$ ως την δυναμική ενέργεια των πυρηνών.

Στο επόμενο μέρος θα χρησιμοποιήσουμε την ηλεκτρονιακή Χαμιλτονιανή που βρήκαμε παραπάνω, ως μοντέλο για την περιγραφή του ηλεκτρονιακού προβλήματος σε μονο- και πολυ-ηλεκτρονιακά διατομικά μόρια.

Μέρος II
Θεωρητικό Υπόβαθρο

Κεφάλαιο 3

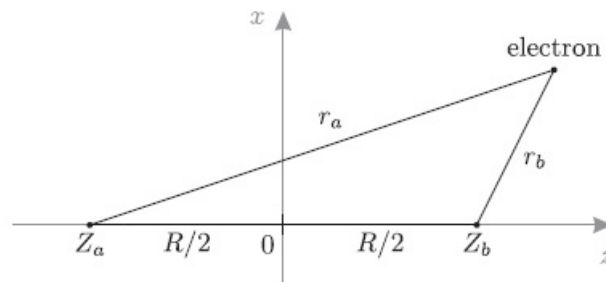
Μονοηλεκτρονιακά Διατομικά Μόρια

3.1 Εισαγωγή

Η μελέτη του του μονοηλεκτρονιακού διατομικού μορίου H_2^+ κρίνεται απαραίτητη για την καλύτερη κατανόηση της παρούσας εργασίας, εφόσον αποτελεί “πρότυπο” για τα μονοηλεκτρονιακά διατομικά μόρια, στις κυματοσυναρτήσεις των οποίων και θα βασιστούμε για τη πραγματοποίηση του υπολογισμού μας. Θα δούμε περιληπτικά την επίλυση του προβλήματος μέσω της *κβαντικής θεωρίας μεταβολών*, ενώ θα σταθούμε περισσότερο στην αναλυτική λύση του προβλήματος· οι λύσεις που θα προκύψουν θα είναι και οι κυματοσυναρτήσεις που θα χρησιμοποιήσουμε για την επίλυση πολυλοκότερων προβλημάτων, και την εύρεση των διηγευμένων καταστάσεων καθώς και των ιδιοτήτων τους.

3.2 Το Μοριακό Ιόν Υδρογόνου H_2^+

Το H_2^+ έχει χρησιμοποιηθεί ως πρότυπο για την εισαγωγή της έννοιας του μοριακού τροχιακού η οποία είναι και η βάση για τη μελέτη



Σχήμα 3.1: Ηλεκτρόνιο δέσιμο από το πεδίο των δύο πυρήνων Η.

πολυηλεκτρονιακών μοριακών συστημάτων. Είναι το απλούστερο μοριακό σύστημα και αποτελείται από 2 πυρήνες Η (2 πρωτόνια) και ένα ηλεκτρόνιο. Πειραματικά έχουν μετρηθεί η απόσταση $\mathbf{R}_{\text{H-H}} = 2.00 \text{ bohr}$ και ενέργεια συνδέσεως $D_e = 2.79 \text{ eV} = 64.3 \text{ Kcal/mol}$. Στα παρακάτω υιοθετούμε την προσέγγιση Born-Oppenheimer “βλ. τμήμα 2.3 στη σελ. 9.”, δηλ. οι πυρήνες θεωρούνται “παγωμένοι” σε κάποια απόσταση \mathbf{R} . Από την εξ. (2.17) η ηλεκτρονιακή Χαμιλτονιανή του συστήματος γράφεται, εκπεφρασμένη σε ατομικές μονάδες (a.u.) :

$$\hat{H}_{el} = -\frac{1}{2}\nabla^2 - \frac{1}{|\mathbf{r} - \mathbf{R}_a|} - \frac{1}{|\mathbf{r} - \mathbf{R}_b|} + \frac{1}{|\mathbf{R}_a - \mathbf{R}_b|} \quad (3.1)$$

όπου έχουμε κατά σειρά :

- την κινητική ενέργεια του ηλεκτρονίου,
- την δυναμική ενέργεια λόγω έλξεως του ηλεκτρονίου από τον πυρήνα a,
- την δυναμική ενέργεια λόγω έλξεως του ηλεκτρονίου από τον πυρήνα b,
- την δυναμική ενέργεια λόγω απώσεως των πυρήνων $\frac{1}{\mathbf{R}}$, η οποία είναι σταθερά.

Η επίλυση της ηλεκτρονιακής εξίσωσης Schrödinger (2.16) (σελ. 10) γίνεται προσεγγιστικά καθώς και αναλυτικά. Ξεκινάμε, με την περιγραφή της προσεγγιστικής λύσης, για λόγους πληρότητας του κεφαλαίου.

3.2.1 Προσεγγιστική λύση μέσω Θεωρίας Μεταβολών

Στη θεωρία μεταβολών, αρχικά “μαντεύουμε” τη μορφή της κυματοσυναρτήσεως, προσεγγίζοντάς την με μια εύλογη δοκιμαστική κυματοσυνάρτηση.

Θεμελειώδης Κατάσταση

Αν χρησιμοποιήσουμε δοκιμαστική κυματοσυνάρτηση της μορφής [6]:

$$\Psi = c_a 1s_a + c_b 1s_b = c_a \frac{1}{\sqrt{\pi}} e^{k|r-\mathbf{R}_a|} + c_b \frac{1}{\sqrt{\pi}} e^{k|r-\mathbf{R}_b|} \quad (3.2)$$

η αναμενόμενη τιμή της ενέργειας θα είναι, βάσει του λόγου Rayleigh-Ritz [5]:

$$\begin{aligned} E &= \frac{\langle \Psi | \hat{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle} \\ &= \frac{|c_a|^2 \langle 1s_a | \hat{H} | 1s_a \rangle + |c_b|^2 \langle 1s_b | \hat{H} | 1s_b \rangle + c_a^* c_b \langle 1s_a | \hat{H} | 1s_b \rangle + c_a c_b^* \langle 1s_b | \hat{H} | 1s_a \rangle}{|c_a|^2 \langle 1s_a | 1s_a \rangle + |c_b|^2 \langle 1s_b | 1s_b \rangle + c_a^* c_b \langle 1s_a | 1s_b \rangle + c_a c_b^* \langle 1s_b | 1s_a \rangle} \end{aligned}$$

όμως $c_a, c_b \in \mathbb{R}$, ο \hat{H} είναι Ερμιτιανός τελεστής, ενώ τα υδρογονοειδή ατομικά τροχιακά είναι κανονικοποιημένα $\langle 1s_i | 1s_i \rangle = 1$, με $i = \{a, b\}$ άρα

$$E = \frac{c_a^2 \langle 1s_a | \hat{H} | 1s_a \rangle + c_b^2 \langle 1s_b | \hat{H} | 1s_b \rangle + 2c_a c_b \langle 1s_a | \hat{H} | 1s_b \rangle}{c_a^2 + c_b^2 + 2c_a c_b \langle 1s_a | 1s_b \rangle} \quad (3.3)$$

Όπως θα δούμε στο κεφάλαιο 4, για να ελαχιστοποιήσουμε ένα συναρτησιοειδές το οποίο υπόκειται σε κάποια εξίσωση δεσμού [31], αρκεί να ελαχιστοποιήσουμε την έκφραση

$$\mathcal{L} = \langle \Psi | \hat{H} | \Psi \rangle - E (\langle \Psi | \Psi \rangle - 1) \quad (3.4)$$

όπου E συντελεστής *Lagrange*, εφαρμόζοντας τις συνθήκες

$$\frac{\partial \mathcal{L}}{\partial c_i} = 0, \quad i = \{a, b\}. \quad (3.5)$$

Μετά από κάποιες πράξεις καταλήγουμε σε ένα σύστημα δύο γραμμικών ομογενών εξισώσεων

$$\dots \Rightarrow \begin{cases} (H_{aa} - E)c_a + (H_{ab} - ES_{ab})c_b = 0, \\ (H_{ab} - ES_{ab})c_a + (H_{aa} - E)c_b = 0. \end{cases} \quad (3.6)$$

όπου $S_{ab} = S_{ba} = \langle 1s_a | 1s_b \rangle = \langle 1s_b | 1s_a \rangle$, $H_{aa} = H_{bb} = \langle 1s_a | \hat{H} | 1s_a \rangle = \langle 1s_b | \hat{H} | 1s_b \rangle$, $H_{ab} = H_{ba} = \langle 1s_a | \hat{H} | 1s_b \rangle = \langle 1s_b | \hat{H} | 1s_a \rangle$ τα διάφορα μητροστοιχεία.

Για να μίν έχει τετρωμένη λύση το σύστημα (3.6) ($c_a = c_b = 0$), πρέπει η ορίζουσα των συντελεστών των αγνώστων να είναι μηδέν :

$$\begin{vmatrix} H_{aa} - E & H_{ab} - ES_{ab} \\ H_{ab} - ES_{ab} & H_{aa} - E \end{vmatrix} = 0 \quad (3.7)$$

όπου καταλίγουμε σε μια αλγεβρική εξίσωση 2ου βαθμού, οι λύσεις της οποίας είναι οι ιδιοτιμές της ενέργειας (σχ. 3.2, σελ. 21) :

$$E_{\pm} = \frac{H_{aa} \pm H_{ab}}{1 \pm S_{ab}} \quad (3.8)$$

Θέτοντας τις ιδιοτιμές (3.8) στις (3.6) λαμβάνουμε τα αντίστοιχα ιδιοδιανύσματα :

I) για $E = E_+$: $c_a = c_b$ και $\Psi_+ = c_a(1s_a + 1s_b)$ και κανονικοποιώντας

$$\langle \Psi_+ | \Psi_+ \rangle = 1 \Rightarrow c_a = \frac{1}{\sqrt{2(1 + S_{ab})}} \quad (3.9)$$

II) για $E = E_-$: $c_a = -c_b$ και $\Psi_- = c_a(1s_a - 1s_b)$ και κανονικοποιώντας

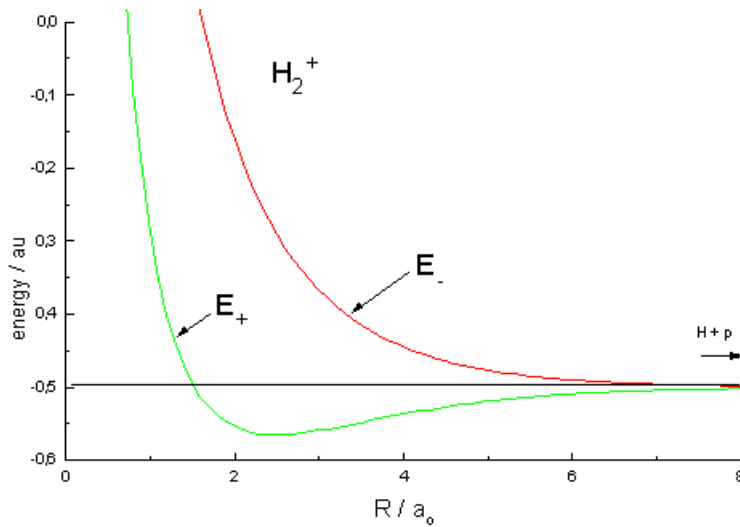
$$\langle \Psi_- | \Psi_- \rangle = 1 \Rightarrow c_a = \frac{1}{\sqrt{2(1 - S_{ab})}} \quad (3.10)$$

Συνολικά έχουμε :

$$\Psi_{\pm} = \frac{1s_a \pm 1s_b}{\sqrt{2(1 \pm S_{ab})}}, \quad E_{\pm} = \frac{H_{aa} \pm H_{ab}}{1 \pm S_{ab}} \quad (3.11)$$

Οι συναρτήσεις Ψ_{\pm} ονομάζονται *μοριακά τροχιακά* κατ' αναλογία με τα *ατομικά τροχιακά* του υδρογόνου. Απο αυτά προκύπτουν, πολλαπλασιαζόμενα με τις συναρτήσεις $\sin \alpha$ ή β , τα *spin-orbitals* :

$$\Phi_{\pm} = \Psi_{\pm} \alpha \quad \text{και} \quad \bar{\Phi}_{\pm} = \Psi_{\pm} \beta \quad (3.12)$$



Σχήμα 3.2: Γραφική παράσταση της ενέργειας συναρτήσει της διαπυρηνικής απόστασεως.

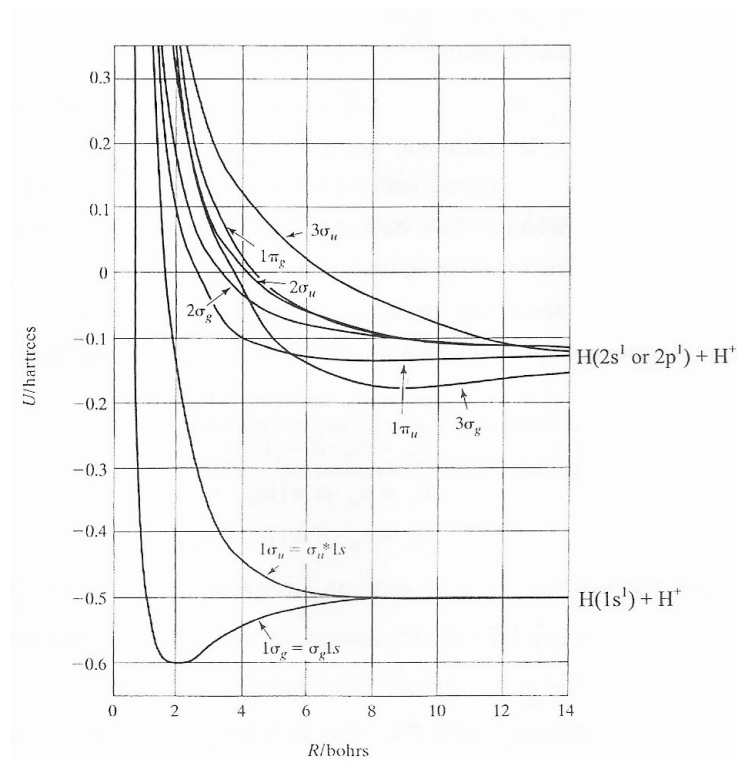
και οι αντίστοιχοι φασματοσκοπικοί όροι που προκύπτουν για το μόριο, θα είναι : ${}^2\Sigma_g^+$ (Φ_+ ή $\bar{\Phi}_+$) και ${}^2\Sigma_u^+$ (Φ_- ή $\bar{\Phi}_-$) όπου έχουμε διπλό εκφυλισμό ως προς το spin $2S + 1 = 2\frac{1}{2} + 1 = 2$ (διπλή κατάσταση).

Διηγερμένες Καταστάσεις

Η αρχική δοκιμαστική κυματοσυνάρτηση (3.2) (σελ.19) μπορεί να γραφεί ως γραμμικός συνδυασμός ατομικών τροχιακών $2s$ και $2p$, επιπλέον των $1s$:

$$\begin{aligned} \Psi = & c_1(1s_a \pm 1s_b) + c_2(2s_a \pm 2s_b) + c_3(2p_{z,a} \pm 2p_{z,b}) \\ & + c_4(2p_{x,a} \pm 2p_{x,b}) + c_5(2p_{y,a} \pm 2p_{y,b}) \end{aligned} \quad (3.13)$$

Βάσει αυτής της κυματοσυνάρτησης προκύπτουν 8 καταστάσεις ($1\sigma_g, 1\sigma_u, 1\pi_g, 1\pi_u, 2\sigma_g, 2\sigma_u, 3\sigma_g, 3\sigma_u$), εκ των οποίων οι δύο είναι διπλά εκφυλισμένες ($1\pi_g, 1\pi_u$), οπότε έχουμε συνολικά 10, δηλ. όσα και τα ατομικά τροχιακά που χρησιμοποιήθηκαν (σχ. 3.3, σελ. 22).



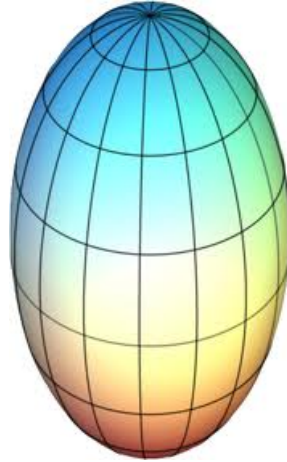
Σχήμα 3.3: Ενέργειες των διηγεμένων καταστάσεων συναρτήσει της διαπυρηνικής απόστασης R .

Κατόπιν, προχωρούμε στην περιγραφή της αναλυτικής επίλυσης του ιδίου προβλήματος, όπου η μόνη προσέγγιση που θα κάνουμε είναι στο επίπεδο των συνεχόμενων κλασμάτων.

3.2.2 Αναλυτική λύση

Το παρόν υποκεφάλαιο είναι αρκετά σημαντικό, εφόσον από αυτό θα προκύψουν οι αναλυτικές λύσεις της εξίσωσης Schrödinger για τα OEDM, που θα χρησιμοποιήσουμε στον υπολογισμό.

Εφόσον το μόριο δεν έχει σφαιρική συμμετρία (όπως έχει το άτομο του H) αλλά μάλλον “ωσειδί” γεωμετρία (σχ. 3.4, σελ. 23) χρησιμοποι-

Σχήμα 3.4: Γεωμετρία του H_2^+ .

ούμε συντεταγμένες επιμήκους σφαιροειδούς¹ (prolate spheroidal, βλ. Α').

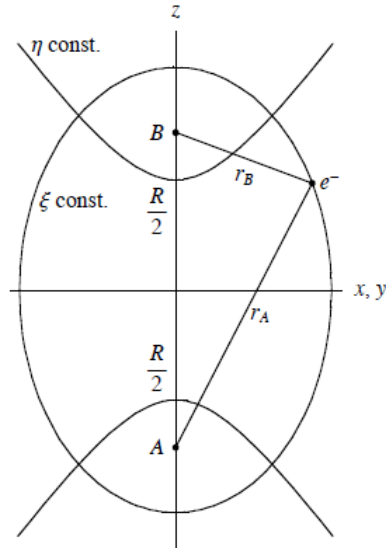
Επιτυγχάνουμε έτσι διαχωρισμό των μεταβλητών της (χρονοανεξάρτητης) εξίσωσης Schrödinger :

$$\hat{H}\Psi(\mathbf{r}) = E\Psi(\mathbf{r}) \Rightarrow \left(\frac{1}{2}\nabla^2 + \frac{1}{r_A} + \frac{1}{r_B} + E \right) \Psi(\mathbf{r}) = 0 \quad (3.14)$$

όπου $r_i = |\mathbf{r} - \mathbf{R}_i|$, $\{i = A, B\}$, τα A, B είναι οι εστίες της έλλειψης (οι θέσεις των 2 πυρήνων) και βρίσκονται επι του άξονα z (σχ. 3.5, σελ. 24) στα σημεία $z = \mp \frac{R}{2}$, $d = \frac{R}{2}$ και τέλος $E \equiv E_{el} < 0$ είναι η ηλεκτρονιακή ενέργεια της δέσμιας κατάστασης. Στην εξ. (3.14) έχουμε παραλείψει τον σταθερό όρο $\frac{1}{R}$, εφόσον αν προσθέσουμε οποιαδήποτε σταθερά σε ένα τελεστή, αυτή απλά προστίθεται στις ιδιοτιμές του, χωρίς να επηρεάζει τις ιδιοσυναρτήσεις του². Οι σφαιροειδείς συντεταγμένες,

¹Απο εδώ και πέρα οι συντεταγμένες επιμήκους σφαιροειδούς θα αναφέρονται ως συντεταγμένες σφαιροειδούς, για λόγους συντομίας.

²Έτσι η $E(\mathbf{R})$ διαφέρει από την συνολική μοριακή ενέργεια $W(\mathbf{R})$ μόνο κατά τον όρο $\frac{1}{R}$, που οφείλεται στην αλληλεπίδραση των πυρήνων. Αυτή είναι μια σημαντική διαφορά εφόσον $\lim_{R \rightarrow 0} E(\mathbf{R}) = 0$, ενώ $\lim_{R \rightarrow 0} W(\mathbf{R}) = \infty$



Σχήμα 3.5: Διαγραμματική απεικόνιση του H_2^+ .

λοιπόν, ορίζονται (παρ. Α', σελ. 87) ως εξής :

$$\xi = \frac{r_A + r_B}{R}, \quad \eta = \frac{r_A - r_B}{R}, \quad \phi = \tan^{-1}\left(\frac{y}{x}\right) \quad (3.15)$$

με $1 \leq \xi \leq \infty$, $-1 \leq \eta \leq 1$ και $0 \leq \phi \leq 2\pi$

Απο τις (3.15) προκύπτει ότι :

$$r_A = \frac{R(\xi + \eta)}{2} \Rightarrow \frac{1}{r_A} = \frac{2}{R(\xi + \eta)} \text{ και } r_B = \frac{R(\xi - \eta)}{2} \Rightarrow \frac{1}{r_B} = \frac{2}{R(\xi - \eta)} \quad (3.16)$$

ενώ η Λαπλασιανή σε σφαιροειδείς συντεταγμένες γράφεται ως (παρ. Α', σελ. 87) :

$$\nabla^2 = \frac{4}{R^2(\xi^2 - \eta^2)} \left\{ \frac{\partial}{\partial \xi} \left[(\xi^2 - 1) \frac{\partial}{\partial \xi} \right] + \frac{\partial}{\partial \eta} \left[(1 - \eta^2) \frac{\partial}{\partial \eta} \right] + \frac{(\xi^2 - \eta^2)}{(\xi^2 - 1)(1 - \eta^2)} \frac{\partial^2}{\partial \phi^2} \right\} \quad (3.17)$$

Αντικαθιστώντας τις (3.17) και (3.16) στην (3.14) παίρνουμε την εξ.

$$(A'.17) \text{ για } \alpha = 2\sqrt{\frac{2}{-E}}, \beta = 0, \text{ και } c = R\sqrt{\frac{-E}{2}} \text{ (παρ. Α', σελ. 89)}$$

$$\begin{aligned} & \frac{\partial}{\partial \eta} \left((1 - \eta^2) \frac{\partial \psi(\mathbf{r})}{\partial \eta} \right) + \frac{\partial}{\partial \xi} \left((\xi^2 - 1) \frac{\partial \psi(\mathbf{r})}{\partial \xi} \right) \\ & + \frac{\xi^2 - \eta^2}{(\xi^2 - 1)(1 - \eta^2)} \frac{\partial^2 \psi(\mathbf{r})}{\partial \phi^2} + \left(\alpha c \xi - c^2(\xi^2 - \eta^2) \right) \psi(\mathbf{r}) = 0 \end{aligned} \quad (3.18)$$

ενώ αντικαθιστώντας την κυματοσυνάρτηση Ψ (παρ. Α'.2.1, σελ. 89) ως $\Psi(\xi, \eta, \phi) = \Xi(\xi)H(\eta)\Phi(\phi)$ στην (3.18) τελικά έχουμε :

$$\frac{d}{d\xi} \left((\xi^2 - 1) \frac{d\Xi(\xi)}{d\xi} \right) - \left(\lambda - ac\xi + c^2\xi^2 + \frac{\mu^2}{\xi^2 - 1} \right) \Xi(\xi) = 0 \quad (3.19\alpha')$$

$$\frac{d}{d\eta} \left((1 - \eta^2) \frac{dH(\eta)}{d\eta} \right) + \left(\lambda + c^2\eta^2 - \frac{\mu^2}{1 - \eta^2} \right) H(\eta) = 0 \quad (3.19\beta')$$

$$\frac{d^2\Phi(\phi)}{d\phi^2} + \mu^2\Phi(\phi) = 0 \quad (3.19\gamma')$$

εκ των οποίων οι (3.19α') και (3.19β') ονομάζονται ακτινική και γωνιακή εξίσωση αντίστοιχα. Έτσι, λοιπόν, επιτυγχάνουμε των διαχωρισμό των μεταβλητών και την αναλυτική επίλυση των διαφορικών εξισώσεων που προκύπτουν, ώστε να πάρουμε τις ιδιοτιμές³ (E) και τις ιδιοσυναρτήσεις ($\Psi(\mathbf{r})$) του τελεστή \hat{H} .

Δεδομένου ότι θα ασχοληθούμε κατά κόρον με τα μονοηλεκτρονιακά διατομικά μόρια (*OEDM's*)⁴, θα αναφερθούμε εδώ αναλυτικότερα στις λύσεις των εξισώσεων (3.19α') και (3.19β') :

α) Γωνιακές σφαιροειδείς συναρτήσεις

Η γωνιακή εξίσωση (3.19β') γράφεται ως εξής :

$$\frac{d}{d\eta} \left((1 - \eta^2) \frac{dH_{N_\eta}^m(\eta)}{d\eta} \right) + \left(\mathcal{A}_{N_\eta}^m(c) + c^2\eta^2 - \frac{m^2}{1 - \eta^2} \right) H_{N_\eta}^m(\eta) = 0 \quad (3.20)$$

³ Οι διαφορικές εξισώσεις (3.19α') και (3.19β') έχουν πόλους στα σημεία $\xi = 1, \eta = \pm 1$ (κανονικά ανώμαλα σημεία). Όμως η φυσική απαίτηση ώστε η κυματοσυνάρτηση να είναι τετραγωνικά ολοκληρώσιμη (L^2) σε όλο το χώρο, υπονοεί πως οι εξισώσεις αυτές δεν πρέπει να αποκλίνουν στα σημεία αυτά. Επιπρόσθετα, το λ πρέπει να έχει την ίδια τιμή και στις δύο εξισώσεις. Αποτέλεσμα αυτών είναι οι τιμές της παραμέτρου λ να περιορίζονται σε συγκεκριμένες (διακριτές) τιμές, τις ιδιοτιμές του μορίου.

⁴ Θα χρησιμοποιήσουμε τις λύσεις των *OEDM's* ως βάση για την περιγραφή πολυηλεκτρονιακών διατομικών μορίων.

όπου η ιδιοτιμή $\mathcal{A}_{N_\eta}^m(c)$ είναι τέτοια ώστε η συνάρτηση $H_{N_\eta}^m(\eta)$ να έχει N_η κόμβους (βλ. Α'2.1, σελ. 90) στο διάστημα $-1 \leq \eta \leq 1$ και να είναι πεπερασμένη στα $\eta = \pm 1$.

Αναπτύσσουμε την $H_{N_\eta}^m(\eta)$ σε γραμμικό συνδυασμό **συναφών πολυωνύμων Legendre** (βλ. Β'3, σελ. 104), εφόσον αυτά έχουν την απαιτούμενη συμπεριφορά στα ιδίμορφα σημεία $\eta = \pm 1$. Έτσι, από την (Β'35) έχουμε⁵ :

$$H_{N_\eta}^m(c; \eta) = \sum_{j=0}^{\infty} ' a_j(c) \mathcal{P}_{m+j}^m(\eta) \quad (3.21)$$

Θέτουμε την (3.21) στην (3.20) :

$$\sum_{j=0}^{\infty} ' a_j(c) \left\{ \frac{d}{d\eta} \left[(1 - \eta^2) \frac{d\mathcal{P}_{m+j}^m(\eta)}{d\eta} \right] + \left(\mathcal{A}_{N_\eta}^m(c) + c^2 \eta^2 - \frac{\mu^2}{1 - \eta^2} \right) \mathcal{P}_{m+j}^m(\eta) \right\} = 0 \quad (3.22)$$

Κατόπιν, προσθέτουμε και στις δύο πλευρές της (3.22) το $(m+j)(m+j+1)$, ώστε να φέρουμε το αριστερό μέρος στη μορφή μίας άπειρης σειράς εξισώσεων Legendre (βλ. (Β'31) με $\kappa \rightarrow m+j$, σελ. 104), οι οποίες θα μηδενίζονται :

$$\begin{aligned} & \sum_{j=0}^{\infty} ' a_j(c) \left\{ \frac{d}{d\eta} \left[(1 - \eta^2) \frac{d\mathcal{P}_{m+j}^m(\eta)}{d\eta} \right] + \left((m+j)(m+j+1) - \frac{\mu^2}{1 - \eta^2} \right) \mathcal{P}_{m+j}^m(\eta) \right\} \\ & = \sum_{j=0}^{\infty} ' a_j(c) [-c^2 \eta^2 - \mathcal{A}_{N_\eta}^m(c) + (m+j)(m+j+1)] \mathcal{P}_{m+j}^m(\eta) \end{aligned} \quad (3.23)$$

Συγκρίνοντας την (3.23) με την (Β'31) βλέπουμε πως το αριστερό μέρος μηδενίζεται, οπότε :

$$\sum_{j=0}^{\infty} ' a_j(c) [-c^2 \eta^2 - \mathcal{A}_{N_\eta}^m(c) + (m+j)(m+j+1)] \mathcal{P}_{m+j}^m(\eta) = 0 \quad (3.24)$$

⁵ Ο τόνος στο άθροισμα υποδηλώνει πως η άθροιση γίνεται μόνο πάνω σε άρτιες τιμές του j , δηλ. η λύση έχει βήμα 2.

Απο τη θεωρία ξέρουμε πως η γεννήτρια συνάρτηση για τα πολυώνυμα Legendre είναι :

$$g(x, t) = \frac{1}{\sqrt{1 - 2xt + t^2}} = \sum_{n=0}^{\infty} \mathcal{P}_n(x)t^n \quad (3.25)$$

Διαφορίζοντας την (3.25) ως προς x , m φορές παίρνουμε την γεννήτρια συνάρτηση για τα συναφή πολυώνυμα Legendre :

$$g_m(x, t) \equiv \frac{\partial^m g(x, t)}{\partial x^m} = \frac{(-1)^m (2m - 1)!!}{(1 - 2xt + t^2)^{m + \frac{1}{2}}} = \sum_{s=0}^{\infty} \mathcal{P}_{s+m}^m(x)t^s \quad (3.26)$$

Εάν τώρα διαφορίσουμε ως προς t την (3.26)⁶ παίρνουμε τελικά τον αναδρομικό τύπο :

$$(s + 1)\mathcal{P}_{s+m+1}^m(x) - (2m + 1 + 2s)x\mathcal{P}_{s+m}^m(x) + (s + 2m)\mathcal{P}_{s+m-1}^m(x) = 0 \quad (3.27)$$

και επαναλαμβάνοντας⁷

$$\begin{aligned} x^2\mathcal{P}_{s+m}^m(x) &= \frac{(s + 2m - 1)(s + 2m)}{[2(s + m) - 1][2(s + m) + 1]}\mathcal{P}_{s+m-2}^m(x) \\ &+ \frac{2(s + m)(s + m + 1) - 2m^2 - 1}{[2(s + m) - 1][2(s + m) + 3]}\mathcal{P}_{s+m-2}^m(x) \\ &+ \frac{(s + 2)(s + 1)}{[2(s + m) + 1][2(s + m) + 3]}\mathcal{P}_{s+m+2}^m(x) \end{aligned} \quad (3.28)$$

Εφαρμόζοντας την (3.28) στην (3.24) και μετά από κάποιες πράξεις έχουμε :

$$\sum_{j=0}^{\infty} \left[\alpha_j^m(c)a_{j+2}(c) + (\beta_j^m(c) - \mathcal{A}_{N_j}^m)a_j(c) + \gamma_j^m(c)a_{j-2}(c) \right] \mathcal{P}_{m+j}^m(\eta) = 0, \quad (3.29)$$

όπου

$$\rightarrow \begin{cases} \alpha_j^m(c) = c^2 \frac{(2m + j + 1)(2m + 2j + 2)}{(2(m + j) + 3)(2(m + j) + 5)}, \\ \beta_j^m(c) = (m + j)(m + j + 1) - c^2 \frac{2(m + j)(m + j + 1) - 2m^2 - 1}{((2(m + j) - 1))(2(m + j) + 3)}, \\ \gamma_j^m(c) = c^2 \frac{j(1 - j)}{(2(m + j) - 3)(2(m + j) - 1)} \end{cases} \quad (3.30)$$

⁶ Το διπλό παραγοντικό υποδηλώνει εδώ ότι το παραγοντικό μειώνεται κατά 2 κάθε φορά : $(n)!! = n \cdot (n - 2) \cdot \dots [9]$.

⁷ Γενικά ισχύει: $x^k \mathcal{P}_l^m(x) = \frac{(l - m + 1)x^{k-1} \mathcal{P}_{l+1}^m(x) + (l + m) \mathcal{P}_{l-1}^m(x)}{2l + 1}$.

Στην (3.29) οι συντελεστές των $\mathcal{P}_{m+j}^m(\eta)$ ⁸ είναι ανεξάρτητοι του η και εφόσον τα $\mathcal{P}_{m+j}^m(\eta)$ είναι γραμμικά ανεξάρτητα⁹, τότε θα πρέπει να είναι ίσοι με το μηδέν. Άρα :

$$\alpha_j^m(c)a_{j+2}(c) + (\beta_j^m(c) - \mathcal{A}_{N_\eta}^m)a_j(c) + \gamma_j^m(c)a_{j-2}(c) = 0 \quad (3.31)$$

Ακολουθώντας τον Power [7], [3] οι παραπάνω ταυτόχρονες εξισώσεις λύνονται με την μέθοδο των συνεχόμενων κλασμάτων (παρ. Β.1.2, σελ. 99). Έτσι, λαμβάνουμε από την (3.31) ότι οι λόγοι των διαδοχικών όρων είναι :

$$\frac{a_j(c)}{a_{j+2}(c)} = \frac{-\alpha_j^m(c)}{\beta_j^m(c) - \mathcal{A}_{N_\eta}^m + \gamma_j^m(c)\frac{a_{j-2}(c)}{a_j(c)}}, \quad (3.32\alpha')$$

$$\frac{a_j(c)}{a_{j-2}(c)} = \frac{-\gamma_j^m(c)}{\beta_j^m(c) - \mathcal{A}_{N_\eta}^m + \alpha_j^m(c)\frac{a_{j+2}(c)}{a_j(c)}}. \quad (3.32\beta')$$

Ασυμπτωτική συμπεριφορά : Υπολογίζουμε την μορφή που παίρνουν οι συντελεστές στις (3.32α') και (3.32β'), καθώς $j \rightarrow \pm\infty$, αναπτύσσοντας σε σειρά *Maclaurin* το λόγο $\frac{a_{j+2}(c)}{a_j(c)}$, ως προς τη μεταβλητή $i = \frac{1}{j}$:

$$\frac{a_{j+2}(c)}{a_j(c)} = \delta_0 + \delta_1 i + \delta_2 i^2 + \delta_3 i^3 + \mathcal{O}(i^4) \quad (3.33)$$

Εν συνεχεία αντικαθιστούμε στην (3.32β') και τελικά για $j \rightarrow +\infty$ έχουμε :

$$\frac{a_j(c)}{a_{j-2}(c)} = \frac{c^2}{4j^2} - \frac{2mc^2}{2j^3} + \mathcal{O}\left(\frac{1}{j}\right)^4 \quad (3.34)$$

⁸ Η απαίτηση να ισχύει $l = m + j \geq |m|$ στα συναφή πολυώνυμα Legendre, γίνεται προφανής από το ότι το άθροισμα ξεκινάει από $j = 0$.

⁹ Εννοείται για τιμές του δείκτη j που διαφέρουν κατά 2, όπως φαίνεται και από την (3.29).

Ομοίως για $j \rightarrow -\infty$ έχουμε :

$$\frac{a_j(c)}{a_{j+2}(c)} = -\frac{c^2}{4j^2} + \frac{c^2}{2j^3} + \mathcal{O}\left(\frac{1}{j}\right)^4 \quad (3.35)$$

Παρατηρούμε πως, και στις δύο περιπτώσεις, δεν υπάρχει εξάρτηση από το m στο κυρίαρχο όρο $\frac{c^2}{4j^2}$. Λύσεις της παραπάνω μορφής καλούνται “ελάχιστες” (βλ. Β.1.2, σελ. 99), και είναι αυτές που θα δώσουν συγκλίνουσες σειρές στην (3.21). Βέβαια, για να συμβεί αυτό, πρέπει και η παράμετρος $\mathcal{A}_{N_\eta}^m$ να παίρνει συγκεκριμένες τιμές¹⁰, όπως φαίνεται και από τις (3.32α') και (3.32β').

Κατ' αυτόν τον τρόπο υπολογίζονται οι συντελεστές ανάπτυξης a_j της εξίσωσης (3.21), και το μόνο που απομένει είναι η κανονικοποίησή της.

Κανονικοποίηση

Για τις γωνιακές σφαιροειδείς συναρτήσεις (βλ. Β.3, σελ. 104), η κανονικοποίηση των Meixner και Schäfke[18] είναι:

$$\int_{-1}^1 \mathcal{S}_n^{m(1)}(c; t) \mathcal{S}_k^{m(1)}(c; t) dt = \frac{2}{2n+1} \frac{(n+m)!}{(n-m)!} \delta_{nk} \xrightarrow{n=k}$$

$$\int_{-1}^1 |\mathcal{S}_n^{m(1)}(c; t)|^2 dt = \frac{2}{2n+1} \frac{(n+m)!}{(n-m)!}, \quad n, m, k \in \mathbb{Z} \quad (3.36)$$

¹⁰ Οι τιμές αυτές ονομάζονται *ιδιοτιμές σφαιροειδούς* για τις οποίες το σύνολο $\{a_j(c)\}$ είναι *minimal* καθώς $k \rightarrow \pm\infty$. Ουσιαστικά, η δήλωση αυτή, είναι ισοδύναμη με τα προβλήματα συνοριακών συνθηκών των δευτεροτάξιων διαφορικών εξισώσεων, όπου οι συνοριακές συνθήκες ικανοποιούνται μόνο για ορισμένες τιμές μιας παραμέτρου, τις ιδιοτιμές. Στη προκειμένη περίπτωση οι “συνοριακές” συνθήκες είναι οι συντελεστές της (3.21) να είναι ελάχιστοι καθώς $k \rightarrow \pm\infty$.

η οποία φυσικά προέρχεται από τη σχέση ορθοκανονικότητας [10] των συναφών πολυωνύμων Legendre,

$$\int_{-1}^1 \mathcal{P}_n^m(t) \mathcal{P}_k^m(t) dt = \frac{2}{2n+1} \frac{(n+m)!}{(n-m)!} \delta_{nk} \xrightarrow{n=k}$$

$$\int_{-1}^1 |\mathcal{P}_n^m(t)|^2 dt = \frac{2}{2n+1} \frac{(n+m)!}{(n-m)!}, \quad n, m, k \in \mathbb{Z} \quad (3.37)$$

Επομένως, στη προκειμένη περίπτωση έχουμε :

$$\int_{-1}^1 |H_{N_\eta}^m(c; t)|^2 dt = 1 \Rightarrow |\mathcal{B}_{N_\eta}^m|^2 \int_{-1}^1 |\mathcal{S}_{N_\eta}^{m(1)}(c; t)|^2 dt = 1 \xrightarrow{(3.36)}$$

$$|\mathcal{B}_{N_\eta}^m|^2 \frac{2}{2N_\eta+1} \frac{(N_\eta+m)!}{(N_\eta-m)!} = 1 \Rightarrow \mathcal{B}_{N_\eta}^m = \sqrt{\frac{2N_\eta+1}{2} \frac{(N_\eta-m)!}{(N_\eta+m)!}} \quad (3.38)$$

Οπότε η (3.21) γίνεται :

$$H_{N_\eta}^m(c; \eta) = \sqrt{\frac{2N_\eta+1}{2} \frac{(N_\eta-m)!}{(N_\eta+m)!}} \sum_{j=0}^{\infty} ' a_j(c) \mathcal{P}_{m+j}^m(\eta) \quad (3.39)$$

Εν συνεχεία, περιγράφουμε τη λύση της ακτινικής εξίσωσης, η οποία θα μας δώσει τις αντίστοιχες ακτινικές συναρτήσεις.

β) Ακτινικές γενικευμένες σφαιροειδείς συναρτήσεις

Η ακτινική εξίσωση (3.19α') γράφεται ως εξής :

$$\frac{d}{d\xi} \left((\xi^2 - 1) \frac{d\Xi_{N_\xi}^m(\xi)}{d\xi} \right) - \left(\mathfrak{R}_{N_\xi}^m - \alpha c \xi + c^2 \xi^2 + \frac{m^2}{\xi^2 - 1} \right) \Xi_{N_\xi}^m(\xi) = 0 \quad (3.40\alpha')$$

όπου η ιδιοτιμή $\mathfrak{R}_{N_\xi}^m(c)$ είναι τέτοια ώστε η συνάρτηση $\Xi_{N_\xi}^m(\xi)$ να έχει N_ξ κόμβους (βλ. Α'.2.1, σελ. 90) στο διάστημα $1 \leq \xi < \infty$, να είναι πεπερασμένη στο $\xi = 1$ και να μηδενίζεται στο $\xi \rightarrow \infty$.

Η καλύτερη ανάπτυξη για τις ακτινικές γενικευμένες σφαιροειδείς συναρτήσεις είναι η δυναμοσειρά του Jaffé (B'.29) (βλ. B'.1.3, σελ. 101) :

$$\Xi(\xi) = (\xi^2 - 1)^{\frac{m}{2}} e^{-c\xi} (\xi + 1)^{\frac{m}{2} - (m+1)} y(x) \quad (3.41)$$

όπου είναι βολικός ο μετασχηματισμός $x = \frac{\xi - 1}{\xi + 1}$, εφόσον “μετατοπίζει” το διάστημα ορισμού από το ημι-άπειρο διάστημα $1 \leq \xi < \infty$ στο πεπερασμένο $0 \leq x < 1$. Θέτοντας την (3.41) στην (3.40α) παίρνουμε :

$$x(1-x)^2 \frac{d^2 y(x)}{dx^2} + [(m-\alpha+3)x^2 + (\alpha-2m-4c-4)x + m+1] \frac{dy(x)}{dx} + \left[\left(\frac{\alpha}{2} - m - 1\right) \left(\frac{\alpha}{2} - 1\right) x + (2c+1) \left(\frac{\alpha}{2} - m - 1\right) + \frac{m\alpha}{2} - c^2 - \mathfrak{R}_{N_\xi}^m(c) \right] y(x) = 0 \quad (3.42)$$

και αντικαθιστώντας την δυναμοσειρά γύρω απ’το $x_0 = 0$,

$$y(x) = \sum_{i=0}^{\infty} g_i x^i \quad (3.43)$$

καταλήγουμε σε μία αναδρομική σχέση τριών όρων για τα g_i :

$$\alpha_i g_{i+1} + (\beta_i - \mathfrak{R}_{N_\xi}^m) + \gamma_i g_{i-1} = 0 \quad (3.44)$$

όπου

$$\Rightarrow \begin{cases} \alpha_i = (i+1)(m+i+1), \\ \beta_i = -c^2 - 2i(i+m) + (2c+1)\left(\frac{\alpha}{2} - 1\right) + (2i+m)\left(\frac{\alpha}{2} - 2c - 1\right), \\ \gamma_i = \left(\frac{\alpha}{2} - i\right)\left(\frac{\alpha}{2} - m - i\right) \end{cases} \quad (3.45)$$

Οι ιδιοτιμές $\mathfrak{R}_{N_\xi}^m$ καθορίζονται, όπως και προηγουμένως, από τη συνθήκη το σύνολο $\{g_i\}$ να είναι “ελάχιστο” καθώς $i \rightarrow \infty$.

Παρατηρούμε από την (3.45), ότι στους συντελεστές των g_i στην (3.44) ,το α δεν πολλαπλασιάζεται με το c . Έτσι, καθώς $E \rightarrow 0$ (βλ. (Α.18)) οι συντελεστές αυτοί γίνονται απεριόριστα μεγάλοι.

Η τελική λύση, λοιπόν, της εξ. (3.14) θα είναι της μορφής :

$$\Psi(\xi, \eta, \phi) = (\xi^2 - 1)^{\frac{m}{2}} e^{-c\xi} (\xi + 1)^{\frac{\alpha}{2} - (m+1)} \sum_{i=0}^{\infty} g_i \left(\frac{\xi - 1}{\xi + 1} \right)^i \times \sum_{j=0}^{\infty} f_j(c) \mathcal{P}_{m+j}^m(\eta) \frac{1}{\sqrt{2\pi}} e^{\pm im\phi} \quad (3.46)$$

όπου βέβαια πρέπει να γίνουν και οι απαραίτητες κανονικοποιήσεις.

Αυτή είναι η γενική μορφή των λύσεων που θα χρησιμοποιήσουμε για τον υπολογισμό πολυπλοκότερων διατομικών μορίων, όπως του H_2 το οποίο θα είναι και το μοντέλο του επόμενου κεφαλαίου (βλ. κεφ. 4).

Κεφάλαιο 4

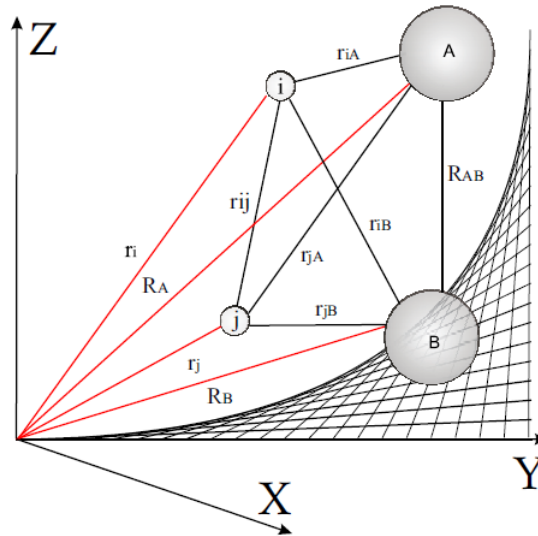
Πολυηλεκτρονιακά Διατομικά Μόρια

4.1 Εισαγωγή

Στο παρόν κεφάλαιο θα εισάγουμε την έννοια της ορίζουσας *Slater*, καθώς και την αντίστοιχη ανάπτυξη που θα χρησιμοποιήσουμε στον υπολογισμό, μέσω της περιγραφής ενός απλού διατομικού μορίου. Επιπλέον, μέσα στο πλαίσιο του ίδιου παραδείγματος, θα περιγράψουμε συνοπτικά την μέθοδο *CI*, που είναι αναγκαία ώστε να έχουμε σωστά αποτελέσματα στο όριο $R \rightarrow \infty$ και γενικώς κατά το δυνατόν ακριβή περιγραφή της ηλεκτρονιακής συσχέτισης. Τέλος, θα αναπτύξουμε την μέθοδο των *απροσδιορίστων συντελεστών Lagrange* ως προαπαιτούμενο της μεθόδου *CI*.

4.2 Το Μόριο του Υδρογόνου H_2

Λαμβάνοντας υπ' όψιν τη προσέγγιση Born-Oppenheimer (κεφ. 2.3, σελ. 9), η ηλεκτρονιακή Χαμιλτονιανή για το μόριο του H_2 (σχ. 4.2, σελ.



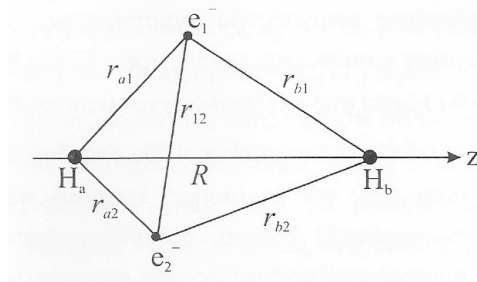
Σχήμα 4.1: Διαγραμματική απεικόνιση ενός πολυηλεκτρονιακού διατομικού μορίου.

35) θα είναι (εκπεφρασμένη σε ατομικές μονάδες (a.u)) :

$$\hat{H}_{el} = -\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{|\mathbf{r}_1 - \mathbf{R}_a|} - \frac{1}{|\mathbf{r}_2 - \mathbf{R}_a|} - \frac{1}{|\mathbf{r}_1 - \mathbf{R}_b|} - \frac{1}{|\mathbf{r}_2 - \mathbf{R}_b|} + \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} + \frac{1}{|\mathbf{R}_a - \mathbf{R}_b|} \quad (4.1)$$

όπου έχουμε κατά σειρά :

- την κινητική ενέργεια του ηλεκτρονίου 1,
- την κινητική ενέργεια του ηλεκτρονίου 2,
- τη δυναμική ενέργεια λόγω έλξης του ηλεκτρονίου 1 από τον πυρήνα a,
- τη δυναμική ενέργεια λόγω έλξης του ηλεκτρονίου 2 από τον πυρήνα a,
- τη δυναμική ενέργεια λόγω έλξης του ηλεκτρονίου 1 από τον πυρήνα b,
- τη δυναμική ενέργεια λόγω έλξης του ηλεκτρονίου 2 από τον πυρήνα b,

Σχήμα 4.2: Διαγραμματική απεικόνιση του H_2 .

- τη δυναμική ενέργεια λόγω απώσεως των ηλεκτρονίων,
- τη δυναμική ενέργεια λόγω απώσεως των πυρήνων $\frac{1}{R}$, η οποία είναι σταθερά.

Η (4.1) μπορεί να γραφεί και ως :

$$\hat{H}_{el} = \hat{h}_1 + \hat{h}_2 + \frac{1}{r_{12}} + \frac{1}{R}, \quad (4.2)$$

$$\text{όπου } \hat{h}_i = -\frac{1}{2}\nabla_i^2 - \frac{1}{r_{ai}} - \frac{1}{r_{bi}} \text{ μονοηλεκτρονιακοί τελεστές} \quad (4.2\alpha')$$

$$\text{με } \frac{1}{r_{12}} \equiv \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}, \frac{1}{r_{ji}} \equiv \frac{1}{|\mathbf{r}_i - \mathbf{R}_j|} \text{ και } \frac{1}{R} \equiv \frac{1}{|\mathbf{R}_a - \mathbf{R}_b|} \quad (4.2\beta')$$

Η ύπαρξη του όρου $\frac{1}{r_{12}}$ στην (4.2), ο οποίος εμπλέκει τις συντεταγμένες και των δύο ηλεκτρονίων, δεν επιτρέπει τον διαχωρισμό των μεταβλητών και κατ' επέκταση την αναλυτική λύση της αντίστοιχης εξίσωσης Schrödinger.

Εισάγουμε έτσι τα *γινόμενα Hartree* ως βάση για την προσεγγιστική επίλυση του παραπάνω προβλήματος.

4.2.1 Γινόμενα Hartree

Αναγκαστικά, λοιπόν, η (4.1) λύνεται προσεγγιστικά μέσω της θεωρίας μεταβολών. Κατά τα γνωστά, θα πρέπει να επιλέξουμε μια

δοκιμαστική κυματοσυνάρτηση, και η πιο προφανής επιλογή θα ήταν ένα γινόμενο μοριακών τροχιακών-spin $\{\psi_i(\mathbf{x}_j)|i, j = 1, 2, \dots, N\}$ του \mathbf{H}_2^+ :

$$\Psi^H(1, 2) = \psi_1(1)\psi_2(2), \quad \text{ή γενικότερα} \quad (4.3)$$

$$\Psi^H(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \psi_i(\mathbf{x}_1)\psi_j(\mathbf{x}_2) \cdots \psi_k(\mathbf{x}_N) \quad (4.4)$$

όπου $\psi_1, \psi_2 = 1\sigma_g, 1\bar{\sigma}_g, 1\sigma_u, 1\bar{\sigma}_u$ και τα 1, 2 ή $\mathbf{x}_i = \{\mathbf{r}_i, \omega\}$ συμβολίζουν τις χωρικές και spin συντεταγμένες των δύο ηλεκτρονίων. Τα γινόμενα αυτά ονομάζονται γινόμενα *Hartree* και χρησιμοποιούνται στην ομώνυμη προσέγγιση *Hartree-Fock*. Είναι εμφανές πως τα γινόμενα Hartree είναι κυματοσυναρτήσεις όπου οι συντεταγμένες των ηλεκτρονίων είναι ανεξάρτητες¹ μεταξύ τους, δηλ. δεν υπάρχει συσχετισμός των ηλεκτρονίων στη κυματοσυνάρτηση. Ακόμα και αν ήταν ανεξάρτητα τα ηλεκτρόνια, υπάρχει ένα βασικό μειονέκτημα στα γινόμενα Hartree: δεν υπακούουν στην παρακάτω αρχή :

Θεώρημα 4.2.1. *Η Αρχή Αντισυμμετρίας του Pauli δεν διακρίνει μεταξύ ταυτοτικών φερμιονίων και απαιτεί η ηλεκτρονιακή κυματοσυνάρτηση να είναι αντισυμμετρική² ως προς την εναλλαγή των χωρικών και spin συντεταγμένων οποιωνδήποτε δύο σωματιδίων[4].*

Επομένως, για να ικανοποιήσουμε την παραπάνω αρχή εισάγουμε σε αυτό το σημείο τις ορίζουσες Slater.

¹ Το οποίο ουσιαστικά σημαίνει πως η πιθανότητα να βρεθεί το ηλεκτρόνιο 1 σε ένα δεδομένο σημείο στο χώρο είναι ανεξάρτητη από τη θέση του ηλεκτρονίου 2. Στη πραγματικότητα, βέβαια, τα δύο ηλεκτρόνια απωθούνται, εξ αιτίας της αλληλεπίδρασης Coulomb, οπότε το ηλεκτρόνιο 1 δεν θα βρεθεί ποτέ στην ίδια περιοχή του χωροχρόνου ταυτόχρονα με το ηλεκτρόνιο 2. Δηλαδή, η κίνηση των δύο ηλεκτρονίων είναι πλήρως συσχετισμένη[5].

² Δηλαδή να αλλάζει πρόσημο: κατ' επέκταση αυτό σημαίνει πως όταν δύο σωματίδια έχουν τις ίδιες συντεταγμένες η κυματοσυνάρτηση θα πρέπει να μηδενίζεται[6].

4.2.2 Ορίζουσες Slater

Τα γινόμενα Hartree, λοιπόν, δεν ικανοποιούν την αρχή αντισυμμετρίας, μπορούμε όμως να κατάκευάσουμε αντισυμμετρικοποιημένες κυματοσυναρτήσεις από αυτά :

$$\Psi(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{\sqrt{2}} \begin{vmatrix} \psi_1(\mathbf{x}_1) & \psi_2(\mathbf{x}_1) \\ \psi_1(\mathbf{x}_2) & \psi_2(\mathbf{x}_2) \end{vmatrix} = \frac{1}{\sqrt{2}} [\psi_1(\mathbf{x}_1)\psi_2(\mathbf{x}_2) - \psi_2(\mathbf{x}_1)\psi_1(\mathbf{x}_2)] \quad (4.5)$$

όπου ο παράγοντας $\frac{1}{\sqrt{2}}$ οφείλεται στην κανονικοποίηση. Προφανώς, όταν εναλλάξουμε τις συντεταγμένες των 2 ηλεκτρονίων η $\Psi(\mathbf{x}_1, \mathbf{x}_2)$ αλλάζει πρόσημο : $\Psi(\mathbf{x}_1, \mathbf{x}_2) = -\Psi(\mathbf{x}_2, \mathbf{x}_1)$, ενώ αν και τα δύο ηλεκτρόνια καταλαμβάνουν το ίδιο τροχιακό, η κυματοσυνάρτηση μηδενίζεται. Η παραπάνω ορίζουσα ονομάζεται *ορίζουσα Slater* και η γενίκευσή της για ένα σύστημα N ηλεκτρονίων και μοριακών τροχιακών (spin-orbitals) $\{\psi_i(\mathbf{x}_j) | i, j, k = 1, 2, \dots, N\}$ είναι :

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \psi_i(\mathbf{x}_1) & \psi_j(\mathbf{x}_1) & \cdots & \psi_k(\mathbf{x}_1) \\ \psi_i(\mathbf{x}_2) & \psi_j(\mathbf{x}_2) & \cdots & \psi_k(\mathbf{x}_2) \\ \vdots & \vdots & & \vdots \\ \psi_i(\mathbf{x}_N) & \psi_j(\mathbf{x}_N) & \cdots & \psi_k(\mathbf{x}_N) \end{vmatrix} \quad (4.6)$$

όπως και προηγουμένως, η εναλλαγή των συντεταγμένων δύο ηλεκτρονίων αντιστοιχεί σε εναλλαγή δύο σειρών της ορίζουσας, που φυσικά αλλάζει το πρόσημό της. Η ορίζουσα Slater συμπεριλαμβάνει τη *συσχέτιση ανταλλαγής*³, δηλ. τη συσχέτιση της κίνησης των ηλεκτρονίων με παράλληλα spin.

Στο σημείο αυτό θα αναλύσουμε τις ορίζουσες σε γινόμενα των στοιχείων τους, εφόσον θα μας χρειαστούν αργότερα για τον υπολογισμό των διαφορών μητροστοιχείων που θα προκύψουν. Η ορίζουσα

³ Τα φαινόμενα ανταλλαγής (exchange) προκύπτουν από την απαίτηση η $|\Psi|^2$ να είναι αναλλοίωτη ως προς την ανταλλαγή των χωρικών και spin συντεταγμένων οποιωνδήποτε δύο ηλεκτρονίων.

(4.6) μπορεί να γραφεί και ως :

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \sum_i^N \sum_j^N \cdots \sum_k^N \varepsilon_{ij\dots k} \psi_i(\mathbf{x}_1) \psi_j(\mathbf{x}_2) \cdots \psi_k(\mathbf{x}_N) \quad (4.7)$$

όπου⁴

$$\varepsilon_{ij\dots k} = \begin{cases} +1, & \text{εάν } ij\dots k: \text{άρτια μετάθεση,} \\ -1, & \text{εάν } ij\dots k: \text{περιττή μετάθεση,} \\ 0, & \text{εάν } ij\dots k: \text{δεν είναι μετάθεση.} \end{cases} \quad (4.8)$$

ο πλήρως αντισυμμετρικός τανυστής N -οστής τάξης (ή σύμβολα Levi-Civita [20]). Η (4.7) μπορεί να αναπτυχθεί και ως :

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \sum_{n=1}^{N!} (-1)^{p_n} \hat{\mathfrak{P}}_n [\psi_i(\mathbf{x}_1) \psi_j(\mathbf{x}_2) \cdots \psi_k(\mathbf{x}_N)] \quad (4.9)$$

όπου $\hat{\mathfrak{P}}_n$ είναι ένας τελεστής που παράγει τη n -οστή εναλλαγή των ηλεκτρονίων $1, 2, \dots, N$ και p_n είναι ο αριθμός των μεταθέσεων που απαιτούνται για να καταλήξουμε σε αυτήν[4].

Επανερχόμενοι τώρα στο μόριο του H_2 , χρησιμοποιώντας τα μοριακά τροχιακά $1\sigma_g, 1\bar{\sigma}_g, 1\sigma_u, 1\bar{\sigma}_u$, σχηματίζουμε $\frac{4!}{(4-2)!2!} = 6$ διαφορετικές οριζουσες Slater : $|1\sigma_g 1\bar{\sigma}_g|, |1\sigma_g 1\sigma_u|, |1\sigma_g 1\bar{\sigma}_u|, |1\bar{\sigma}_g 1\sigma_u|, |1\bar{\sigma}_g 1\bar{\sigma}_u|, |1\sigma_u 1\bar{\sigma}_u|$ ⁵ Λαμβάνοντας γραμμικούς συνδυασμούς των παραπάνω οριζουσών Slater, ώστε να σχηματίσουμε κυματοσυναρτήσεις που είναι ιδιοσυναρτήσεις⁶ και του τελεστή \hat{S}^2 , τελικά έχουμε :

⁴Η (4.7) είναι ισοδύναμη με την Λαπλασιανή ανάπτυξη της οριζουσας σε υποοριζουσες :

$$D_N = \sum_{j=1}^N a_{ij} (-1)^{i+j} M_{ij}$$

όπου M_{ij} : είναι μία οριζουσα τάξης $N-1$ που παράγεται από την αρχική απαλείφοντας την i σειρά και την j στήλη[9].

⁵ Χρησιμοποιώ εδώ χάριν συντομίας τον κλασικό συμβολισμό της κανονικοποιημένης οριζουσας Slater : $\Psi(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) \equiv |\psi_i \psi_j \cdots \psi_k|$ [4].

⁶ Εφόσον ο τελεστής \hat{S}^2 εξαρτάται μόνο από τις spin συντεταγμενες ενώ η Χαμιλτονιανή \hat{H}_{el} μόνο από τις χωρικές, μετατίθενται μεταξύ τους $[\hat{S}^2, \hat{H}_{el}] = 0$ και κατά

Δοκιμαστικές Κυματοσυναρτήσεις για το μόριο του H₂:

Κυματοσυνάρτηση	<i>S</i>	<i>m_s</i>	Φασματοσκοπικός όρος
$ 1\sigma_g 1\bar{\sigma}_g\rangle$	0	0	$1^1\Sigma_g^+$
$ 1\sigma_g 1\sigma_u\rangle$	1	1	$3\Sigma_u^+$
$\frac{1}{\sqrt{2}}(1\bar{\sigma}_g 1\sigma_u\rangle + 1\sigma_g 1\bar{\sigma}_u\rangle)$	1	0	
$ 1\bar{\sigma}_g 1\bar{\sigma}_u\rangle$	1	-1	
$\frac{1}{\sqrt{2}}(1\bar{\sigma}_g 1\sigma_u\rangle - 1\sigma_g 1\bar{\sigma}_u\rangle)$	0	0	$1^1\Sigma_u^+$
$ 1\sigma_u 1\bar{\sigma}_u\rangle$	0	0	$2^1\Sigma_g^+$

4.2.3 Μητροστοιχεία

Κατόπιν υπολογίζουμε τις ενέργειες για κάθε κατάσταση ως αναμενόμενες τιμές της Χαμιλτονιανής (4.2) : π.χ. για την κατάσταση $1^1\Sigma_g^+$ θα έχουμε $E = \langle 1\sigma_g 1\bar{\sigma}_g | \hat{H}_{el} | 1\sigma_g 1\bar{\sigma}_g \rangle$ από την οποία θα πρέπει να υπολογίσουμε ξεχωριστά, τα διάφορα μητροστοιχεία που παράγονται: μονοηλεκτρονιακά: $\langle 1\sigma_g 1\bar{\sigma}_g | \hat{h}_i | 1\sigma_g 1\bar{\sigma}_g \rangle$, διηλεκτρονιακά: $\langle 1\sigma_g 1\bar{\sigma}_g | \frac{1}{r_{12}} | 1\sigma_g 1\bar{\sigma}_g \rangle$, καθώς και τα επικαλύψεως : $\langle 1\sigma_g 1\bar{\sigma}_g | 1\sigma_g 1\bar{\sigma}_g \rangle$.

Τελικά έχουμε :

$$E [1^1\Sigma_g^+] = 2h_{gg} + J_{gg} + \frac{1}{R} \quad (4.10\alpha')$$

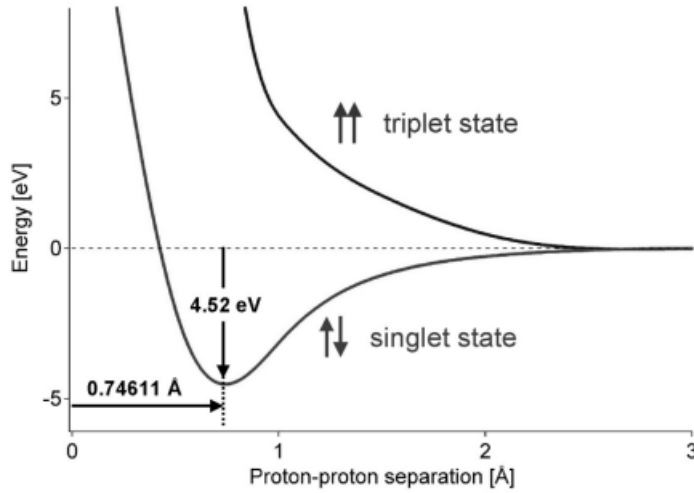
$$\text{με } h_{gg} = \langle 1\sigma_g | \hat{h} | 1\sigma_g \rangle \quad (4.10\beta')$$

$$\text{και } J_{gg} = \langle 1\sigma_g(1)1\sigma_g(2) | \frac{1}{r_{12}} | 1\sigma_g(1)1\sigma_g(2) \rangle \quad (4.10\gamma')$$

οπου η (4.10γ') ονομάζεται ολκλήρωμα Coulomb. Επαναλαμβάνουμε και για τις υπόλοιπες καταστάσεις, οπότε συνολικά έχουμε [6] :

Ενέργειες για το μόριο του H₂:

συνέπεια θα πρέπει να έχουν κοινές ιδιοσυναρτήσεις[5].



Σχήμα 4.3: Γραφική παράσταση της ενέργειας συναρτήσει της διαπυρηνικής αποστάσεως, των καταστάσεων $1^1\Sigma_g^+$ και $3^3\Sigma_u^+$, για το H_2 .

Κατάσταση	Ενέργεια
$1^1\Sigma_g^+$	$2h_{gg} + J_{gg} + \frac{1}{R}$
$3^3\Sigma_u^+$	$h_{gg} + h_{uu} + J_{gu} - K_{gu} + \frac{1}{R}$
$1^1\Sigma_u^+$	$h_{gg} + h_{uu} + J_{gu} + K_{gu} + \frac{1}{R}$
$2^1\Sigma_g^+$	$2h_{uu} + J_{uu} + \frac{1}{R}$

$$\text{με } J_{gu} = \langle 1\sigma_g(1)1\sigma_u(2) | \frac{1}{r_{12}} | 1\sigma_g(1)1\sigma_u(2) \rangle \quad (4.11\alpha')$$

$$K_{gu} = \langle 1\sigma_g(1)1\sigma_u(1) | \frac{1}{r_{12}} | 1\sigma_g(2)1\sigma_u(2) \rangle \quad (4.11\beta')$$

όπου K_{gu} : ολοκλήρωμα Ανταλλαγής (Exchange).

Τέλος, εξετάζουμε την ασυμπτωτική συμπεριφορά των παραπάνω καταστάσεων, ώστε να επιβεβαιώσουμε την ορθότητά τους.

4.2.4 Συμπεριφορά στο όριο $R \rightarrow \infty$

Είναι λογικό να αναμένουμε από τη διάσπαση του μορίου H_2 , δύο άτομα H στη θεμελειώδη κατάσταση: $H_2(1^1\Sigma_g^+) \rightarrow H(^2S; 1s^1) + H(^2S; 1s^1)$. Όμως, εάν πάρουμε την κυματοσυνάρτηση της θεμελειώδους κατάστασης του H_2 , και την αναπτύξουμε σε γραμμικούς συνδυασμούς ατομικών τροχιακών (LCAO) :

$$\begin{aligned} |1\sigma_g 1\bar{\sigma}_g| &= |(s_a(1) + s_b(1))(\bar{s}_a(2) + \bar{s}_b(2))| \\ &= |s_a(1)\bar{s}_a(2)| + |s_a(1)\bar{s}_b(2)| + |s_b(1)\bar{s}_a(2)| + |s_b(1)\bar{s}_b(2)| \end{aligned} \quad (4.12)$$

παρατηρούμε πως η κυματοσυνάρτηση έχει 50% ιοντικό και 50% ομοιοπολικό χαρακτήρα⁷. Ο χαρακτήρας αυτός ισχύει, πέραν της απόστασης ισορροπίας του μορίου, και για $R \rightarrow \infty$ για την ακρίβεια, ο υπολογισμός της ενέργειας στο άπειρο βάσει της (4.12) είναι :

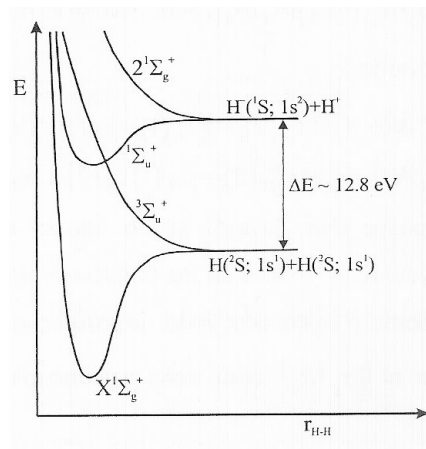
$E = \frac{2E_H + E_{H^-+H^+}}{2}$. Σύμφωνα όμως με τα παραπάνω, η ενέργεια θα έπρεπε να είναι $2E_H$. Για να διορθώσουμε, λοιπόν, τη συμπεριφορά στο άπειρο, θα πρέπει να “αναμιξούμε” την $|1\sigma_g 1\bar{\sigma}_g|$ με οριζουσες κατάλληλης συμμετρίας ώστε να εξαλειφθούν οι ιοντικοί όροι. Ένα παράδειγμα είναι:

$$\Psi(\mathbf{x}_1, \mathbf{x}_2) = \sum_{n=1}^2 c_n |\Phi_n\rangle = c_1 |1\sigma_g 1\bar{\sigma}_g| + c_2 |1\sigma_u 1\bar{\sigma}_u| \quad (4.13)$$

Σε αυτό συνίσταται ουσιαστικά η αλληλεπίδραση απεικονίσεων (CI) στην ανάμιξη κατάλληλων οριζουσών Slater, όπου κάθε ορίζουσα λαμβάνεται με διαφορετική βαρύτητα. Ο λόγος που λαμβάνουμε τη συγκεκριμένη ορίζουσα γίνεται προφανής αν την αναπτύξουμε όπως προηγουμένως:

$$\begin{aligned} |1\sigma_u 1\bar{\sigma}_u| &= |(s_a(1) - s_b(1))(\bar{s}_a(2) - \bar{s}_b(2))| \\ &= |s_a(1)\bar{s}_a(2)| - |s_a(1)\bar{s}_b(2)| - |s_b(1)\bar{s}_a(2)| + |s_b(1)\bar{s}_b(2)| \end{aligned} \quad (4.14)$$

⁷ Αυτό φαίνεται από την (4.12) όπου στην πρώτη και τέταρτη ορίζουσα τα δύο ηλεκτρόνια τοποθετούνται στο τροχιακό ενός εκ των δύο ατόμων, ενώ αντίθετα στη δεύτερη και στη τρίτη τα ηλεκτρόνια ισοκατανέμονται μεταξύ των δύο ατόμων[6].



Σχήμα 4.4: Γραφική παράσταση της ενέργειας συναρτήσει της διαπυρηνικής αποστάσεως, των καταστάσεων $1^1\Sigma_g^+$, $1^1\Sigma_u^+$, $2^1\Sigma_g^+$ και $3^1\Sigma_u^+$, για το H_2 .

οπότε αν στην (4.13) αντικαταστήσουμε $c_1 = -c_2$, οι ιοντικοί όροι αλληλοαναιρούνται και απομένουν οι ομοιοπολικοί⁸.

Στο επόμενο τμήμα εξετάζουμε την μέθοδο *CI* πιο διεξοδικά, εφόσον θα τη χρησιμοποιήσουμε στον υπολογισμό των συντελεστών ανάπτυξης των κυματοσυναρτίσεων Ψ .

4.3 Αλληλεπίδραση Απεικονίσεων

Ξεκινάμε την περιγραφή της μεθόδου *CI*, με έναν αρκετά περιεκτικό ορισμό της:

Ορισμός 4.3.1. Η *CI* συνίσταται στην ανάπτυξη της κυματοσυναρτίσεως ως γραμμικό συνδυασμό οριζουσών Slater Φ_n κατάλληλης συμμετρίας και spin, έκαστη πολλαπλασιασμένη με ένα συντελεστή c_n , οι οποίοι προσδιορίζονται ελαχιστοποιώντας την έκφραση της συνολικής ενέργειας ως προς αυτούς.

⁸ Τα παραπάνω βέβαια ισχύουν για $R \rightarrow \infty$ στην ισορροπία $c_2 \ll c_1$, δεδομένου ότι η $|\sigma_g 1\bar{\sigma}_g|$ είναι πολύ καλή προσέγγιση της πραγματικής κυματοσυναρτίσεως[6].

4.3.1 Πλήρης κυματοσυνάρτηση CI

Έστω $|\Psi_0\rangle$ η ακριβής πολυ-ηλεκτρονιακή κυματοσυνάρτηση, τότε αν η $|\Phi_0\rangle$ είναι μία καλή προσέγγιση αυτής, γνωρίζουμε από τη θεωρία μεταβολών πως μια καλύτερη προσέγγιση είναι :

$$|\Psi_0\rangle = c_0|\Phi_0\rangle + \sum_{ar} c_a^r |\Phi_a^r\rangle + \sum_{\substack{a<b \\ r<s}} c_{ab}^{rs} |\Phi_{ab}^{rs}\rangle + \sum_{\substack{a<b<c \\ r<s<t}} c_{abc}^{rst} |\Phi_{abc}^{rst}\rangle + \dots \quad (4.15)$$

όπου $|\Phi_a^r\rangle$: είναι μία απλά διεγερμένη ορίζουσα⁹, $|\Phi_{ab}^{rs}\rangle$: είναι μια διπλά διεγερμένη ορίζουσα κ.ο.κ. Οι περιορισμοί στα αθροίσματα επιβάλλονται ώστε μια συγκεκριμένη ορίζουσα να ληφθεί υπ' όψιν μία φορά μόνο¹⁰.

Στον υπολογισμό θα χρησιμοποιήσουμε μια ανάλογη ανάπτυξη CI όπου θέτουμε ως ορίζουσες Slater τις D_i^p , η οποίες αποτελούνται από τα παραμετροποιημένα spin-τροχιακά $\{\psi_k(\mathbf{x}_j) = u_k(\mathbf{r}_j)s(\omega)\}$, όπου $\mathbf{x}_j = \{\mathbf{r}_j, \omega\}$ και

$$s(\omega) = \begin{cases} \alpha & , \text{ για spin πάνω,} \\ \beta & , \text{ για spin κάτω.} \end{cases} \quad (4.16)$$

Άρα, αν έχουμε ένα σύνολο M χωρικών τροχιακών $\{u_k|k = 1, 2, \dots, M\}$ μπορούμε να σχηματίσουμε από αυτό ένα σύνολο $2M$ spin-τροχιακών $\{\psi_k|k = 1, 2, \dots, 2M\}$, ενώ η συνθήκη ορθοκανονικότητάς¹¹ τους είναι :

$$\int \psi_i^*(\mathbf{x}_k)\psi_j(\mathbf{x}_k) d\mathbf{x} \equiv \langle \psi_i | \psi_j \rangle = \delta_{ij} \quad (4.17)$$

⁹ Είναι σαν την $|\Phi_0\rangle$ με τη διαφορά ότι ένα ηλεκτρόνιο έχει διεγερθεί από το \mathbf{x}_a στο \mathbf{x}_r spin-τροχιακό' δηλ. στην ουσία αντικαταστάθηκε στην $|\Phi_0\rangle$ το \mathbf{x}_a με το \mathbf{x}_r , ομοίως και για τις n -πλές διεγερμένες ορίζουσες[4].

¹⁰ Εννοείται, φυσικά, πως η (4.15) πλησιάζει την ακριβή όσο το σύνολο βάσης τείνει να γίνει πλήρες.

¹¹ Αυτό ισχύει επειδή οι δύο συναρτήσεις spin $\alpha(\omega)$ και $\beta(\omega)$ είναι πλήρεις και ορθοκανονικές : $\int s_i^*(\omega)s_j(\omega) d\omega = \langle s_i | s_j \rangle = \delta_{ij}$, όπου $s_i = \alpha$ ή β .

Τέλος, υπάρχουν και οι συντελεστές a_{pi} των D_{pi} ¹², που καθορίζουν τη συμμετρία της κατάστασης. Κάθε απεικόνιση σχηματίζεται από μία κεντρική απεικόνιση μέσω απλών, διπλών και γενικώς πολλαπλών διεγέρσεων, και για κάθε διεγερση υπολογίζονται όλες οι γραμμικά ανεξάρτητες (συμμετρικά αποδεκτές) απεικονίσεις¹³.

$$|\Phi_p\rangle = \sum_{i=1}^K a_{pi}|D_{pi}\rangle \quad (4.18\alpha')$$

$$\begin{aligned} |\Psi\rangle &= \sum_{p=0}^{n_{cnf}} c_p |\Phi_p\rangle = c_0 |\Phi_0\rangle + c_1 |\Phi_1\rangle + c_2 |\Phi_2\rangle + c_3 |\Phi_3\rangle + \dots \\ &= c_0 \sum_{i=1}^K a_{0i} |D_{0i}\rangle + c_1 \sum_{i=1}^K a_{1i} |D_{1i}\rangle \\ &\quad + c_2 \sum_{i=1}^K a_{2i} |D_{2i}\rangle + c_3 \sum_{i=1}^K a_{3i} |D_{3i}\rangle + \dots \end{aligned} \quad (4.18\beta')$$

$$\langle \Psi | \Psi \rangle = \int \dots \int \Psi^*(\mathbf{x}_1, \dots, \mathbf{x}_n) \Psi(\mathbf{x}_1, \dots, \mathbf{x}_n) d\mathbf{x}_1 \dots d\mathbf{x}_n = 1 \quad (4.18\gamma')$$

Το επόμενο στάδιο για την πληρέστερη περιγραφή της μεθόδου είναι ο ορισμός των *συντελεστών Lagrange*, που θα μας επιτρέψει να υπολογίσουμε τις βέλτιστες τιμές των συντελεστών c_p .

4.3.2 Συντελεστές Lagrange

Το πρόβλημα είναι να καθορίσουμε για ποιές τιμές των συντελεστών c_p και των διαφορών παραμέτρων των spin-τροχιακών, η αναμενόμενη τιμή της Χαμιτονιανής, $E = \langle \Psi | \hat{H} | \Psi \rangle$, γίνεται ελάχιστη. Εφόσον, βάσει της (4.19), επιτρέψαμε μόνο γραμμικές μεταβολές στην $|\Psi\rangle$, όπου το $\{|\Phi_p\rangle\}$ είναι ένα δεδομένο σύνολο $\{n_{cnf} \equiv N\}$ συναρτήσεων βάσης, τότε το πρόβλημα να βρούμε το σύνολο των βέλτιστων συντελεστών, $\{c_p\}$,

¹²Το p εκτός από δείκτης του αθροίσματος υποδεικνύει και την διεγερση : $p = 1 \rightarrow$ απλή διεγερση κ.ο.κ.

¹³ Αυτός ο υπολογισμός γίνεται μέσω του προγράμματος QCPE HEDIAG για περισσότερες πληροφορίες βλ. [2]

μπορεί να αναχθεί σε διαγωνοποίηση μήτρας.

Εφόσον $E[\Psi] = \langle \Psi | \hat{H} | \Psi \rangle$, το $E[\Psi]$ είναι συναρτησιοειδές της Ψ . Άν λοιπόν, μεταβάλλουμε την Ψ κατά ένα αυθαίρετα μικρό ποσό¹⁴, έχουμε:

$$\Psi \rightarrow \Psi + \delta\Psi \quad (4.19)$$

οπότε η ενέργεια γίνεται

$$\begin{aligned} E[\Psi + \delta\Psi] &= \langle \Psi + \delta\Psi | \hat{H} | \Psi + \delta\Psi \rangle \\ &= E[\Psi] + [\langle \delta\Psi | \hat{H} | \Psi \rangle + \langle \Psi | \hat{H} | \delta\Psi \rangle] + \dots \\ &= E[\Psi] + \delta E + \dots \end{aligned} \quad (4.20)$$

όπου δE : η πρώτη μεταβολή της E , που περιλαμβάνει όλους τους γραμμικούς όρους. Η συνθήκη¹⁵ για την ελαχιστοποίηση της $E[\Psi]$ ως προς τη Ψ είναι:

$$\delta E = 0 \quad (4.21)$$

Για τη δοκιμαστική κυματοσυνάρτηση

$$|\Psi\rangle = \sum_{p=1}^N c_p |\Phi_p\rangle \quad (4.22)$$

έχουμε:

- Στη περίπτωση που οι συναρτήσεις βάσης $\{|\Phi_p\rangle\} \in \mathbb{R}$ και είναι ορθοκανονικές

$$\langle \Phi_p | \Phi_q \rangle = \langle \Phi_q | \Phi_p \rangle = \delta_{pq} \quad (4.23)$$

ενώ η αναπαράσταση της Χαμιλτονιανής στη βάση $\{|\Phi_p\rangle\}$ είναι μία συμμετρική¹⁶ μήτρα \mathbf{H} , διάστασης $N \times N$, με μητροστοιχεία

$$(\mathbf{H})_{pq} = H_{pq} = H_{qp} = \langle \Phi_p | \hat{H} | \Phi_q \rangle \quad (4.24)$$

¹⁴ Αυτό το κάνουμε στη πράξη μεταβάλλοντας τις παραμέτρους από τις οποίες εξαρτάται η Ψ , π.χ. z_a^*, z_b^* κ.λ.π.

¹⁵ Η συνθήκη αυτή μας δίνει τη στάσιμη τιμή της E για αυθαίρετη μεταβολή της Ψ . Συνήθως, βέβαια, το στάσιμο σημείο είναι και ελάχιστο.

¹⁶ Εφόσον η Χαμιλτονιανή είναι Ερμιτιανή και η βάση πραγματική.

Θέλουμε να ελαχιστοποιήσουμε την έκφραση της ενέργειας

$$E = \langle \Psi | \hat{H} | \Psi \rangle = \sum_{p=1}^N \sum_{q=1}^N c_p^* c_q \langle \Phi_p | \hat{H} | \Phi_q \rangle \xrightarrow{(4.24)} \sum_{p=1}^N \sum_{q=1}^N c_p^* c_q H_{pq} \quad (4.25)$$

$$= \sum_{p=1}^N \sum_{q=1}^N c_p^* c_q \int \cdots \int \Phi_p^*(\mathbf{x}_1, \dots, \mathbf{x}_n) \hat{H} \Phi_q(\mathbf{x}_1, \dots, \mathbf{x}_n) d\mathbf{x}_1 \cdots d\mathbf{x}_n$$

υπό τον περιορισμό ότι η δοκιμαστική κυματοσυνάρτηση να παραμένει κανονικοποιημένη :

$$\langle \Psi | \Psi \rangle = 1 \Leftrightarrow \sum_{p=1}^N \sum_{q=1}^N c_p^* c_q \langle \Phi_p | \Phi_q \rangle = 1 \quad (4.26)$$

$$\xrightarrow{(4.23)} \sum_{p=1}^N \sum_{q=1}^N c_p^* c_q \delta_{pq} = 1 \Rightarrow \sum_{p=1}^N |c_p|^2 = 1 \quad (4.27)$$

Όμως δεν μπορούμε απλά να λύσουμε το σύνολο εξισώσεων

$$\frac{\partial}{\partial c_k} \langle \Phi_p | \hat{H} | \Phi_q \rangle = 0 \quad (4.28)$$

με $k = 1, 2, \dots, N$, εφόσον υπάρχει η δεσμευτική σχέση (4.27) με αποτέλεσμα οι N παράμετροι να μίν είναι όλες γραμμικά ανεξάρτητες, αλλά μόνο $N - 1$ εξ αυτών. Σε περιπτώσεις όπου θέλουμε να ελαχιστοποιήσουμε ένα συναρτησοειδές (4.25) που υπόκειται σε κάποιο περιορισμό (4.27) χρησιμοποιούμε τη μέθοδο των απροσδιορίστων συντελεστών *Lagrange*¹⁷: αρχικά ορίζουμε το συναρτησοειδές :

$$\begin{aligned} \mathcal{L}(c_1, \dots, c_N, E) &= \langle \Psi | \hat{H} | \Psi \rangle - E (\langle \Psi | \Psi \rangle - 1) \\ &= \sum_{p=1}^N \sum_{q=1}^N c_p^* c_q \langle \Phi_p | \hat{H} | \Phi_q \rangle - E \left(\sum_{p=1}^N \sum_{q=1}^N c_p^* c_q \langle \Phi_p | \Phi_q \rangle - 1 \right) \\ &= \sum_{p=1}^N \sum_{q=1}^N c_p^* c_q H_{pq} - E \left(\sum_{p=1}^N |c_p|^2 - 1 \right) \end{aligned} \quad (4.29)$$

όπου E είναι ο πολλαπλασιαστής *Lagrange*¹⁷. Αν επιλέξουμε αυθαίρετα τις c_1, c_2, \dots, c_{N-1} ως γραμμικά ανεξάρτητες, οπότε το c_N

¹⁷ Ουσιαστικά προσθέσαμε το μηδέν στην (4.36), εφόσον η Ψ είναι κανονικοποιημένη, δηλ. πρακτικά είναι το ίδιο συναρτησοειδές κατά συνέπεια το ελάχιστο, τόσο του E όσο και του \mathcal{L} , βρίσκεται για τις ίδιες τιμές των συντελεστών[4].

να καθορίζεται από τη συνθήκη κανονικοποίησης (4.38), τότε ελαχιστοποίηση του \mathcal{L} (αντί του E) ως προς το c_k θα μας δώσει:

$$\delta\mathcal{L} = \sum_{k=1}^N \frac{\partial\mathcal{L}}{\partial c_k} \delta c_k = 0, \quad \forall \delta c_k \quad (4.30)$$

ενώ η αναγκαία και ικανή συνθήκη για να ισχύει αυτό είναι :

$$\frac{\partial\mathcal{L}}{\partial c_k} = 0, \quad k = 1, 2, \dots, N-1, N \quad (4.31)$$

όμως το $\frac{\partial\mathcal{L}}{\partial c_N}$ δεν είναι απαραίτητα μηδέν. Έδώ διαφαίνεται και ο ρόλος των “απροσδιορίστων” παλλαπλασιαστών Lagrange: επιλέγουμε το E έτσι ώστε $\frac{\partial\mathcal{L}}{\partial c_N} = 0$. Αντικαθιστώντας την (4.29) έχουμε:

$$\begin{aligned} \delta\mathcal{L} &= \sum_{p=1}^N \sum_{q=1}^N \frac{\partial c_p}{\partial c_k} c_q H_{pq} + \sum_{p=1}^N \sum_{q=1}^N c_p \frac{\partial c_q}{\partial c_k} H_{qp} - E \sum_{p=1}^N 2c_p \frac{\partial c_p}{\partial c_k} = 0 \\ &\Rightarrow \sum_{q=1}^N c_q H_{kq} + \sum_{p=1}^N c_p H_{pk} - 2Ec_k = 0 \\ &\xrightarrow{(4.25)} \sum_p H_{pq} - Ec_p = 0 \quad (4.32) \end{aligned}$$

Τέλος, εισάγοντας το διάνυσμα στήλη \mathbf{c} με στοιχεία τα $\{c_p\}$, το σύνολο εξισώσεων (4.32) γράφεται ως το κλασικό πρόβλημα ιδιοτιμής :

$$\mathbf{H}\mathbf{c} = E\mathbf{c} \quad (4.33)$$

- Στη γενική περίπτωση όπου τα $\{|\Phi_p\rangle\} \in \mathbb{C}$ και δεν είναι ορθοκανονικά έχουμε :

$$\mathcal{H}_{pq} = \langle \Phi_p | \hat{H} | \Phi_q \rangle \quad (4.34)$$

$$\mathcal{S}_{pq} = \langle \Phi_p | \Phi_q \rangle \quad (4.35)$$

όπου η (4.35) είναι το ολοκλήρωμα επικαλύψεως. Ομοίως με τα προηγούμενα, θέλουμε να ελαχιστοποιήσουμε την έκφραση της ενέργειας

$$\begin{aligned} E = \langle \Psi | \hat{H} | \Psi \rangle &= \sum_{p=1}^N \sum_{q=1}^N c_p^* c_q \langle \Phi_p | \hat{H} | \Phi_q \rangle \\ &= \sum_{p=1}^N \sum_{q=1}^N c_p^* c_q \int \cdots \int \Phi_p^*(\mathbf{x}_1, \dots, \mathbf{x}_n) \hat{H} \Phi_q(\mathbf{x}_1, \dots, \mathbf{x}_n) d\mathbf{x}_1 \cdots d\mathbf{x}_n \end{aligned} \quad (4.36)$$

υπό τον περιορισμό ότι η δοκιμαστική κυματοσυνάρτηση να παραμένει κανονικοποιημένη :

$$\langle \Psi | \Psi \rangle = 1 \Leftrightarrow \quad (4.37)$$

$$\langle \Psi | \Psi \rangle - 1 = \sum_{p=1}^N \sum_{q=1}^N c_p^* c_q \langle \Phi_p | \Phi_q \rangle - 1 = 0 \quad (4.38)$$

$$\Leftrightarrow \sum_{p=1}^N \sum_{q=1}^N c_p^* c_q \int \cdots \int \Phi_p^*(\mathbf{x}_1, \dots, \mathbf{x}_n) \Phi_q(\mathbf{x}_1, \dots, \mathbf{x}_n) d\mathbf{x}_1 \cdots d\mathbf{x}_n - 1 = 0$$

Κάνουμε χρήση της μεθόδου Lagrange, όπου αντί να ελαχιστοποιήσουμε την

$$\delta E = 0 \Rightarrow \delta \langle \Phi_p | \hat{H} | \Phi_q \rangle = 0 \quad (4.39)$$

ελαχιστοποιούμε το παρακάτω συναρτησιοειδές, ως προς τους συντελεστές c_p

$$\begin{aligned} \mathcal{L}(c_1, \dots, c_N, E) &= \langle \Psi | \hat{H} | \Psi \rangle - E (\langle \Psi | \Psi \rangle - 1) \\ &= \sum_{p=1}^N \sum_{q=1}^N c_p^* c_q \langle \Phi_p | \hat{H} | \Phi_q \rangle - E \left(\sum_{p=1}^N \sum_{q=1}^N c_p^* c_q \langle \Phi_p | \Phi_q \rangle - 1 \right) \end{aligned} \quad (4.40)$$

Θέτοντας την μεταβολή πρώτης τάξης ίση με το μηδέν

$$\begin{aligned} \delta \mathcal{L} &= \sum_{p=1}^N \sum_{q=1}^N \delta c_p^* c_q \langle \Phi_p | \hat{H} | \Phi_q \rangle - E \sum_{p=1}^N \sum_{q=1}^N \delta c_p^* c_q \langle \Phi_p | \Phi_q \rangle \\ &+ \sum_{p=1}^N \sum_{q=1}^N c_p^* \delta c_q \langle \Phi_p | \hat{H} | \Phi_q \rangle - E \sum_{p=1}^N \sum_{q=1}^N c_p^* \delta c_q \langle \Phi_p | \Phi_q \rangle = 0 \end{aligned} \quad (4.41)$$

ενώ συλλέγοντας τους όρους και αλλάζοντας τους δείκτες του δεύτερου σκέλους της εξίσωσης $p \leftrightarrow q$, παίρνουμε :

$$\begin{aligned} \delta \mathcal{L} = & \sum_{p=1}^N \delta c_p^* \left[\sum_{q=1}^N (\mathcal{H}_{pq} c_q - E \mathcal{S}_{pq} c_q) \right] \\ & + \underbrace{\sum_{p=1}^N \delta c_p \left[\sum_{q=1}^N (\mathcal{H}_{qp} c_q^* - E \mathcal{S}_{qp} c_q^*) \right]}_{\text{μγαδικός συζυγής}} = 0 \end{aligned} \quad (4.42)$$

Εφόσον τα δc_p^* ¹⁸ τα πήραμε αυθαίρετα εξ αρχής, η ποσότητα εντός των αγκυλών θα πρέπει να είναι μηδέν. Άρα :

$$\sum_{q=1}^N \mathcal{H}_{pq} c_q = E \sum_{q=1}^N \mathcal{S}_{pq} c_q \quad (4.43\alpha')$$

$$\sum_{q=1}^N \mathcal{H}_{qp} c_q^* = E \sum_{q=1}^N \mathcal{S}_{qp} c_q^* \quad (4.43\beta')$$

ή αντίστοιχα καταλίγουμε στο πρόβλημα γενικευμένης ιδιοτιμής με μιγαδικούς συντελεστές :

$$\mathbf{Hc} = E\mathbf{Sc} \quad (4.44)$$

όπου \mathbf{S} είναι η μήτρα επικαλύψεως, με στοιχεία τα $(\mathbf{S})_{pq} = \mathcal{S}_{pq}$. Η επίλυση της παραπάνω εξίσωσης μπτροών, μέσω της αναγωγής της σε απλό πρόβλημα ιδιοτιμής, περιγράφεται στο κεφάλαιο 5.

Στο επόμενο μέρος θα ασχοληθούμε με την ανάπτυξη των απαραίτητων μπτοστοιχείων για τον υπολογισμό του συναρτησειοειδούς \mathcal{F} (κεφ. 1, σελ. 3), καθώς και τις διάφορες απλοποιητικές συμβάσεις που θα διευκολύνουν τον τελικό υπολογισμό.

¹⁸ Στην (4.42) τα c_p^* και c_p αντιμετωπίζονται ως γραμμικά ανεξάρτητες μεταβλητές εφόσον είναι μιγαδικά. Αυτό γίνεται εμφανές αν θέσουμε $c_p = a+ib$, $c_p^* = a-ib$, με $a, b \in \mathbb{R}$ · τότε $\frac{\partial}{\partial c_p^*} \equiv \frac{1}{2} \left(\frac{\partial}{\partial a} + i \frac{\partial}{\partial b} \right) \rightarrow \frac{\partial c_p}{\partial c_p^*} = 0$.

Μέρος III
Κυρίως Μέρος

Κεφάλαιο 5

Ανάπτυξη Μητροστοιχείων

5.1 Εισαγωγή

Στο κεφάλαιο αυτό δείχνουμε πώς προκύπτουν τα διάφορα μητροστοιχεία που θα χρειαστούν στον υπολογισμό και τα αναπτύσσουμε αναλυτικά. Χωρίζουμε τα μητροστοιχεία σε ολοκληρώματα επικαλύψεως, μονοπλεκτρονιακά και διπλεκτρονιακά, με τελικό σκοπό τον προγραμματισμό τους και την ταχύτερη υπολογιστική επίλυση του συνολικού προβλήματος .

5.2 Αναμενόμενη Τιμή

Στο κεφάλαιο 4 είδαμε πώς προσεγγίζουμε την κυματοσυνάρτηση με μία ανάπτυξη CI η οποία αποτελείται από N_{cnf} απεικονίσεις [2]:

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) = \sum_{p=1}^{N_{cnf}} c_p \Phi_p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n), \quad (5.1\alpha')$$

$$\Phi_p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) = \sum_{i=1}^{K(N_{cnf})} a_{pi} D_{pi}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n), \quad (5.1\beta')$$

$$\langle \Psi | \Psi \rangle = 1 \Rightarrow \iint_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \Psi^*(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \Psi(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) d\mathbf{x}_1 d\mathbf{x}_2 \cdots d\mathbf{x}_n = 1 \quad (5.1\gamma')$$

$$\{\mathbf{x}_i\} = \{(\mathbf{r}_i, \omega)\} \quad (5.1\delta')$$

όπου D_{pi} : ορίζουσες Slater που αποτελούνται από παραμετροποιημένα ορθοκανονικά spin-τροχιακά $\{\psi_k(\mathbf{x}_j) = u_k(\mathbf{r}_j)s(\omega)\}$,

$$s(\omega) = \begin{cases} \alpha & , \text{ για spin πάνω,} \\ \beta & , \text{ για spin κάτω.} \end{cases} \quad (5.2)$$

Η Χαμιλτονιανή δύο κέντρων γράφεται ως εξής (κεφ. 4.2, σελ. 35):

$$\hat{H} = \sum_{i=1}^n \hat{h}_i + \sum_{i=1}^n \sum_{\substack{j=1 \\ i>j}}^n \frac{1}{r_{ij}}, \quad (5.3\alpha')$$

$$\hat{h}_i = -\frac{1}{2} \nabla_i^2 - \frac{Z_a}{r_{ai}} - \frac{Z_b}{r_{bi}}, \quad (5.4)$$

όπου \hat{h}_i : μονοπλεκτρονιακοί τελεστές, $\frac{1}{r_{ij}} \equiv \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}$, $\frac{1}{r_{li}} \equiv \frac{1}{|\mathbf{r}_i - \mathbf{R}_l|}$, Z_a, Z_b : τα πυρηνικά φορτία των πυρήνων a, b αντίστοιχα και $r_{x,y}$ η απόσταση μεταξύ των σωματιδίων x και y .

Η αναμενόμενη τιμή της πλεκτρονιακής Χαμιλτονιανής είναι :

$$E = \langle \Psi | \hat{H} | \Psi \rangle \stackrel{(5.1\alpha')}{=} \sum_{p=1}^{N_{cnf}} \sum_{q=1}^{N_{cnf}} c_p c_q \langle \Phi_p | \hat{H} | \Phi_q \rangle = \sum_{p=1}^{N_{cnf}} \sum_{q=1}^{N_{cnf}} c_p c_q \mathcal{H}_{pq} \Rightarrow \quad (5.5\alpha')$$

$$\stackrel{(5.1\beta')}{\Rightarrow} \sum_{p=1}^{N_{cnf}} \sum_{q=1}^{N_{cnf}} c_p c_q \sum_{i=1}^K \sum_{j=1}^K a_{pi} a_{qj} \langle D_{pi} | \hat{H} | D_{qj} \rangle \quad (5.5\beta')$$

$$\text{με } \mathcal{H}_{pq} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \Phi_p^*(\mathbf{x}_1, \dots, \mathbf{x}_n) \hat{H} \Phi_q(\mathbf{x}_1, \dots, \mathbf{x}_n) d\mathbf{x}_1 \cdots d\mathbf{x}_n \quad (5.5\gamma')$$

5.2.1 Διαγωνοποίηση της \mathcal{H}_{pq}

Όπως δείξαμε στο υποκεφάλαιο 4.3.2, η βελτιστοποίηση της ενέργειας ως προς τους συντελεστές c_p , είναι ισοδύναμη με το γενικευμένο πρόβλημα ιδιοτιμής (4.44)

$$\mathbf{Hc} = E\mathbf{Sc} \quad (5.6)$$

όπου \mathbf{S} είναι η μήτρα επικάλυψης, με στοιχεία τα $(\mathbf{S})_{pq} = \mathcal{S}_{pq}$, για τα οποία ισχύει :

$$\mathcal{S}_{pq} = \int \cdots \int \Phi_p^*(\mathbf{x}_1, \dots, \mathbf{x}_n) \Phi_q(\mathbf{x}_1, \dots, \mathbf{x}_n) d\mathbf{x}_1 \cdots d\mathbf{x}_n$$

Η (4.44) μπορεί εύκολα να αναχθεί σε ένα σύνθετο πρόβλημα ιδιοτιμής, εφόσον η \mathbf{S} είναι Ερμιτιανή και θετικά ορισμένη¹, μέσω ενός μετασχηματισμού *Cholesky* [33], $\mathbf{S} = \mathbf{L}\mathbf{L}^\dagger$, όπου \mathbf{L} είναι μία κάτω τριγωνική μήτρα² με στοιχεία $(\mathbf{L})_{ij} : \mathbb{R} \times \{(\mathbf{L})_{ii} \in \mathbb{R}^+\}$ · κάθε Ερμιτιανή θετικά ορισμένη μήτρα έχει έναν μοναδικό μετασχηματισμό *Cholesky*³· επιπλέον, επειδή $(\mathbf{S})_{ij} \in \mathbb{R} \rightarrow (\mathbf{L})_{ij} \in \mathbb{R}$, ισχύει ότι $\mathbf{L}^\dagger \equiv \mathbf{L}^\top \rightarrow \mathbf{S} = \mathbf{L}\mathbf{L}^\top$.

Θέτοντας $\mathbf{S} = \mathbf{L}\mathbf{L}^\dagger$ στην (4.44), έχουμε

$$\begin{aligned} \mathbf{H}\mathbf{c} &= \mathbf{E}\mathbf{L} \underbrace{\mathbf{L}^\dagger \mathbf{c}}_{\mathbf{c}'} \\ \Rightarrow \mathbf{H}(\mathbf{L}^\dagger)^{-1} \underbrace{\mathbf{L}^\dagger \mathbf{c}}_{\mathbf{c}'} &= \mathbf{E}\mathbf{L}\mathbf{c}' \\ \Rightarrow \underbrace{\mathbf{L}^{-1}\mathbf{H}(\mathbf{L}^\dagger)^{-1}}_{\mathbf{H}'} \mathbf{c}' &= \mathbf{E} \underbrace{\mathbf{L}^{-1}\mathbf{L}}_{\mathbf{I}} \mathbf{c}' \\ \Rightarrow \mathbf{H}'\mathbf{c}' &= \mathbf{E}\mathbf{c}' \end{aligned} \quad (5.7)$$

1

Ορισμός 5.2.1. Κάθε συμμετρική $n \times n$ Ερμιτιανή μήτρα \mathbf{S} λέγεται θετικά ορισμένη (*positive definite*), εάν $\mathbf{c}^\dagger \mathbf{S} \mathbf{c} \in \mathbb{R}^+$, $\forall \{\mathbf{c} \in \mathbb{C} : \mathbf{c} \neq 0\}$

² Η τριγωνική μήτρα είναι μια ειδική περίπτωση τετραγωνικής μήτρας·

Ορισμός 5.2.1. Μία τετραγωνική μήτρα καλείται κάτω τριγωνική, αν όλα τα στοιχεία πάνω από τη κύρια διαγώνιο είναι μηδέν. Μία κάτω τριγωνική μήτρα συμβολίζεται ως \mathbf{L} (*Lower*)· έτσι, έστω \mathbf{S} : τετραγωνική μήτρα, αν $(\mathbf{S})_{ij} = 0, \forall \{i = 1, \dots, n-1, j = i+1\}$, τότε $\mathbf{S} \equiv \mathbf{L}$.

Ορισμός 5.2.1. Ομοίως, μία τετραγωνική μήτρα καλείται πάνω τριγωνική, αν όλα τα στοιχεία κάτω από τη κύρια διαγώνιο είναι μηδέν. Μία πάνω τριγωνική μήτρα συμβολίζεται ως \mathbf{U} (*Upper*)· έτσι, έστω \mathbf{S} : τετραγωνική μήτρα, αν $(\mathbf{S})_{ij} = 0, \forall \{i = 1, \dots, n, j = i-1\}$, τότε $\mathbf{S} \equiv \mathbf{U}$.

³ Επομένως και κάθε συμμετρική, θετικά ορισμένη μήτρα, με πραγματικές τιμές.

δηλ. καταλήξαμε σε ένα κλασσικό πρόβλημα ιδιοτιμής, χωρίς να αλλάξει η ιδιοτιμή, εφόσον στην ουσία κάναμε έναν μοναδιαίο μετασχηματισμό.

Μπαίνουμε, τώρα στο κύριο μέρος αυτού του κεφαλαίου, που δεν είναι άλλο από την διεξοδική ανάπτυξη των μητροστοιχείων που θα προκύψουν την εισαγωγή των οριζουσών Slater.

5.2.2 Ανάπτυξη των οριζουσών Slater

Τα μητροστοιχεία $\langle D_{pi} | \hat{H} | D_{qj} \rangle$, που είναι ολοκληρώματα μεταξύ των οριζουσών Slater, ανάγονται στα ολοκληρώματα επικαλύψεως, μονοπλεκτρονιακά και διπλεκτρονιακά: έστω $\{\phi_l(\mathbf{x}_k) = u_l(\mathbf{r}_k)s(\omega_k)\}$ τα spin-τροχιακά της “αριστερής” ορίζουσας (D_{pi}) και $\{\psi_n(\mathbf{x}_m) = v_n(\mathbf{r}_m)s(\omega_m)\}$ της “δεξιάς” (D_{qj}). Τότε, η ορίζουσα γράφεται (κεφ. 4.6, σελ. 37):

$$D_{pi}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_N(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_N(\mathbf{x}_2) \\ \vdots & \vdots & & \vdots \\ \phi_1(\mathbf{x}_N) & \phi_2(\mathbf{x}_N) & \cdots & \phi_N(\mathbf{x}_N) \end{vmatrix} \quad (5.8)$$

και

$$D_{qj}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \psi_1(\mathbf{x}_1) & \psi_2(\mathbf{x}_1) & \cdots & \psi_N(\mathbf{x}_1) \\ \psi_1(\mathbf{x}_2) & \psi_2(\mathbf{x}_2) & \cdots & \psi_N(\mathbf{x}_2) \\ \vdots & \vdots & & \vdots \\ \psi_1(\mathbf{x}_N) & \psi_2(\mathbf{x}_N) & \cdots & \psi_N(\mathbf{x}_N) \end{vmatrix} \quad (5.9)$$

Κατόπιν, αναπτύσσουμε τις ορίζουσες Slater σύμφωνα με την (4.7):

$$D_{qj}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \sum_a^N \sum_b^N \cdots \sum_c^N \varepsilon_{ab\dots c} \psi_a(\mathbf{x}_1) \psi_b(\mathbf{x}_2) \cdots \psi_c(\mathbf{x}_N) \quad (5.10)$$

$$D_{pi}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \sum_d^N \sum_e^N \cdots \sum_f^N \varepsilon_{de\dots f} \phi_d(\mathbf{x}_1) \phi_e(\mathbf{x}_2) \cdots \phi_f(\mathbf{x}_N) \quad (5.11)$$

Στο πρόγραμμα τα συμβολα Levi-Civita υπολογίζονται με εύρεση όλων των δυνατών μεταθεσεων των +/- επί του πλήθους των τροχιακών εκάστης ορίζουσας Slater.

Θέτοντας τις (5.11), (5.10) και (5.3α') στην $\langle D_{pi}|\hat{H}|D_{qj}\rangle$ έχουμε :

$$\begin{aligned}
& \langle D_{pi}|\hat{H}|D_{qj}\rangle = \\
& = \frac{1}{N!} \left\langle \sum_{d=1}^N \sum_{e=1}^N \cdots \sum_{f=1}^N \varepsilon_{de\dots f} \phi_d(\mathbf{x}_1) \phi_e(\mathbf{x}_2) \cdots \phi_f(\mathbf{x}_N) \left| \sum_{i=1}^N \hat{h}_i \right. \right. \\
& \quad \left. \left. + \sum_{\substack{i=1 \\ i>j}}^N \frac{1}{r_{ij}} \left| \sum_{a=1}^N \sum_{b=1}^N \cdots \sum_{c=1}^N \varepsilon_{ab\dots c} \psi_a(\mathbf{x}_1) \psi_b(\mathbf{x}_2) \cdots \psi_c(\mathbf{x}_N) \right. \right. \right\rangle \\
& = \frac{1}{N!} \sum_{a=1}^N \sum_{b=1}^N \cdots \sum_{c=1}^N \sum_{d=1}^N \sum_{e=1}^N \cdots \sum_{f=1}^N \varepsilon_{ab\dots c} \varepsilon_{de\dots f} \\
& \quad \times \left(\langle \phi_d(\mathbf{x}_1) | \hat{h}_1 | \psi_a(\mathbf{x}_1) \rangle \langle \phi_e(\mathbf{x}_2) | \psi_b(\mathbf{x}_2) \rangle \cdots \langle \phi_f(\mathbf{x}_N) | \psi_c(\mathbf{x}_N) \rangle \right. \\
& \quad + \langle \phi_d(\mathbf{x}_1) | \psi_a(\mathbf{x}_1) \rangle \langle \phi_e(\mathbf{x}_2) | \hat{h}_2 | \psi_b(\mathbf{x}_2) \rangle \cdots \langle \phi_f(\mathbf{x}_N) | \psi_c(\mathbf{x}_N) \rangle + \cdots \\
& \quad + \langle \phi_d(\mathbf{x}_1) | \psi_a(\mathbf{x}_1) \rangle \langle \phi_e(\mathbf{x}_2) | \psi_b(\mathbf{x}_2) \rangle \cdots \langle \phi_f(\mathbf{x}_N) | \hat{h}_N | \psi_c(\mathbf{x}_N) \rangle \\
& \quad \left. + \langle \phi_d(\mathbf{x}_1) \phi_e(\mathbf{x}_2) \left| \frac{1}{r_{12}} \right| \psi_a(\mathbf{x}_1) \psi_b(\mathbf{x}_2) \right\rangle \cdots \langle \phi_f(\mathbf{x}_N) | \psi_c(\mathbf{x}_N) \rangle + \cdots \right) \quad (5.12)
\end{aligned}$$

Αν στην (5.12) αντικαταστήσουμε τα spin-τροχιακά με το γινόμενο του χωρικού τροχιακού επί τη συνάρτηση spin, τότε θα διαχωριστούν τα ολοκληρώματα ως προς τις χωρικές μεταβλητές από τα ολοκληρώματα ως προς τις spin, εφόσον είναι ανεξάρτητες μεταξύ τους. Δεδομένου ότι η ορθοκανονικότητα των spin συναρτήσεων επιβάλλει

$$\int s_i^*(\omega) s_j(\omega) d\omega = \langle s_i | s_j \rangle = \delta_{ij}, \quad \text{όπου } s_{i,j} = \alpha \text{ ή } \beta, \quad (5.13)$$

στην (5.12) θα απομείνουν μόνο τα ολοκληρώματα των χωρικών τροχιακών, τα οποία δεν μηδενίζονται εξ' αιτίας του spin. Φυσικά, αυτό είναι πολύ χρήσιμο, αφού δεν χρειάζεται να προχωρήσουμε στον υπολογισμό όλων των παραπάνω ολοκληρωμάτων (και ειδικά των διπλεκτρονιακών, που είναι σχετικά χρονοβόρος) παρά μόνο σ' αυτά που

θα “επιβιώσουν”⁴. Άρα η (5.12) γίνεται :

$$\begin{aligned}
\langle D_{pi} | \hat{H} | D_{qj} \rangle &= \frac{1}{N!} \sum_{a=1}^N \sum_{b=1}^N \cdots \sum_{c=1}^N \sum_{d=1}^N \sum_{e=1}^N \cdots \sum_{f=1}^N \varepsilon_{ab\cdots c} \varepsilon_{de\cdots f} \\
&\times \left(\langle u_d(\mathbf{r}_1) | \hat{h}_1 | v_a(\mathbf{r}_1) \rangle \langle u_e(\mathbf{r}_2) | v_b(\mathbf{r}_2) \rangle \cdots \langle u_f(\mathbf{r}_N) | v_c(\mathbf{r}_N) \rangle \right. \\
&+ \langle u_d(\mathbf{r}_1) | v_a(\mathbf{r}_1) \rangle \langle u_e(\mathbf{r}_2) | \hat{h}_2 | v_b(\mathbf{r}_2) \rangle \cdots \langle u_f(\mathbf{r}_N) | v_c(\mathbf{r}_N) \rangle + \cdots \\
&+ \langle u_d(\mathbf{r}_1) | v_a(\mathbf{r}_1) \rangle \langle u_e(\mathbf{r}_2) | v_b(\mathbf{r}_2) \rangle \cdots \langle u_f(\mathbf{r}_N) | \hat{h}_N | v_c(\mathbf{r}_N) \rangle \\
&\left. + \langle u_d(\mathbf{r}_1) u_e(\mathbf{r}_2) | \frac{1}{r_{12}} | v_a(\mathbf{r}_1) v_b(\mathbf{r}_2) \rangle \cdots \langle u_f(\mathbf{r}_N) | v_c(\mathbf{r}_N) \rangle + \cdots \right) \quad (5.14)
\end{aligned}$$

Ακολουθούμε την ίδια διαδικασία και για την επικάλυψη.

5.3 Επικάλυψη

Ομοίως με το τμήμα 5.2 βρίσκουμε και τα μητροστοιχεία επικάλυψης:

$$\begin{aligned}
\langle D_{pi} | D_{qj} \rangle &= \\
&= \frac{1}{N!} \left\langle \sum_{d=1}^N \sum_{e=1}^N \cdots \sum_{f=1}^N \varepsilon_{de\cdots f} \phi_d(\mathbf{x}_1) \phi_e(\mathbf{x}_2) \cdots \phi_f(\mathbf{x}_N) \right| \\
&\times \sum_{a=1}^N \sum_{b=1}^N \cdots \sum_{c=1}^N \varepsilon_{ab\cdots c} \psi_a(\mathbf{x}_1) \psi_b(\mathbf{x}_2) \cdots \psi_c(\mathbf{x}_N) \left. \right\rangle \\
&= \frac{1}{N!} \sum_{a=1}^N \sum_{b=1}^N \cdots \sum_{c=1}^N \sum_{d=1}^N \sum_{e=1}^N \cdots \sum_{f=1}^N \varepsilon_{ab\cdots c} \varepsilon_{de\cdots f} \\
&\times \left(\langle u_d(\mathbf{r}_1) | v_a(\mathbf{r}_1) \rangle \langle u_e(\mathbf{r}_2) | v_b(\mathbf{r}_2) \rangle \cdots \langle u_f(\mathbf{r}_N) | v_c(\mathbf{r}_N) \rangle \right) \quad (5.15)
\end{aligned}$$

Όπως και στην (5.14), αντικαταστήσαμε τα spin-τροχιακά με τα χωρικά, χρησιμοποιώντας τους επιπλέον βαθμούς ελευθερίας των συναρτίσεων spin καθώς και την ορθοκανονικότητα τους (5.13).

⁴ Το ίδιο ισχύει και για τα σύμβολα Levi-Civita, τα οποία επιλέγουν μερικά μόνο από τα ολοκληρώματα, πρώτου προβούμε στον υπολογισμό τους.

5.4 Υπολογισμός Ολοκληρωμάτων

Είδαμε πώς καταλήγουμε από τα μητροστοιχεία μεταξύ οριζουσών Slater σε βασικά ολοκληρώματα: τώρα θα αναλύσουμε περαιτέρω τα επιμέρους ολοκληρώματα, ώστε να διευκολύνουμε τον τελικό υπολογισμό καθώς και το προγραμματισμό τους.

Στα παρακάτω, θα εκφράσουμε τα spin-τροχιακά και την Χαμιλτονιανή σε επιμήκεις σφαιροειδείς συντεταγμένες, με διάνυσμα θέσεως του ηλεκτρονίου $\mathbf{r} = (\xi, \eta, \phi)$ (παρ. Α', σελ. 83), εφόσον η γεωμετρία του μορίου παραπέμπει σε επιμήκεις σφαιροειδείς, ενώ η εξίσωση *Schrödinger* είναι διαχωρίσιμη στο συγκεκριμένο ορθογώνιο σύστημα συντεταγμένων (παρ. Α'.2, σελ. 87).

5.4.1 OEDO

Τα χωρικά τροχιακά, είναι λύσεις της εξίσωσης *Schrödinger* σε επιμήκεις σφαιροειδείς συντεταγμένες (παρ. Α'.2.1, σελ. 89) για διατομικά μόρια ενός ηλεκτρονίου με πυρήνες Z_a, Z_b ⁵ και κατά συνέπεια ονομάζονται *τροχιακά μονοηλεκτρονιακού διατομικού μορίου (OEDO [3])*. Αυτά εκφράζονται ως

$$u_i(\mathbf{r}) = \Xi_i(\xi)H_i(\eta)e^{\pm im\phi}, \quad m = 0, 1, 2, \dots \quad (5.16)$$

όπου Ξ_i : η ακτινική συνάρτηση γενικευμένου σφαιροειδούς (παρ. Β'.2, σελ. 104)

$$\Xi(\xi) = (\xi^2 - 1)^{\frac{m}{2}} e^{-p\xi} (\xi + 1)^{\frac{R(Z_a + Z_b)}{2p} - (m+1)} \sum_{j=0}^{\infty} g_j \left(\frac{\xi - 1}{\xi + 1} \right)^j \quad (5.17)$$

και H_i : η γωνιακή συνάρτηση γενικευμένου σφαιροειδούς (παρ. Β'.3, σελ. 104), η οποία έχει τρεις διαφορετικές εκφράσεις [3] ανάλογα με το είδος του μορίου:

⁵ Τα οποία χρησιμοποιούνται ως μή γραμμικές παράμετροι μεταβολών

I. για ομοπυρηνικό διατομικό μόριο :

$$H(\eta) = \sum_{j=0}^{\infty} f_j \mathcal{P}_{m+j}^m(\eta) \quad (5.18)$$

(κεφ. 3.2.2, σελ. 25)

II. για εταιροπυρηνικό διατομικό μόριο με μικρό R :

$$H(\eta) = e^{\pm p\eta} \sum_{j=0}^{\infty} f_j \mathcal{P}_{m+j}^m(\pm\eta) \quad (5.19)$$

III. για εταιροπυρηνικό διατομικό μόριο με μεγάλο R :

$$H(\eta) = (1 - \eta^2)^{\frac{m}{2}} e^{\pm p\eta} \sum_{j=0}^{\infty} f_j (1 \mp \eta)^j \quad (5.20)$$

Στα παραπάνω, το όριο του μεγάλου R πρακτικά σημαίνει $R > R_{jump}$, όπου R_{jump} παράμετρος που δίνεται από το πρόγραμμα του Power OEDM [7]. Τα \pm καθορίζονται στο πρόγραμμα από την παράμετρο $icen$, που επίσης παράγεται από το OEDM και σχετίζεται με το φορτίο των δύο πυρήνων και κατ' επέκταση σε ποίο πυρήνα βρίσκεται πλεισιέστερα το ηλεκτρόνιο (κατα μέσο όρο).

Τα $\Xi(\xi)$ και $H(\eta)$ είναι λύσεις των γενικευμένων εξισώσεων σφαιροειδούς που προκύπτουν από την αντικατάσταση της (5.16) στην εξίσωση του Schrödinger (παρ. Α.2.1, σελ. 89) :

$$\begin{aligned} & \frac{d}{d\xi} \left[(\xi^2 - 1) \frac{d\Xi(\xi)}{d\xi} \right] \\ & + \left[-c + R(z_a^* + z_b^*)\xi - p^2(\xi^2 - 1) - \frac{m^2}{\xi^2 - 1} \right] \Xi(\xi) = 0 \end{aligned} \quad (5.21\alpha')$$

$$\begin{aligned} & \frac{d}{d\eta} \left[(1 - \eta^2) \frac{dH(\eta)}{d\eta} \right] \\ & + \left[c - R(z_a^* - z_b^*)\eta - p^2(1 - \eta^2) - \frac{m^2}{1 - \eta^2} \right] H(\eta) = 0 \end{aligned} \quad (5.21\beta')$$

$$\frac{d^2\Phi(\phi)}{d\phi^2} + m^2\Phi(\phi) = 0 \quad (5.21\gamma')$$

όπου c, p^2 και m είναι οι σταθερές διαχωρισμού [2] που σχετίζονται με την ενέργεια ως

$$E = -\frac{2p^2}{R^2} + \frac{Z_a Z_b}{R} \quad (5.22)$$

Αντικατάσταση των $\Xi(\xi)$ και $H(\eta)$ στις εξισώσεις (5.21α') και (5.21β') αντίστοιχα, οδηγούν σε αναδρομικές σχέσεις τριών όρων της μορφής[3]:

$$\alpha_j g_{j-1} + \beta_j g_j + \gamma_j g_{j+1} = 0, \quad j = 0, 1, 2, \dots \quad (5.23\alpha')$$

$$g_{-1} = 0 \quad \text{και} \quad (5.23\beta')$$

$$\alpha_j f_{j-1} + \beta_j f_j + \gamma_j f_{j+1} = 0, \quad j = 0, 1, 2, \dots \quad (5.24\alpha')$$

$$f_{-1} = 0 \quad (5.24\beta')$$

όπου τα α_j, β_j και γ_j είναι συναρτήσεις των R, Z_a, Z_b, p, c και m . Τα παραπάνω συστήματα αναδρομικών σχέσεων λύνονται από τον Power με τη μέθοδο των συνεχόμενων κλασμάτων [16], το πρόγραμμα του οποίου έχουμε συμπεριλάβει στο παρόν πρόγραμμα, καθώς μας τροφοδοτεί με τις παραμέτρους $c, p, \{f_j\}, \{g_j\}$ για κάθε τροχιακό.

Τέλος, το στοιχείο όγκου γράφεται ως

$$d^3r = \frac{R^3}{8} (\xi^2 - \eta^2) d\xi d\eta d\phi, \quad (5.25)$$

$$\text{με } \xi = \frac{r_a + r_b}{R}, \quad \eta = \frac{r_a - r_b}{R} \quad (5.26)$$

$$\text{και } 1 \leq \xi \leq \infty, \quad -1 \leq \eta \leq 1 \quad (5.27)$$

όπου r_i : η απόσταση του ηλεκτρονίου από το πυρήνα i , $R \equiv \frac{1}{|\mathbf{R}_a - \mathbf{R}_b|}$ η διαπυρηνική απόσταση, \mathbf{R}_i : διάνυσμα θέσεως του πυρήνα i και ϕ : το αζιμούθιο με $0 \leq \phi \leq 2\pi$.

Βασιζόμενοι στα παραπάνω, είμαστε πλέον σε θέση να αναλύσουμε τα παραγόμενα ολοκληρώματα.

5.4.2 Ολοκληρώματα Επικαλύψεως

Αρχίζουμε από την πιο απλή περίπτωση, των ολοκληρωμάτων επικαλύψεως, που προκύπτουν από τις (5.14) και (5.15).

$$\begin{aligned}
& \langle u_l(\mathbf{r}_i) | v_k(\mathbf{r}_i) \rangle \stackrel{(5.16)}{=} \int_{-\infty}^{\infty} u_l^*(\mathbf{r}_i) v_m(\mathbf{r}_i) d^3 r_i \\
&= \iiint \Xi_l^*(\xi_i) H_l^*(\eta_i) e^{-im_l \phi_i} \Xi_k(\xi_i) H_k(\eta_i) e^{im_k \phi_i} \frac{R^3}{8} (\xi_i^2 - \eta_i^2) d\xi_i d\eta_i d\phi_i \\
&= \frac{R^3}{8} \iint \Xi_l^*(\xi_i) H_l^*(\eta_i) \Xi_k(\xi_i) H_k(\eta_i) (\xi_i^2 - \eta_i^2) d\xi_i d\eta_i \underbrace{\int_0^{2\pi} e^{i(m_k - m_l)\phi_i} d\phi_i}_{2\pi \delta_{lk}} \\
&= \frac{\pi R^3}{4} \delta_{lk} \left(\int_1^{\infty} \xi_i^2 \Xi_l^*(\xi_i) \Xi_k(\xi_i) d\xi_i \int_{-1}^1 H_l^*(\eta_i) H_k(\eta_i) d\eta_i \right. \\
&\quad \left. - \int_1^{\infty} \Xi_l^*(\xi_i) \Xi_k(\xi_i) d\xi_i \int_{-1}^1 \eta_i^2 H_l^*(\eta_i) H_k(\eta_i) d\eta_i \right) \quad (5.28)
\end{aligned}$$

5.4.3 Μονοπλεκτρονιακά Ολοκληρώματα

Για τα μονοπλεκτρονιακά ολοκληρώματα, αρχικά ορίζουμε τις διαφορές παραμέτρους που θα χρειαστούν: $z = Z_a + Z_b$, $\zeta = Z_a - Z_b$, $z^* = z_a^* + z_b^*$ και $\zeta^* = z_a^* - z_b^*$ όπου Z_l : οι πραγματικοί πυρήνες, ενώ z_i^* : το ενεργό πυρηνικό φορτίο του πυρήνα i , που χρησιμοποιούμε στο πρόγραμμα ως μη γραμμική παράμετρο βελτιστοποίησης [2]. Εφόσον εμπλέκεται η μονοπλεκτρονιακή Χαμιλτονιανή θα χρειαστούμε την έκφρασή της σε επιμήκεις σφαιροειδείς συντεταγμένες: η Λαπλασιανή γράφεται (παρ. Α'2, σελ. 87):

$$\nabla^2 = \frac{4}{R^2(\xi^2 - \eta^2)} \left\{ \frac{\partial}{\partial \xi} \left[(\xi^2 - 1) \frac{\partial}{\partial \xi} \right] + \frac{\partial}{\partial \eta} \left[(1 - \eta^2) \frac{\partial}{\partial \eta} \right] + \frac{(\xi^2 - \eta^2)}{(\xi^2 - 1)(1 - \eta^2)} \frac{\partial^2}{\partial \phi^2} \right\} \quad (5.29)$$

Βασιζόμενοι στα παραπάνω έχουμε :

$$\begin{aligned}
& \langle u_l(\mathbf{r}_i) | \hat{h}_i | v_k(\mathbf{r}_i) \rangle \stackrel{(5.16)}{=} \int_{-\infty}^{\infty} u_l^*(\mathbf{r}_i) \hat{h}_i v_m(\mathbf{r}_i) d^3 r_i \\
& \stackrel{(5.4)}{=} \int_{-\infty}^{\infty} u_l^*(\mathbf{r}_i) \left(-\frac{1}{2} \nabla_i^2 - \frac{Z_a}{r_{ai}} - \frac{Z_b}{r_{bi}} \right) v_m(\mathbf{r}_i) d^3 r_i \\
& = \underbrace{-\frac{1}{2} \int_{-\infty}^{\infty} u_l^*(\mathbf{r}_i) \nabla_i^2 v_m(\mathbf{r}_i) d^3 r_i}_A - \underbrace{\int_{-\infty}^{\infty} u_l^*(\mathbf{r}_i) \left(\frac{Z_a}{r_{ai}} + \frac{Z_b}{r_{bi}} \right) v_m(\mathbf{r}_i) d^3 r_i}_B \quad (5.30)
\end{aligned}$$

Στα επόμενα, για λόγους απλότητας, παραλείπουμε των δείκτη i των ηλεκτρονίων. Το πρώτο σκέλος της (5.30) μέσω των (5.29), (5.16) και (5.25) γίνεται :

$$\begin{aligned}
A & \stackrel{(5.16)}{=} \frac{1}{2} \iiint \Xi_l^*(\xi) H_l^*(\eta) e^{-im_l \phi} \frac{4}{R^2(\xi^2 - \eta^2)} \left\{ \frac{\partial}{\partial \xi} \left[(\xi^2 - 1) \frac{\partial}{\partial \xi} \right] \right. \\
& \quad \left. + \frac{\partial}{\partial \eta} \left[(1 - \eta^2) \frac{\partial}{\partial \eta} \right] + \frac{(\xi^2 - \eta^2)}{(\xi^2 - 1)(1 - \eta^2)} \frac{\partial^2}{\partial \phi^2} \right\} \Xi_k(\xi) H_k(\eta) e^{im_k \phi} \frac{R^3}{8} (\xi^2 - \eta^2) d\xi d\eta d\phi \\
& = \frac{R}{4} \left\{ \int_1^{\infty} \Xi_l^*(\xi) \frac{d}{d\xi} \left[(\xi^2 - 1) \frac{d\Xi_k(\xi)}{d\xi} \right] d\xi \int_{-1}^1 H_l^*(\eta) H_k(\eta) \underbrace{\int_0^{2\pi} e^{i(m_k - m_l)\phi} d\phi}_{2\pi\delta_{lk}} \right. \\
& \quad \left. + \int_1^{\infty} \Xi_l^*(\xi) \Xi_k(\xi) d\xi \int_{-1}^1 H_l^*(\eta) \frac{d}{d\eta} \left[(1 - \eta^2) \frac{dH_k(\eta)}{d\eta} \right] d\eta \underbrace{\int_0^{2\pi} e^{i(m_k - m_l)\phi} d\phi}_{2\pi\delta_{lk}} \right. \\
& \quad \left. + \iint \Xi_l^*(\xi) H_l^*(\eta) \frac{(\xi^2 - \eta^2)}{(\xi^2 - 1)(1 - \eta^2)} \Xi_k(\xi) H_k(\eta) d\xi d\eta \int_0^{2\pi} e^{-im_l \phi} \frac{d^2(e^{im_k \phi})}{d\phi^2} d\phi \right\} \quad (5.31)
\end{aligned}$$

Παρατηρούμε πώς στο πρώτο και το δεύτερο ολοκλήρωμα του πρώτου και του δεύτερου σκέλους της (5.31), αντίστοιχα, ο διαφορικός τελεστής έχει την ίδια ακριβώς μορφή με αυτόν του πρώτου σκέλους των (5.21α') και (5.21β')· ξαναγράφουμε τις τελευταίες ως

$$\begin{aligned} & \frac{d}{d\xi} \left[(\xi^2 - 1) \frac{d\xi_k(\xi)}{d\xi} \right] \\ &= \left[c_k - R z_k^* \xi + p_k^2 (\xi^2 - 1) + \frac{m_k^2}{\xi^2 - 1} \right] \xi_k(\xi) \end{aligned} \quad (5.32\alpha')$$

$$\begin{aligned} & \frac{d}{d\eta} \left[(1 - \eta^2) \frac{dH_k(\eta)}{d\eta} \right] \\ &= \left[-c_k + R \zeta_k^* \eta + p_k^2 (1 - \eta^2) + \frac{m_k^2}{1 - \eta^2} \right] H_k(\eta) \end{aligned} \quad (5.32\beta')$$

και αντικαθιστούμε στην (5.31) τις παραπάνω ισότητες :

$$\begin{aligned} A &= \frac{\pi R}{2} \delta_{lk} \left\{ \int_1^\infty \Xi_l^*(\xi) \left[c_k - R z_k^* \xi + p_k^2 (\xi^2 - 1) + \frac{m_k^2}{\xi^2 - 1} \right] d\xi \int_{-1}^1 H_l^*(\eta) H_k(\eta) \right. \\ &+ \int_1^\infty \Xi_l^*(\xi) \Xi_k(\xi) d\xi \int_{-1}^1 H_l^*(\eta) \left[-c_k + R \zeta_k^* \eta + p_k^2 (1 - \eta^2) + \frac{m_k^2}{1 - \eta^2} \right] d\eta \\ &+ \left. \iint \Xi_l^*(\xi) H_l^*(\eta) \left(\frac{-m_k^2}{\xi^2 - 1} + \frac{-m_k^2}{1 - \eta^2} \right) \Xi_k(\xi) H_k(\eta) d\xi d\eta \right\} \\ &= \frac{\pi R}{2} \delta_{lk} \left\{ \iint \Xi_l^*(\xi) \Xi_k(\xi) H_l^*(\eta) H_k(\eta) \left[p_k^2 (\xi^2 - \eta^2) - R z_k^* \xi + R \zeta_k^* \eta \right] d\xi d\eta \right\} \end{aligned} \quad (5.33)$$

Εν συνεχεία, θα πρέπει να εκφράσουμε το παράγοντα που εμφανίζεται στο δεύτερο σκέλος της (5.30) ως προς ξ και η , μέσω των (5.16) :

$$(5.16) \rightarrow \begin{cases} \xi = \frac{|\mathbf{r}_i - \mathbf{R}_a| + |\mathbf{r}_i - \mathbf{R}_b|}{|\mathbf{R}_a - \mathbf{R}_b|}, \\ \eta = \frac{|\mathbf{r}_i - \mathbf{R}_a| - |\mathbf{r}_i - \mathbf{R}_b|}{|\mathbf{R}_a - \mathbf{R}_b|}. \end{cases} \quad (5.34)$$

όπου $\mathbf{r}_i \equiv \mathbf{r}$, $\mathbf{R} \equiv |\mathbf{R}_a - \mathbf{R}_b|$ και $\mathbf{r}_{xi} \equiv \mathbf{r}_x \equiv |\mathbf{r}_i - \mathbf{R}_x|$, οπότε

$$\Rightarrow \begin{cases} \frac{1}{r_a} = \frac{2}{(\xi + \eta)R}, \\ \frac{1}{r_b} = \frac{2}{(\xi - \eta)R}. \end{cases} \quad (5.35)$$

Αντικαθιστώντας τις (5.35), το B σκέλος παίρνει τη μορφή :

$$\begin{aligned}
B &\stackrel{(5.16)}{=} \iiint \Xi_l^*(\xi) H_l^*(\eta) e^{-im_l\phi} \left[\frac{2Z_a}{(\xi + \eta)R} + \frac{2Z_b}{(\xi - \eta)R} \right] \Xi_k(\xi) H_k(\eta) e^{im_k\phi} \frac{R^3}{8} (\xi^2 - \eta^2) d\xi d\eta d\phi \\
&= \frac{R^2}{4} \iint \Xi_l^*(\xi) H_l^*(\eta) \Xi_k(\xi) H_k(\eta) [(Z_a + Z_b)\xi - (Z_a - Z_b)\eta] d\xi d\eta \underbrace{\int_0^{2\pi} e^{i(m_k - m_l)\phi} d\phi}_{2\pi\delta_{lk}} \\
&= \frac{\pi R}{2} \delta_{lk} \iint \Xi_l^*(\xi) H_l^*(\eta) \Xi_k(\xi) H_k(\eta) [R(z\xi - \zeta\eta)] d\xi d\eta \tag{5.36}
\end{aligned}$$

Τελικώς η (5.30), μέσω των (5.33) και (5.36) παίρνει τη μορφή :

$$\begin{aligned}
\langle u_l(\mathbf{r}_i) | \hat{h}_i | v_k(\mathbf{r}_i) \rangle = \\
- \frac{\pi R}{2} \delta_{lk} \iint \Xi_l^*(\xi) H_l^*(\eta) \Xi_k(\xi) H_k(\eta) [p_k^2(\xi^2 - \eta^2) + R(z - z_k^*)\xi - R(\zeta - \zeta_k^*)\eta] d\xi d\eta \tag{5.37}
\end{aligned}$$

Προς απλοποίηση και συντόμευση των υπολογισμών, ερευνούμε περαιτέρω τα γωνιακά ολοκληρώματα :

Γωνιακά ολοκληρώματα

Απο τις (5.28) και (5.37) φαίνεται πως καταλήγουμε σε τρία είδη γωνιακών ολοκληρωμάτων για την ομοκυρηνική περίπτωση έχουμε

$$\begin{aligned}
&\int_{-1}^1 d\eta H_l^*(\eta) \Xi_k(\xi) H_k(\eta) \eta^b, \quad \{b \in \mathbb{N}^0 : [0, 2]\} \xrightarrow{(5.18)} \\
&\sum_{i=0}^{\infty} \sum_{k=0}^{\infty} f_i^* f_k \underbrace{\int_{-1}^1 d\eta \mathcal{P}_{m+i}^{m*}(\eta) \mathcal{P}_{m'+k}^{m'}(\eta) \eta^b}_{\mathcal{A}_{\eta^b}} \tag{5.38}
\end{aligned}$$

Λαμβάνοντας υπ' όψιν την ομοτιμία $(-1)^{l+m}$ από την (5.51), των \mathcal{P}_l^m έχουμε

$$\mathcal{A}_{\eta^b} = \begin{cases} 0, & \text{εάν } 2m + i + 2m' + k + b = 2n \quad , \\ a, & \text{εάν } 2m + i + 2m' + k + b = 2n + 1 \quad . \end{cases} \tag{5.39}$$

όπου $n \in \mathbb{N}^0$ και $a \in \mathbb{R}$.

Καταλήξαμε έτσι, σε μια σύμβαση για τα μονοπλεκτρονιακά και επικαλύψεως ολοκληρώματα, που απλοποιεί τον υπολογισμό, εφόσον μας ενημερώνει αν πρέπει να προβούμε στον υπολογισμό των ακτινικών ολοκληρωμάτων, δεδομένου ότι αυτά πολλαπλασιάζονται με τα γωνιακά.

Θα δούμε παρακάτω, ότι υπάρχει αντίστοιχη σύμβαση και για τα διπλεκτρονιακά ολοκληρώματα (στην ομοσφαιρική περίπτωση), η οποία μάλιστα έχει μεγάλη επίπτωση στο συνολικό χρόνο του υπολογισμού.

5.4.4 Διπλεκτρονιακά Ολοκληρώματα

Τέλος, αναλύουμε τα διπλεκτρονιακά ολοκληρώματα τα οποία εμφανίζουν και το μεγαλύτερο βαθμό πολυπλοκότητας. Σε αυτό το σημείο, θα πρέπει να εκφράσουμε των τελεστή δύο ηλεκτρονίων βάσει του ορθογώνιου συστήματος συντεταγμένων που έχουμε επιλέξει. Έτσι, υιοθετούμε την ανάπτυξη κατά *Neumann* του $\frac{1}{|\mathbf{r} - \mathbf{r}'|}$ σε επιμήκεις σφαιροειδείς συντεταγμένες [17]:

$$\frac{1}{r_{12}} = \frac{8\pi}{R} \sum_{\lambda=0}^{\infty} \sum_{\mu=-\lambda}^{\lambda} (-1)^{\mu} \frac{(\lambda - \mu)!}{(\lambda + \mu)!} \mathcal{P}_{\lambda}^{\mu}(\xi_{<}) \mathcal{Q}_{\lambda}^{\mu}(\xi_{>}) Y_{\lambda}^{\mu}(\eta_1, \phi_1) Y_{\lambda}^{\mu*}(\eta_2, \phi_2) \quad (5.40)$$

όπου $\mathcal{P}_{\lambda}^{\mu}$ και $\mathcal{Q}_{\lambda}^{\mu}$ οι συναφείς συναρτήσεις *Legendre* πρώτου και δευτέρου είδους, αντίστοιχα, Y_{λ}^{μ} : οι σφαιρικές αρμονικές, ενώ $\xi_{<} = \min(\xi_1, \xi_2)$ και $\xi_{>} = \max(\xi_1, \xi_2)$. Οι $f_l^m : \mathcal{P}_l^m, \mathcal{Q}_l^m$ ορίζονται ως

$$f_l^m(x) = (-1)^m |1 - x^2|^{\frac{m}{2}} \frac{d^m f_l(x)}{dx^m} \quad (5.41)$$

ενώ τα $-m$ σχετίζονται με τα m :

$$f_l^{-m}(x) = (-1)^m \frac{(l - m)!}{(l + m)!} f_l^m(x), \quad m \in \mathbb{Z} \quad (5.42)$$

όπου $\mathcal{P}_l(x)$ οι (απλές) συναρτήσεις Legendre πρώτου είδους, που ορίζονται βάσει του τύπου *Rodrigues* :

$$\mathcal{P}_l(x) := \frac{1}{2^l l!} \left(\frac{d}{dx} \right)^l (x^2 - 1)^l. \quad (5.43)$$

Επομένως η (5.41) μέσω της (5.43) γίνεται :

$$\mathcal{P}_l^m(x) := \frac{(-1)^m}{2^l l!} (1 - x^2)^{\frac{m}{2}} \frac{d^{l+m}}{dx^{l+m}} (x^2 - 1)^l \quad (5.44)$$

Εισάγουμε τώρα, τις κανονικοποιημένες συναφείς συναρτήσεις Legendre πρώτου είδους

$$P_l^m(x) = C_l^m \mathcal{P}_l^m(x) \quad (5.45)$$

οπότε κανονικοποιώντας

$$\begin{aligned} \int_{-1}^1 P_l^m(t) P_k^m(t) dt &= \delta_{lk} \xrightarrow{l=k} \\ |C_l^m|^2 \int_{-1}^1 |\mathcal{P}_l^m(t)|^2 dt &= 1 \xrightarrow{(?)} \\ |C_l^m|^2 \frac{2}{2l+1} \frac{(l+m)!}{(l-m)!} &= 1 \\ C_l^m &= \sqrt{\frac{2l+1}{2} \frac{(l-m)!}{(l+m)!}} \end{aligned} \quad (5.46)$$

Άρα, τελικά έχουμε

$$P_l^m(x) := \sqrt{\frac{2l+1}{2} \frac{(l-m)!}{(l+m)!}} \mathcal{P}_l^m(x) \quad (5.47)$$

Οι (απλές) συναρτήσεις Legendre δευτέρου είδους, ορίζονται βάσει του τύπου :

$$Q_l(x) := \frac{1}{2} \log \left(\frac{x+1}{x-1} \right) \mathcal{P}_l(x) - \sum_{m=1}^l \frac{1}{m} \mathcal{P}_{m-1}(x) \mathcal{P}_{l-m}(x) \quad (5.48)$$

Η ομοτιμία (parity) των \mathcal{P}_l φαίνεται από τη σχέση

$$\mathcal{P}_l(-x) = (-1)^l \mathcal{P}_l(x), \quad (5.49)$$

οπότε

$$\mathcal{P}_l(x) = \begin{cases} \text{άρτιο,} & \text{εάν } l = 2n, n \in \mathbb{Z} \\ \text{περιττό,} & \text{εάν } l = 2n + 1, n \in \mathbb{Z} \end{cases}, \quad (5.50)$$

Μπορούμε να πάρουμε τα \mathcal{P}_l^m παραγωγίζοντας τα \mathcal{P}_l m φορές, όπου βέβαια κάθε παραγωγή αλλάζει και το πρόσημο, και κατόπιν, βάσει της (5.41), πολλαπλασιάζουμε με το παράγοντα $(x^2-1)^{\frac{m}{2}}$, που έχει άρτια ομοτιμία. Επομένως η ομοτιμία των \mathcal{P}_l^m θα εξαρτάται από το $l+m$ [10],

$$\mathcal{P}_l^m(-x) = (-1)^{l+m} \mathcal{P}_l^m(x) \quad (5.51)$$

Αντίθετα, για τα συναφή πολυώνυμα Legendre δευτέρου είδους η ομοτιμία εξαρτάται από το $l+m+1$:

$$\mathcal{Q}_l^m(-x) = (-1)^{l+m+1} \mathcal{Q}_l^m(x) \quad (5.52)$$

Οι συναφείς συναρτήσεις Legendre δευτέρου είδους $\mathcal{Q}_l^m : \mathbb{C} \setminus (-\infty, 1] \rightarrow \mathbb{C}$, $l+m \notin -\mathbb{N}$ εκφράζονται, για $|x| > 1$, συναρτήσει των υπεργεωμετρικών [11] συναρτήσεων ${}_2F_1$ ως:

$$\begin{aligned} \mathcal{Q}_l^m(x) &:= e^{im\pi} 2^{-l-1} \sqrt{\pi} \frac{\Gamma(l+m+1)}{\Gamma(l+\frac{3}{2})} x^{-l-m-1} (x^2-1)^{\frac{m}{2}} \\ &\times {}_2F_1\left(1 + \frac{l}{2} + \frac{m}{2}, \frac{1}{2} + \frac{l}{2} + \frac{m}{2}; l + \frac{3}{2}; \frac{1}{x^2}\right) \end{aligned} \quad (5.53)$$

όπου $\Gamma(n) = (n-1)!$: οι συναρτήσεις γάμμα, ενώ η υπεργεωμετρική⁶ συνάρτηση, ${}_2F_1 : \mathbb{C}^2 \times (\mathbb{C} \setminus -\mathbb{N}_0)^1 \times \{z \in \mathbb{C} : |z| < 1\} \rightarrow \mathbb{C}$, ορίζεται ως δυναμοσειρά :

$${}_2F_1(a, b; c; z) = \sum_{n=0}^{\infty} \frac{(a)_n (b)_n}{(c)_n} \frac{z^n}{n!} \quad (5.54)$$

⁶ Η γενικευμένη υπεργεωμετρική συνάρτηση ${}_pF_q : \mathbb{C}^p \times (\mathbb{C} \setminus -\mathbb{N}_0)^q \times \{z \in \mathbb{C} : |z| < 1\} \rightarrow \mathbb{C}$ [27] ορίζεται ως

$${}_pF_q(a_1, \dots, a_p; b_1, \dots, b_q; z) := \sum_{n=0}^{\infty} \frac{\prod_{k=1}^p (a_k)_n}{\prod_{k=1}^q (b_k)_n} \frac{z^n}{n!}$$

όπου $(q)_n : \mathbb{C} \rightarrow \mathbb{C} \ \forall n \in \mathbb{N}_0$: το σύμβολο *Pochhammer* ή αυξάνον παραγοντικό, το οποίο ορίζεται ακολούθως

$$(q)_n := \prod_{j=1}^n (q + j - 1), \quad (5.55)$$

ενώ ισχύει και $(q)_n = q \cdot (q - 1) \cdots (q + n - 1) = \frac{\Gamma(q + n)}{\Gamma(q)}$.

Οι σφαιρικές αρμονικές ορίζονται ως

$$Y_l^m(\theta, \phi) := \sqrt{\frac{2l + 1}{4\pi} \frac{(l - m)!}{(l + m)!}} \mathcal{P}_l^m(\cos \theta) e^{im\phi} \quad (5.56)$$

και η συνθήκη ορθοκανονικότητας είναι

$$\int_0^{2\pi} d\phi \int_0^\pi \sin \theta d\theta Y_l^{m*}(\theta, \phi) Y_{l'}^{m'}(\theta, \phi) = \delta_{ll'} \delta_{mm'}. \quad (5.57)$$

Στη πράξη, η (5.40) είναι κάπως δύσχρηστη, οπότε την ξαναγράφουμε στην παρακάτω μορφή [28] :

$$\begin{aligned} \frac{1}{|\mathbf{r} - \mathbf{r}'|} &= \frac{2}{R} \sum_{\lambda=0}^{\infty} \sum_{\mu=0}^{\lambda} (-1)^\mu \alpha_\mu (2\lambda + 1) \left[\frac{(\lambda - \mu)!}{(\lambda + \mu)!} \right]^2 \\ &\times \mathcal{P}_\lambda^\mu(\xi_{<}) \mathcal{Q}_\lambda^\mu(\xi_{>}) \mathcal{P}_\lambda^\mu(\eta_1) \mathcal{P}_\lambda^{\mu*}(\eta_2) \cos[\mu(\phi_1 - \phi_2)] \end{aligned} \quad (5.58)$$

όπου $\alpha_\mu = 2 - \delta_{\mu,0}$. Κατόπιν αποδεικνύουμε την ισοδυναμία της (5.58) με την (5.40)

Απόδειξη. Αρχικά αντικαθιστούμε την (5.56) στην (5.40)

$$\frac{1}{r_{12}} = \frac{2}{R} \sum_{\lambda=0}^{\infty} \sum_{\mu=-\lambda}^{\lambda} (-1)^\mu (2\lambda + 1) \left[\frac{(\lambda - \mu)!}{(\lambda + \mu)!} \right]^2 \mathcal{P}_\lambda^\mu(\xi_{<}) \mathcal{Q}_\lambda^\mu(\xi_{>}) \mathcal{P}_\lambda^\mu(\eta_1) \mathcal{P}_\lambda^{\mu*}(\eta_2) e^{i\mu(\phi_1 - \phi_2)} \quad (5.59)$$

Θέλουμε να απαλλαγούμε από την άθροιση πάνω στα αρνητικά μ , οπότε χωρίζουμε το άθροισμα

$$\begin{aligned} \frac{1}{r_{12}} &= \frac{2}{R} \sum_{\lambda=0}^{\infty} \left(\sum_{\mu=-\lambda}^{-1} (-1)^\mu (2\lambda + 1) \left[\frac{(\lambda - \mu)!}{(\lambda + \mu)!} \right]^2 \mathcal{P}_\lambda^\mu(\xi_{<}) \mathcal{Q}_\lambda^\mu(\xi_{>}) \mathcal{P}_\lambda^\mu(\eta_1) \mathcal{P}_\lambda^{\mu*}(\eta_2) e^{i\mu(\phi_1 - \phi_2)} \right. \\ &+ \sum_{\mu=1}^{\lambda} (-1)^\mu (2\lambda + 1) \left[\frac{(\lambda - \mu)!}{(\lambda + \mu)!} \right]^2 \mathcal{P}_\lambda^\mu(\xi_{<}) \mathcal{Q}_\lambda^\mu(\xi_{>}) \mathcal{P}_\lambda^\mu(\eta_1) \mathcal{P}_\lambda^{\mu*}(\eta_2) e^{i\mu(\phi_1 - \phi_2)} \\ &\left. + (-1)^0 (2\lambda + 1) \left[\frac{(\lambda - 0)!}{(\lambda + 0)!} \right]^2 \mathcal{P}_\lambda^0(\xi_{<}) \mathcal{Q}_\lambda^0(\xi_{>}) \mathcal{P}_\lambda^0(\eta_1) \mathcal{P}_\lambda^{0*}(\eta_2) e^{i0(\phi_1 - \phi_2)} \right) \end{aligned} \quad (5.60)$$

Το πρώτο σκέλος της (5.60), εντός της παρενθέσεως, γίνεται βάσει της (5.42) :

$$\begin{aligned}
\frac{1}{r_{12}} &= \frac{2}{R} \sum_{\lambda=0}^{\infty} \left(\sum_{\mu=-\lambda}^{-1} (-1)^{-\mu} (-1)^{-2\mu} (2\lambda+1) \left[\frac{(\lambda+\mu)!}{(\lambda-\mu)!} \right]^2 \mathcal{P}_{\lambda}^{-\mu}(\xi_{<}) \mathcal{Q}_{\lambda}^{-\mu}(\xi_{>}) \mathcal{P}_{\lambda}^{-\mu}(\eta_1) \mathcal{P}_{\lambda}^{-\mu*}(\eta_2) e^{i\mu(\phi_1-\phi_2)} \right. \\
&+ \sum_{\mu=1}^{\lambda} (-1)^{\mu} (2\lambda+1) \left[\frac{(\lambda-\mu)!}{(\lambda+\mu)!} \right]^2 \mathcal{P}_{\lambda}^{\mu}(\xi_{<}) \mathcal{Q}_{\lambda}^{\mu}(\xi_{>}) \mathcal{P}_{\lambda}^{\mu}(\eta_1) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) e^{i\mu(\phi_1-\phi_2)} \\
&+ \left. (-1)^0 (2\lambda+1) \left[\frac{(\lambda-0)!}{(\lambda+0)!} \right]^2 \mathcal{P}_{\lambda}^0(\xi_{<}) \mathcal{Q}_{\lambda}^0(\xi_{>}) \mathcal{P}_{\lambda}^0(\eta_1) \mathcal{P}_{\lambda}^{0*}(\eta_2) e^{i0(\phi_1-\phi_2)} \right) \\
&\xrightarrow{-\mu \rightarrow \mu} \frac{2}{R} \sum_{\lambda=0}^{\infty} \left(\sum_{\mu=1}^{\lambda} (-1)^{\mu} (2\lambda+1) \left[\frac{(\lambda-\mu)!}{(\lambda+\mu)!} \right]^2 \mathcal{P}_{\lambda}^{\mu}(\xi_{<}) \mathcal{Q}_{\lambda}^{\mu}(\xi_{>}) \mathcal{P}_{\lambda}^{\mu}(\eta_1) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) e^{-i\mu(\phi_1-\phi_2)} \right. \\
&+ \sum_{\mu=1}^{\lambda} (-1)^{\mu} (2\lambda+1) \left[\frac{(\lambda-\mu)!}{(\lambda+\mu)!} \right]^2 \mathcal{P}_{\lambda}^{\mu}(\xi_{<}) \mathcal{Q}_{\lambda}^{\mu}(\xi_{>}) \mathcal{P}_{\lambda}^{\mu}(\eta_1) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) e^{i\mu(\phi_1-\phi_2)} \\
&+ \left. (-1)^0 (2\lambda+1) \left[\frac{(\lambda-0)!}{(\lambda+0)!} \right]^2 \mathcal{P}_{\lambda}^0(\xi_{<}) \mathcal{Q}_{\lambda}^0(\xi_{>}) \mathcal{P}_{\lambda}^0(\eta_1) \mathcal{P}_{\lambda}^{0*}(\eta_2) e^{i0(\phi_1-\phi_2)} \right) \tag{5.61}
\end{aligned}$$

όπου θέσαμε $-\mu \rightarrow \mu$ στον πρώτο όρο, διότι η άθροιση ήταν πάνω σε αρνητικά μ : επιπλέον αντιστρέψαμε τα όρια του αθροίσματος εφόσον ισχύει η αντιμεταθετική ιδιότητα⁷ και κατά συνέπεια αυτό είναι αναλλοίωτο ως προς την αντιστροφή.

Λαμβάνοντας υπ' όψιν ότι οι δύο πρώτοι όροι εντός παρενθέσεως είναι ίσοι, εκτός του εκθετικού, καθώς και ότι

$$e^{i\mu(\phi_1-\phi_2)} + e^{-i\mu(\phi_1-\phi_2)} = 2 \cos [\mu(\phi_1 - \phi_2)] \tag{5.62}$$

η (5.61) μπορεί να γραφεί ισοδύναμα, συμπεριλαμβανομένου του όρου

⁷ Η μεταθετική ιδιότητα, που έχει μεταφερθεί λανθασμένα στην ελληνική γλώσσα ως αντιμεταθετική, είναι η γνωστή :

Ορισμός 5.4.1. Στο σώμα των ακεραίων αριθμών η πράξη της πρόσθεσης και του πολλαπλασιασμού έχουν την αντιμεταθετική ιδιότητα :

$$\begin{aligned}
a + b &= b + a \\
a * b &= b * a, \quad \forall a, b \in \mathbb{Z}
\end{aligned}$$

για $\mu = 0$, ως :

$$\frac{2}{R} \sum_{\lambda=0}^{\infty} \sum_{\mu=0}^{\lambda} (-1)^{\mu} \alpha_{\mu} (2\lambda + 1) \left[\frac{(\lambda - \mu)!}{(\lambda + \mu)!} \right]^2 \mathcal{P}_{\lambda}^{\mu}(\xi_{<}) \mathcal{Q}_{\lambda}^{\mu}(\xi_{>}) \mathcal{P}_{\lambda}^{\mu}(\eta_1) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) \cos [\mu(\phi_1 - \phi_2)]$$

με

$$\alpha_{\mu} = \begin{cases} 1, & \text{εάν } \mu = 0 \text{ ,} \\ 2, & \text{εάν } \mu > 1 \text{ .} \end{cases} \quad (5.63)$$

□

Επιστρέφουμε τώρα στην ανάπτυξη των διπλεκτρονιακών ολοκληρωμάτων⁸. Ξεκινώντας από την (5.14), αντικαθιστούμε τα τροχιακά από την (5.16) και χωρίζουμε τα ακτινικά ολοκληρώματα σε δύο διαστήματα [2]: $[1, \xi_1]$ και $[\xi_1, \infty)$:

$$\begin{aligned} & \left\langle u_i(\mathbf{r}_1) u_j(\mathbf{r}_2) \left| \frac{1}{r_{12}} \right| v_k(\mathbf{r}_1) v_l(\mathbf{r}_2) \right\rangle \\ &= \iint u_i^*(\mathbf{r}_1) u_j^*(\mathbf{r}_2) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} v_k(\mathbf{r}_1) v_l(\mathbf{r}_2) d\mathbf{r}_1^3 d\mathbf{r}_2^3 \\ &\stackrel{(5.40)}{=} \frac{8\pi}{R} \left(\frac{R^3}{8} \right)^2 \sum_{\lambda=0}^{\infty} \sum_{\mu=-\lambda}^{\lambda} (-1)^{\mu} \frac{(\lambda - \mu)!}{(\lambda + \mu)!} \frac{(2\lambda + 1)(\lambda - \mu)!}{4\pi(\lambda + \mu)!} \\ & \times \iiint \Xi_i^*(\xi_1) \Xi_j^*(\xi_2) \Xi_k(\xi_1) \Xi_l(\xi_2) \\ & \times H_i^*(\eta_1) H_j^*(\eta_2) H_k(\eta_1) H_l(\eta_2) e^{i(m_k - m_i)\phi_1} e^{i(m_l - m_j)\phi_2} \\ & \times \mathcal{P}_{\lambda}^{\mu}(\xi_{<}) \mathcal{Q}_{\lambda}^{\mu}(\xi_{>}) \mathcal{P}_{\lambda}^{\mu}(\eta_1) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) \\ & \times e^{i\mu\phi_1} e^{-i\mu\phi_2} (\xi_1^2 - \eta_1^2)(\xi_2^2 - \eta_2^2) d\xi_1 d\xi_2 d\eta_1 d\eta_2 d\phi_1 d\phi_2 \longrightarrow \end{aligned}$$

⁸ Διατηρούμε το μγαδικό * στις συναρτήσεις παρά το γεγονός ότι είναι πραγματικές, ώστε να είναι πιο ευανάγνωστη η παραπάνω μακροσκελής έκφραση.

$$\begin{aligned}
 & \rightarrow = \frac{\pi R^5}{8} \frac{1}{4\pi} \sum_{\lambda=0}^{\infty} \sum_{\mu=-\lambda}^{\lambda} (-1)^{\mu} (2\lambda + 1) \left[\frac{(\lambda - \mu)!}{(\lambda + \mu)!} \right]^2 \\
 & \quad \times \iiint \Xi_i^*(\xi_1) \Xi_k(\xi_1) H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_{\lambda}^{\mu}(\eta_1) (\xi_1^2 - \eta_1^2) \\
 & \quad \times \Xi_j^*(\xi_2) \Xi_l(\xi_2) H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) (\xi_2^2 - \eta_2^2) \\
 & \quad \times \mathcal{P}_{\lambda}^{\mu}(\xi_{<}) \mathcal{Q}_{\lambda}^{\mu}(\xi_{>}) d\xi_1 d\xi_2 d\eta_1 d\eta_2 \\
 & \quad \times \underbrace{\int_0^{2\pi} e^{i(m_k + \mu - m_i)\phi_1} d\phi_1}_{2\pi\delta_{m_i, m_k + \mu}} \underbrace{\int_0^{2\pi} e^{i[m_l - (\mu + m_j)]\phi_2} d\phi_2}_{2\pi\delta_{m_l, m_j + \mu}} \\
 & = \frac{\pi^2 R^5}{8} \sum_{\lambda=0}^{\infty} \sum_{\mu=-\lambda}^{\lambda} (-1)^{\mu} (2\lambda + 1) \left[\frac{(\lambda - \mu)!}{(\lambda + \mu)!} \right]^2 \delta_{m_i, m_k + \mu} \delta_{m_l, m_j + \mu} \\
 & \quad \times \int_1^{\infty} d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \int_{-1}^1 d\eta_1 (\xi_1^2 - \eta_1^2) H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_{\lambda}^{\mu}(\eta_1) \\
 & \quad \times \int_1^{\infty} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{P}_{\lambda}^{\mu}(\xi_{<}) \mathcal{Q}_{\lambda}^{\mu}(\xi_{>}) \int_{-1}^1 d\eta_2 (\xi_2^2 - \eta_2^2) H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) \\
 & = \frac{\pi^2 R^5}{8} \sum_{\lambda=0}^{\infty} \sum_{\mu=-\lambda}^{\lambda} (-1)^{\mu} (2\lambda + 1) \left[\frac{(\lambda - \mu)!}{(\lambda + \mu)!} \right]^2 \delta_{m_i, m_k + \mu} \delta_{m_l, m_j + \mu} \\
 & \quad \times \int_1^{\infty} d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \int_{-1}^1 d\eta_1 (\xi_1^2 - \eta_1^2) H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_{\lambda}^{\mu}(\eta_1) \\
 & \quad \times \left(\mathcal{Q}_{\lambda}^{\mu}(\xi_1) \int_1^{\xi_1} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{P}_{\lambda}^{\mu}(\xi_2) \xi_2^2 \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) \right. \\
 & \quad - \mathcal{Q}_{\lambda}^{\mu}(\xi_1) \int_1^{\xi_1} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{P}_{\lambda}^{\mu}(\xi_2) \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) \eta_2^2 \\
 & \quad + \mathcal{P}_{\lambda}^{\mu}(\xi_1) \int_{\xi_1}^{\infty} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{Q}_{\lambda}^{\mu}(\xi_2) \xi_2^2 \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) \\
 & \quad \left. - \mathcal{P}_{\lambda}^{\mu}(\xi_1) \int_{\xi_1}^{\infty} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{Q}_{\lambda}^{\mu}(\xi_2) \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) \eta_2^2 \right) \quad (5.64)
 \end{aligned}$$

Αναπτύσσουμε περαιτέρω την (5.64)

$$\begin{aligned}
& \frac{\pi^2 R^5}{8} \sum_{\lambda=0}^{\infty} \sum_{\mu=-\lambda}^{\lambda} (-1)^\mu (2\lambda + 1) \left[\frac{(\lambda - \mu)!}{(\lambda + \mu)!} \right]^2 \delta_{m_i, m_k + \mu} \delta_{m_l, m_j + \mu} \\
& \times \left(\int_1^{\infty} d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{Q}_\lambda^\mu(\xi_1) \xi_1^2 \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \right. \\
& \times \int_1^{\xi_1} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{P}_\lambda^\mu(\xi_2) \xi_2^2 \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \\
& - \int_1^{\infty} d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{Q}_\lambda^\mu(\xi_1) \xi_1^2 \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \\
& \times \int_1^{\xi_1} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{P}_\lambda^\mu(\xi_2) \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \eta_2^2 \\
& + \int_1^{\infty} d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{P}_\lambda^\mu(\xi_1) \xi_1^2 \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \\
& \times \int_{\xi_1}^{\infty} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{Q}_\lambda^\mu(\xi_2) \xi_2^2 \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \\
& - \int_1^{\infty} d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{P}_\lambda^\mu(\xi_1) \xi_1^2 \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \\
& \times \int_{\xi_1}^{\infty} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{Q}_\lambda^\mu(\xi_2) \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \eta_2^2 \\
& - \int_1^{\infty} d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{Q}_\lambda^\mu(\xi_1) \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \eta_1^2 \\
& \times \int_1^{\xi_1} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{P}_\lambda^\mu(\xi_2) \xi_2^2 \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \\
& + \int_1^{\infty} d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{Q}_\lambda^\mu(\xi_1) \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \eta_1^2 \\
& \times \int_1^{\xi_1} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{P}_\lambda^\mu(\xi_2) \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \eta_2^2 \\
& - \int_1^{\infty} d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{P}_\lambda^\mu(\xi_1) \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \eta_1^2 \\
& \times \int_{\xi_1}^{\infty} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{Q}_\lambda^\mu(\xi_2) \xi_2^2 \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \\
& + \int_1^{\infty} d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{P}_\lambda^\mu(\xi_1) \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \eta_1^2 \\
& \times \int_{\xi_1}^{\infty} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{Q}_\lambda^\mu(\xi_2) \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \eta_2^2 \Big) \quad (5.65)
\end{aligned}$$

Αν αντί της (5.40), χρησιμοποιήσουμε τον τύπο (5.58), ώστε να αποφύγουμε τις επιπλοκές από την άθροιση στα “αρνητικά” μ και να επιτύχουμε την συντόμευση του υπολογισμού, μέσω τις εμφανίσεις περισσότερων δέλτα του Kronecker, τότε

$$\begin{aligned}
& \left\langle u_i(\mathbf{r}_1)u_j(\mathbf{r}_2) \left| \frac{1}{r_{12}} \right| v_k(\mathbf{r}_1)v_l(\mathbf{r}_2) \right\rangle \\
&= \iint u_i^*(\mathbf{r}_1)u_j^*(\mathbf{r}_2) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} v_k(\mathbf{r}_1)v_l(\mathbf{r}_2) d\mathbf{r}_1^3 d\mathbf{r}_2^3 \\
&\stackrel{(5.58)}{=} \frac{2}{R} \left(\frac{R^3}{8} \right)^2 \sum_{\lambda=0}^{\infty} \sum_{\mu=0}^{\lambda} (-1)^\mu \alpha_\mu (2\lambda + 1) \left[\frac{(\lambda - \mu)!}{(\lambda + \mu)!} \right]^2 \\
&\quad \times \iiint \iiint \Xi_i^*(\xi_1)\Xi_j^*(\xi_2)\Xi_k(\xi_1)\Xi_l(\xi_2) \\
&\quad \times H_i^*(\eta_1)H_j^*(\eta_2)H_k(\eta_1)H_l(\eta_2) e^{i(m_k - m_i)\phi_1} e^{i(m_l - m_j)\phi_2} \\
&\quad \times \mathcal{P}_\lambda^\mu(\xi_<) \mathcal{Q}_\lambda^\mu(\xi_>) \mathcal{P}_\lambda^\mu(\eta_1) \mathcal{P}_\lambda^{\mu*}(\eta_2) \\
&\quad \times \cos[\mu(\phi_1 - \phi_2)] (\xi_1^2 - \eta_1^2)(\xi_2^2 - \eta_2^2) d\xi_1 d\xi_2 d\eta_1 d\eta_2 d\phi_1 d\phi_2 \longrightarrow
\end{aligned}$$

$$\begin{aligned}
& \rightarrow = \frac{R^5}{32} \sum_{\lambda=0}^{\infty} \sum_{\mu=0}^{\lambda} (-1)^{\mu} \alpha_{\mu} (2\lambda + 1) \left[\frac{(\lambda - \mu)!}{(\lambda + \mu)!} \right]^2 \\
& \quad \times \iiint \Xi_i^*(\xi_1) \Xi_k(\xi_1) H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_{\lambda}^{\mu}(\eta_1) (\xi_1^2 - \eta_1^2) \\
& \quad \times \Xi_j^*(\xi_2) \Xi_l(\xi_2) H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) (\xi_2^2 - \eta_2^2) \\
& \quad \times \mathcal{P}_{\lambda}^{\mu}(\xi_{<}) \mathcal{Q}_{\lambda}^{\mu}(\xi_{>}) d\xi_1 d\xi_2 d\eta_1 d\eta_2 \\
& \quad \times \underbrace{\int_0^{2\pi} \int_0^{2\pi} e^{i(m_k - m_i)\phi_1} e^{i(m_l - m_j)\phi_2} \cos[\mu(\phi_1 - \phi_2)] d\phi_1 d\phi_2}_{2\pi^2 (\delta_{m_i, m_k + \mu} \delta_{m_l, m_j + \mu} + \delta_{m_i, m_k - \mu} \delta_{m_l, m_j - \mu})} \\
& = \frac{\pi^2 R^5}{8} \sum_{\lambda=0}^{\infty} \sum_{\mu=0}^{\lambda} (-1)^{\mu} \alpha_{\mu} \frac{(2\lambda + 1)}{2} \left[\frac{(\lambda - \mu)!}{(\lambda + \mu)!} \right]^2 (\delta_{m_i, m_k + \mu} \delta_{m_l, m_j + \mu} + \delta_{m_i, m_k - \mu} \delta_{m_l, m_j - \mu}) \\
& \quad \times \int_1^{\infty} d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \int_{-1}^1 d\eta_1 (\xi_1^2 - \eta_1^2) H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_{\lambda}^{\mu}(\eta_1) \\
& \quad \times \int_1^{\infty} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{P}_{\lambda}^{\mu}(\xi_{<}) \mathcal{Q}_{\lambda}^{\mu}(\xi_{>}) \int_{-1}^1 d\eta_2 (\xi_2^2 - \eta_2^2) H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) \\
& = \frac{\pi^2 R^5}{8} \sum_{\lambda=0}^{\infty} \sum_{\mu=0}^{\lambda} (-1)^{\mu} \alpha_{\mu} \frac{(2\lambda + 1)}{2} \left[\frac{(\lambda - \mu)!}{(\lambda + \mu)!} \right]^2 (\delta_{m_i, m_k + \mu} \delta_{m_l, m_j + \mu} + \delta_{m_i, m_k - \mu} \delta_{m_l, m_j - \mu}) \\
& \quad \times \int_1^{\infty} d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \int_{-1}^1 d\eta_1 (\xi_1^2 - \eta_1^2) H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_{\lambda}^{\mu}(\eta_1) \\
& \quad \times \left(\mathcal{Q}_{\lambda}^{\mu}(\xi_1) \int_1^{\xi_1} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{P}_{\lambda}^{\mu}(\xi_2) \xi_2^2 \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) \right. \\
& \quad - \mathcal{Q}_{\lambda}^{\mu}(\xi_1) \int_1^{\xi_1} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{P}_{\lambda}^{\mu}(\xi_2) \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) \eta_2^2 \\
& \quad + \mathcal{P}_{\lambda}^{\mu}(\xi_1) \int_{\xi_1}^{\infty} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{Q}_{\lambda}^{\mu}(\xi_2) \xi_2^2 \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) \\
& \quad \left. - \mathcal{P}_{\lambda}^{\mu}(\xi_1) \int_{\xi_1}^{\infty} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{Q}_{\lambda}^{\mu}(\xi_2) \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_{\lambda}^{\mu*}(\eta_2) \eta_2^2 \right) (5.66)
\end{aligned}$$

Ομοίως, αναπτύσσουμε περαιτέρω

$$\begin{aligned}
& \frac{\pi^2 R^5}{8} \sum_{\lambda=0}^{\infty} \sum_{\mu=0}^{\lambda} (-1)^\mu \alpha_\mu \frac{(2\lambda+1)}{2} \left[\frac{(\lambda-\mu)!}{(\lambda+\mu)!} \right]^2 (\delta_{m_i, m_k + \mu} \delta_{m_l, m_j + \mu} + \delta_{m_i, m_k - \mu} \delta_{m_l, m_j - \mu}) \\
& \times \left(\int_1^\infty d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{Q}_\lambda^\mu(\xi_1) \xi_1^2 \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \right. \\
& \times \int_1^{\xi_1} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{P}_\lambda^\mu(\xi_2) \xi_2^2 \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \\
& - \int_1^\infty d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{Q}_\lambda^\mu(\xi_1) \xi_1^2 \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \\
& \times \int_1^{\xi_1} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{P}_\lambda^\mu(\xi_2) \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \eta_2^2 \\
& + \int_1^\infty d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{P}_\lambda^\mu(\xi_1) \xi_1^2 \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \\
& \times \int_{\xi_1}^\infty d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{Q}_\lambda^\mu(\xi_2) \xi_2^2 \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \\
& - \int_1^\infty d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{P}_\lambda^\mu(\xi_1) \xi_1^2 \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \\
& \times \int_{\xi_1}^\infty d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{Q}_\lambda^\mu(\xi_2) \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \eta_2^2 \\
& - \int_1^\infty d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{Q}_\lambda^\mu(\xi_1) \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \eta_1^2 \\
& \times \int_1^{\xi_1} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{P}_\lambda^\mu(\xi_2) \xi_2^2 \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \\
& + \int_1^\infty d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{Q}_\lambda^\mu(\xi_1) \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \eta_1^2 \\
& \times \int_1^{\xi_1} d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{P}_\lambda^\mu(\xi_2) \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \eta_2^2 \\
& - \int_1^\infty d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{P}_\lambda^\mu(\xi_1) \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \eta_1^2 \\
& \times \int_{\xi_1}^\infty d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{Q}_\lambda^\mu(\xi_2) \xi_2^2 \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \\
& + \int_1^\infty d\xi_1 \Xi_i^*(\xi_1) \Xi_k(\xi_1) \mathcal{P}_\lambda^\mu(\xi_1) \int_{-1}^1 d\eta_1 H_i^*(\eta_1) H_k(\eta_1) \mathcal{P}_\lambda^\mu(\eta_1) \eta_1^2 \\
& \times \left. \int_{\xi_1}^\infty d\xi_2 \Xi_j^*(\xi_2) \Xi_l(\xi_2) \mathcal{Q}_\lambda^\mu(\xi_2) \int_{-1}^1 d\eta_2 H_j^*(\eta_2) H_l(\eta_2) \mathcal{P}_\lambda^{\mu*}(\eta_2) \eta_2^2 \right) \quad (5.67)
\end{aligned}$$

Παρατηρούμε πως με την μετατροπή σε άθροιση πάνω στα μή αρνητικά μ μόνο, η συνεισφορά των αρνητικών μ μεταφέρθηκε στα δέλτα του *Kronecker*⁹. Είναι προφανές πως με την μετατροπή αυτή, ο υπολογισμός των διλεκτρονιακών μητροστοιχείων, θα είναι πιο εύκολος και πιο σύντομος, καθώς αθροίζουμε μόνο για $\mu \geq 0$.

Εξαγωγή των Kronecker δέλτα

Κατόπιν αποδεικνύουμε τις σχέσεις

$$\int_0^{2\pi} e^{i(m-\mu)\phi} d\phi = 2\pi\delta_{m,\mu} \quad (5.69)$$

$$\int_0^{2\pi} \int_0^{2\pi} e^{im\phi_1} e^{im'\phi_2} \cos[\mu(\phi_1 - \phi_2)] d\phi_1 d\phi_2 = 2\pi^2(\delta_{m,-\mu}\delta_{m',\mu} + \delta_{m,\mu}\delta_{m',-\mu}) \quad (5.70)$$

όπου $m = m_k - m_i$ ή $m = m_l - m_j$ και $m' = m_l - m_j$.

Απόδειξη. Αρχίζουμε από την (5.69),

$$\begin{aligned} \int_0^{2\pi} e^{i(m-\mu)\phi} d\phi &= \int_0^{2\pi} \cos[(m-\mu)\phi] d\phi + i \int_0^{2\pi} \sin[(m-\mu)\phi] d\phi \\ &= \begin{cases} \int_0^{2\pi} 1 d\phi + i \int_0^{2\pi} 0 d\phi, & \text{εάν } m = \mu, \\ \left[\frac{\sin[(m-\mu)\phi]}{m-\mu} \right]_0^{2\pi} - i \left[\frac{\cos[(m-\mu)\phi]}{m-\mu} \right]_0^{2\pi}, & \text{εάν } m \neq \mu. \end{cases} \\ &= \begin{cases} 2\pi, & \text{εάν } m = \mu \\ 0, & \text{εάν } m \neq \mu \end{cases} \Rightarrow \int_0^{2\pi} e^{i(m-\mu)\phi} d\phi = 2\pi\delta_{m,\mu} \end{aligned}$$

□

⁹ Το δέλτα του Kronecker ορίζεται ως εξής :

$$\delta_{m,m'} := \begin{cases} 1, & \text{εάν } m = m' \\ 0, & \text{εάν } m \neq m'. \end{cases} \quad (5.68)$$

Απόδειξη. Για την (5.70) έχουμε :

$$\begin{aligned} & \int_0^{2\pi} \int_0^{2\pi} e^{im\phi_1} e^{im'\phi_2} \cos[\mu(\phi_1 - \phi_2)] d\phi_1 d\phi_2 \\ &= \int_0^{2\pi} e^{im\phi_1} \cos \mu\phi_1 d\phi_1 \int_0^{2\pi} e^{im'\phi_2} \cos \mu\phi_2 d\phi_2 \\ & \quad + \int_0^{2\pi} e^{im\phi_1} \sin \mu\phi_1 d\phi_1 \int_0^{2\pi} e^{im'\phi_2} \sin \mu\phi_2 d\phi_2 \end{aligned} \quad (5.71)$$

Όμως από τη ταυτότητα του *Euler* ισχύει :

$$\int_0^{2\pi} e^{im\phi} \cos \mu\phi d\phi = \int_0^{2\pi} \cos m\phi \cos \mu\phi d\phi + i \int_0^{2\pi} \sin m\phi \cos \mu\phi d\phi \quad (5.72)$$

Για να υπολογίσουμε τα παραπάνω ολοκληρώματα, χρησιμοποιούμε τις γνωστές τριγωνομετρικές ταυτότητες :

$$\begin{aligned} & \rightarrow \begin{cases} \cos[(m + \mu)\phi] = \cos m\phi \cos \mu\phi - \sin m\phi \sin \mu\phi \\ \cos[(m - \mu)\phi] = \cos m\phi \cos \mu\phi + \sin m\phi \sin \mu\phi \end{cases} \\ & \xrightarrow{(+)/(-)} \begin{cases} \cos m\phi \cos \mu\phi = \frac{\cos[(m + \mu)\phi] + \cos[(m - \mu)\phi]}{2} \\ \sin m\phi \sin \mu\phi = \frac{\cos[(m + \mu)\phi] - \cos[(m - \mu)\phi]}{2} \end{cases} \end{aligned} \quad (5.73)$$

και

$$\begin{aligned} & \rightarrow \begin{cases} \sin[(m + \mu)\phi] = \sin m\phi \cos \mu\phi + \cos m\phi \sin \mu\phi \\ \sin[(m - \mu)\phi] = \sin m\phi \cos \mu\phi - \cos m\phi \sin \mu\phi \end{cases} \\ & \xrightarrow{(+)/(-)} \begin{cases} \sin m\phi \cos \mu\phi = \frac{\sin[(m + \mu)\phi] + \sin[(m - \mu)\phi]}{2} \\ \cos m\phi \sin \mu\phi = \frac{\sin[(m + \mu)\phi] - \sin[(m - \mu)\phi]}{2} \end{cases} \end{aligned} \quad (5.74)$$

Αντικαθιστώντας τις (5.73) και (5.74) στην (5.72) έχουμε

$$\begin{aligned} & \int_0^{2\pi} \frac{\cos[(m + \mu)\phi] + \cos[(m - \mu)\phi]}{2} d\phi + i \int_0^{2\pi} \frac{\sin[(m + \mu)\phi] + \sin[(m - \mu)\phi]}{2} d\phi \\ &= \frac{1}{2} \left[\int_0^{2\pi} \cos[(m + \mu)\phi] d\phi + \int_0^{2\pi} \cos[(m - \mu)\phi] d\phi \right. \\ & \quad \left. + i \left(\int_0^{2\pi} \sin[(m + \mu)\phi] d\phi + \int_0^{2\pi} \sin[(m - \mu)\phi] d\phi \right) \right] \end{aligned} \quad (5.75)$$

Όμως

$$\int_0^{2\pi} \cos [(m \pm \mu)\phi] d\phi = \begin{cases} 2\pi, & \text{εάν } m = \mp\mu \\ 0, & \text{εάν } m \neq \mp\mu \end{cases} = 2\pi\delta_{m,\mp\mu} \quad (5.76)$$

και

$$\int_0^{2\pi} \sin [(m \pm \mu)\phi] d\phi = 0 \quad (5.77)$$

Επομένως η (5.75), μέσω των (5.76) και (5.77), γίνεται:

$$\frac{1}{2} (2\pi\delta_{m,-\mu} + 2\pi\delta_{m,\mu})$$

Τελικά, για την (5.72), έχουμε

$$\int_0^{2\pi} e^{im\phi} \cos \mu\phi d\phi = \pi (\delta_{m,-\mu} + \delta_{m,\mu}) \quad (5.78)$$

Ομοίως, για το ολοκλήρωμα με το ημίτονο έχουμε :

$$\int_0^{2\pi} e^{im\phi} \sin \mu\phi d\phi = \int_0^{2\pi} \cos m\phi \sin \mu\phi d\phi + i \int_0^{2\pi} \sin m\phi \sin \mu\phi d\phi \quad (5.79)$$

και μέσω των (5.73) και (5.74) παίρνει τη μορφή

$$\begin{aligned} & \int_0^{2\pi} \frac{\sin[(m+\mu)\phi] - \sin[(m-\mu)\phi]}{2} d\phi + i \int_0^{2\pi} \frac{\cos[(m-\mu)\phi] - \cos[(m+\mu)\phi]}{2} d\phi \\ &= \frac{1}{2} \left[\int_0^{2\pi} \sin[(m+\mu)\phi] d\phi - \int_0^{2\pi} \sin[(m-\mu)\phi] d\phi \right. \\ & \quad \left. + i \left(\int_0^{2\pi} \cos[(m-\mu)\phi] d\phi - \int_0^{2\pi} \cos[(m+\mu)\phi] d\phi \right) \right] \quad (5.80) \end{aligned}$$

ενώ παίρνοντας υπ' όψιν τις (5.76) και (5.77), από την (5.80) έχουμε

$$\frac{1}{2} (i2\pi\delta_{m,\mu} - i2\pi\delta_{m,-\mu})$$

Τελικά, η (5.79) γίνεται:

$$\int_0^{2\pi} e^{im\phi} \sin \mu\phi d\phi = i\pi (\delta_{m,\mu} - \delta_{m,-\mu}) \quad (5.81)$$

Έχοντας υπολογίσει τα δύο επιμέρους ολοκληρώματα (5.78) και (5.81), η αρχική (5.71) λαμβάνει τη ζητούμενη μορφή :

$$\begin{aligned} & \int_0^{2\pi} \int_0^{2\pi} e^{im\phi_1} e^{im'\phi_2} \cos [\mu(\phi_1 - \phi_2)] d\phi_1 d\phi_2 \\ & = 2\pi^2 (\delta_{m,-\mu} \delta_{m',\mu} + \delta_{m,\mu} \delta_{m',-\mu}) \end{aligned}$$

□

Γωνιακά ολοκληρώματα

Όπως βλέπουμε από την (5.65) τα γωνιακά ολοκληρώματα έχουν τη μορφή

$$\int_{-1}^1 d\eta H_i^*(\eta) H_k(\eta) \mathcal{P}_\lambda^\mu(\eta) \eta^b, \quad b \in \{0, 2\} \quad (5.82)$$

όπου η ολοκλήρωση γίνεται σε συμμετρικά όρια· άρα, εάν η υπό ολοκλήρωση συνάρτηση είναι περιττή, τότε το ολοκλίρωμα θα είναι μηδέν. Η συνάρτηση $H(\eta)$ έχει τρεις διαφορετικές μορφές· η (5.18) για ομοσπυρηνικό μόριο, εμπεριέχει τα συναφή πολυώνυμα Legendre πρώτου είδους, τα οποία και καθορίζουν την ομοτιμία της $H(\eta)$:

$$\sum_{i=0}^{\infty} \sum_{k=0}^{\infty} f_i^* f_k \underbrace{\int_{-1}^1 d\eta \mathcal{P}_{m+i}^{m*}(\eta) \mathcal{P}_{m+k}^m(\eta) \mathcal{P}_\lambda^\mu(\eta) \eta^b}_A, \quad b \in \{0, 2\} \quad (5.83)$$

Όμως, από την (5.51) βλέπουμε πως τα \mathcal{P}_l^m έχουν ομοτιμία $(-1)^{l+m}$ άρα για τα γωνιακά ολοκληρώματα έχουμε

$$A = \begin{cases} 0, & \text{εάν } 2m + i + 2m' + k + \lambda + \mu + b = 2n \\ a, & \text{εάν } 2m + i + 2m' + k + \lambda + \mu + b = 2n + 1 \end{cases} \quad (5.84)$$

όπου $n \in \mathbb{N}^0$ και $a \in \mathbb{R}$.

Η παραπάνω εξίσωση δεσμού μας επιτρέπει να υπολογίζουμε στην (5.65), μόνο τα ακτινικά ολοκληρώματα τα οποία επιβιώνουν από τον μηδενισμό των αντίστοιχων γωνιακών. Αυτό είναι πολύ σημαντικό, δεδομένου ότι η δυσκολία του υπολογισμού των ακτινικών ολοκληρωμάτων, αυξάνει σημαντικά τον συνολικό χρόνο του υπολογισμού.

5.5 Αντί Επιλόγου

Ολοκληρώσαμε εδώ την ανάλυση των μπροστοιχείων και την αναγωγή τους σε βασικά ολοκληρώματα, τα οποία και κωδικοποιούνται ώστε να λυθούν αριθμητικά μέσω ηλεκτρονικού υπολογιστή. Κατόπιν, βρίσκονται από αυτά οι αναμενόμενες τιμές καθώς και η επικάλυψη, που είναι απαραίτητες για τον υπολογισμό του συναρτησοειδούς \mathcal{F} . Η όλη διαδικασία επαναλαμβάνεται (αυτόματα) για διαφορετικά πλέον z^* και κατ' επέκταση OEDM, μέχρι να βρεθεί η ελάχιστη τιμή του \mathcal{F} . Οι καινούριες καταστάσεις που βρήκαμε από τον τελευταίο υπολογισμό, αποθηκεύονται (αυτόματα) ώστε να χρησιμοποιηθούν την επόμενη φορά ως κατώτερες καταστάσεις.

Μέρος IV
Παραρτήματα

[subsection]

Παράρτημα Α΄

Σφαιροειδείς Συντεταγμένες και η Εξίσωση Schrödinger

Α΄.1 Σφαιροειδείς συντεταγμένες

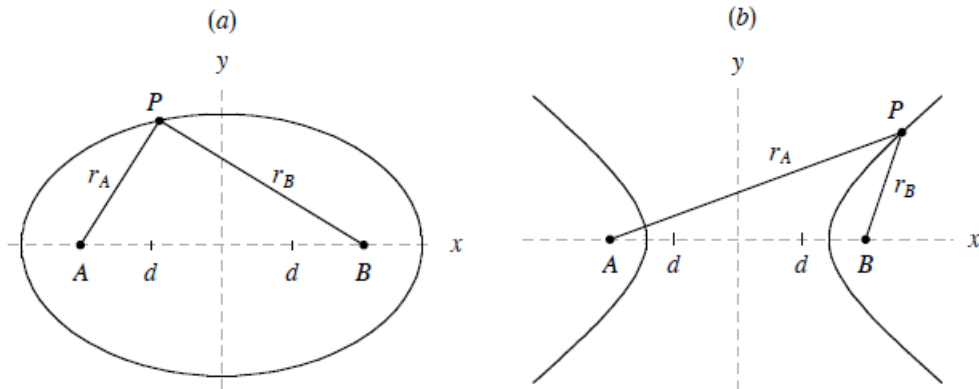
Α΄.1.1 Γεωμετρία Έλλειψης και Υπερβολής

Ορισμός Α΄.1.1.1. Έλλειψη είναι μια κλειστή καμπύλη που περιγράφεται από ένα σύνολο σημείων για τα οποία το άθροισμα των αποστάσεων από τα σημεία A και B (οι εστίες) είναι σταθερό (βλ. *a*) Α΄.1, σχ. 84).

Η ευθεία που ενώνει τα A και B ονομάζεται μεγάλος άξονας (ο x -άξονας στην εικόνα), ενώ η ευθεία που τέμνει μεσοκάθετα τον προηγούμενο ονομάζεται μικρός άξονας (ο y -άξονας) με $AB = 2d$. Ορθογώνια ως προς την έλλειψη είναι η υπερβολή :

Ορισμός Α΄.1.1.2. Υπερβολή είναι μια καμπύλη που περιγράφεται από ένα σύνολο σημείων για τα οποία η διαφορά μεταξύ των αποστάσεων από τα σημεία A και B είναι σταθερή (βλ. *b*) Α΄.1, σχ. 84).

Η υπερβολή δεν είναι κλειστή καμπύλη, αλλά αποτελείται από 2 ξεχωριστές καμπύλες.



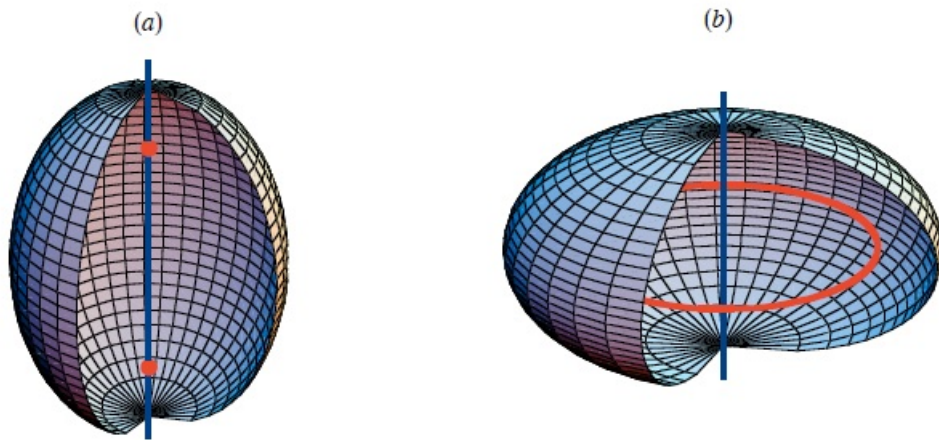
Σχήμα Α'.1: Έλλειψη και υπερβολή στο επίπεδο xy στο a) το $r_A + r_B$ είναι σταθερό, ενώ στο b) το $r_A - r_B$ είναι σταθερό.

Τα επιμήκη και πεπλατυσμένα σφαιροειδή:

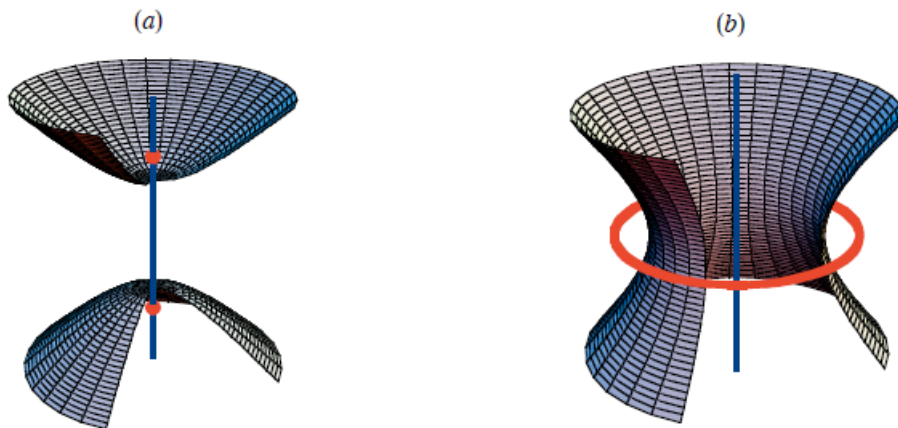
είναι οι επιφάνειες που σχηματίζονται εκ περιστροφής της έλλειψης, περί τον μεγάλο και τον μικρό άξονα αντίστοιχα. Οι εστίες A και B παραμένουν αμετάβλητες, από την περιστροφή, στο επιμήκες σφαιροειδές (σχ. a Α'.2 και Α'.3, σελ. 85). Αντίθετα, οι εστίες A και B περιστρέφονται στο πεπλατυσμένο σφαιροειδές, οπότε σχηματίζεται "εστιακός κύκλος". Τέλος, περιστρέφοντας την υπερβολή περι τον μεγάλο και τον μικρό άξονα παράγεται το υπερβολοειδές δύο επιφανειών και το υπερβολοειδές μίας επιφάνειας, αντίστοιχα (σχ. b Α'.2 και Α'.3, σελ. 85).

Α'.1.2 Επιμήκεις και Πεπλατυσμένες Σφαιροειδείς Συντεταγμένες

Οι σφαιροειδείς συντεταγμένες, (ξ, η, ϕ) , κατασκευάζονται από τον συνδυασμό των επιμήκων και πεπλατυσμένων σφαιροειδών με τα αντίστοιχα υπερβολοειδή. Το αζιμούθιο, $\phi = \tan^{-1}(\frac{y}{x})$, μετράει τη γωνία η οποία αυξάνει δεξιόστροφα γύρω από τον άξονα z , τα ξ και η ονομάζονται ακτινική και γωνιακή συντεταγμένη αντίστοιχα, επειδή ανάγονται στη σφαιρική ακτινική και γωνιακή μεταβλητή αντίστοι-



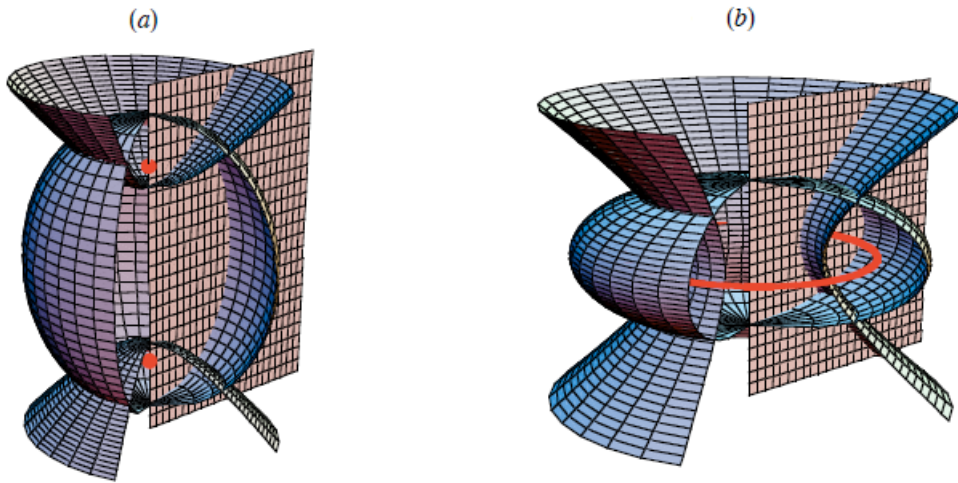
Σχήμα Α.2: α) Επίμηκες και β) πεπλατυσμένο σφαιροειδές ως επιφάνειες εκ περιστροφής.



Σχήμα Α.3: α) Υπερβολοειδές δύο και β) ενός φύλλου ως επιφάνειες εκ περιστροφής.

χα, στο σφαιρικό όριο¹. Τα επιμήκη σφαιροειδή και τα υπερβολοειδή δύο "φύλλων" είναι επιφάνειες σταθερού ξ και n αντίστοιχα, για τις επιμήκεις σφαιροειδείς συντεταγμένες (σχ. α Α.4, σελ. 86). Για τις παπλατυσμένες σφαιροειδείς συντεταγμένες οι επιφάνειες σταθερού ξ και n είναι τα πεπλατυσμένα σφαιροειδή και τα υπερβολοειδή ενός "φύλλου" (σχ. β Α.4, σελ. 86).

¹ $\lim_{d \rightarrow 0} \xi = r$, $\lim_{d \rightarrow 0} \eta = \theta$



Σχήμα Α'.4: Επιφάνειες για σταθερές a) επιμήκειες και b) πεπλατυσμένες συντεταγμένες. Τα σφαιροειδή είναι επιφάνειες σταθερού ξ , τα υπερβολοειδή σταθερού η και τα ημι-επίπεδα σταθερού ϕ .

Α'.1.3 Σχέσεις Μετασχηματισμού

Οι επιμήκειες σφαιροειδείς συντεταγμένες (ξ, η, ϕ) σχετίζονται με τις καρτεσιανές (x, y, z) με τους μετασχηματισμούς :

$$x = d \sqrt{(1 - \eta^2)(\xi^2 - 1)} \cos \phi \quad (\text{A'.1}\alpha')$$

$$y = d \sqrt{(1 - \eta^2)(\xi^2 - 1)} \sin \phi \quad (\text{A'.1}\beta')$$

$$z = d\eta\xi \quad (\text{A'.1}\gamma')$$

με όρια

$$1 \leq \xi \leq \infty, \quad -1 \leq \eta \leq 1 \quad \text{και} \quad 0 \leq \phi \leq 2\pi \quad (\text{A'.2})$$

Για πεπλατυσμένες σφαιροειδείς συντεταγμένες οι αντίστοιχοι μετασχηματισμοί είναι :

$$x = d \sqrt{(1 - \eta^2)(\xi^2 + 1)} \cos \phi \quad (\text{A'.3}\alpha')$$

$$y = d \sqrt{(1 - \eta^2)(\xi^2 + 1)} \sin \phi \quad (\text{A'.3}\beta')$$

$$z = d\eta\xi \quad (\text{A'.3}\gamma')$$

με όρια

$$1 \leq \xi \leq \infty, \quad -1 \leq \eta \leq 1 \quad \text{και} \quad 0 \leq \phi \leq 2\pi \quad (\text{A'.4})$$

Σε επιμήκειες συντεταγμένες οι ξ και η σχετίζονται με τις εστιακές ακτίνες r_A και r_B ως εξής :

$$\xi = \frac{r_A + r_B}{2d}, \quad \eta = \frac{r_A - r_B}{2d} \quad (\text{A'.5})$$

Παρόμοια σύνδεση για τις πεπλατυσμένες συντεταγμένες δεν υπάρχει, εφόσον το πεπλατυσμένο σφαιροειδές έχει εστιακό κύκλο αντί για 2 εστίες και κατα συνέπεια τα r_A και r_B δεν ορίζονται καλώς.

Α'.2 Η εξίσωση Schrödinger σε σφαιροειδείς συντεταγμένες

Η ηλεκτρονική εξίσωση Schrödinger (2.16) για ένα διατομικό μόριο, λαμβάνοντας υπόψιν την προσέγγιση Born-Oppenheimer (κεφ. 2.3, σελ. 9), παίρνει τη μορφή :

$$\hat{H}\Psi(\mathbf{r}) = E\Psi(\mathbf{r}) \Rightarrow \left(\frac{1}{2}\nabla^2 + \frac{Z_A}{r_A} + \frac{Z_B}{r_B} + E \right) \Psi(\mathbf{r}) = 0, \quad (\text{A'.6})$$

όπου Z_A και Z_B τα πυρηνικά φορτία των πυρήνων A και B αντίστοιχα. Αρχικά θα πρέπει να εκφράσουμε τη Λαπλασιανή σε σφαιροειδείς συντεταγμένες :

Για γενικευμένες καμπυλόγραμμες συντεταγμένες (q_1, q_2, q_3) η Λαπλασιανή γράφεται[10] ως :

$$\nabla^2 = \frac{1}{h_1 h_2 h_3} \times \left(\frac{\partial}{\partial q_1} \left(\frac{h_2 h_3}{h_1} \frac{\partial}{\partial q_1} \right) + \frac{\partial}{\partial q_2} \left(\frac{h_3 h_1}{h_2} \frac{\partial}{\partial q_2} \right) + \frac{\partial}{\partial q_3} \left(\frac{h_1 h_2}{h_3} \frac{\partial}{\partial q_3} \right) \right) \quad (\text{A'.7})$$

όπου οι μετρικοί συντελεστές $h_i = \left| \frac{\partial \mathbf{r}}{\partial q_i} \right|$ ικανοποιούν την :

$$|d\mathbf{r}|^2 = \sum_{i=1}^3 (dr_i)^2 = dx^2 + dy^2 + dz^2 = h_1^2 dq_1^2 + h_2^2 dq_2^2 + h_3^2 dq_3^2, \quad (\text{A'.8})$$

$$\text{με } d\mathbf{r} = \sum_{i=1}^3 dr_i \hat{\mathbf{e}}_i = dx \hat{\mathbf{e}}_1 + dy \hat{\mathbf{e}}_2 + dz \hat{\mathbf{e}}_3 \quad (\text{A'.9})$$

$$\text{και } r_i = r_i(q_1, q_2, q_3) \Rightarrow dr_i = \sum_{j=1}^3 \frac{\partial r_i}{\partial q_j} dq_j, \quad (\text{A'.10})$$

Στο αποτέλεσμα αυτό καταλήγουμε τετραγωνίζοντας την παραπάνω (A'.10), οπότε παίρνουμε :

$$(dr_k)^2 = \sum_{i=1}^3 \sum_{j=1}^3 \frac{\partial r_k}{\partial q_i} \frac{\partial r_k}{\partial q_j} dq_i dq_j \quad (\text{A'.11})$$

και αθροίζοντας την (A'.11) για όλες τις συνιστώσες :

$$\begin{aligned} (dr)^2 &= \sum_{k=1}^3 \sum_{i=1}^3 \sum_{j=1}^3 \frac{\partial r_k}{\partial q_i} \frac{\partial r_k}{\partial q_j} dq_i dq_j \Rightarrow \\ \Rightarrow (dr)^2 &= \sum_{i=1}^3 \sum_{j=1}^3 g_{ij} dq_i dq_j \end{aligned} \quad (\text{A'.12})$$

$$\text{όπου } g_{ij} = \frac{\partial x}{\partial q_i} \frac{\partial x}{\partial q_j} + \frac{\partial y}{\partial q_i} \frac{\partial y}{\partial q_j} + \frac{\partial z}{\partial q_i} \frac{\partial z}{\partial q_j}, \quad (\text{A'.13})$$

η μετρική του χώρου.

Η εξίσωση (A'.13) μπορεί να θεωρηθεί σαν εσωτερικό γινόμενο ενός διανύσματος στη κατεύθυνση dq_i , με συνιστώσες $\left(\frac{\partial x}{\partial q_i}, \frac{\partial y}{\partial q_i}, \frac{\partial z}{\partial q_i} \right)$ με ένα παρόμοιο διάνυσμα στην κατεύθυνση dq_j . Αν οι συντεταγμένες q_i είναι κάθετες, οι συντελεστές g_{ij} μηδενίζονται για $i \neq j$.

Άρα, για ορθογώνια συστήματα συντεταγμένων, όπως οι ελλειπτικές συντεταγμένες, οι εξ. (A'.13) και (A'.12) εξειδικεύονται στις :

$$q_i \perp q_j \rightarrow g_{ij} = 0 \quad \forall i \neq j \Rightarrow g_{ii} = \sum_{k=1}^3 \frac{\partial^2 r_k}{\partial q_i^2} \quad (\text{A'.14})$$

$$\text{και } (dr)^2 = \sum_{i=1}^3 g_{ii} (dq_i)^2 = \sum_{i=1}^3 (h_i dq_i)^2, \quad (\text{A'.15})$$

οπου καταλήξαμε στην αρχική εξ. (A'.8).

Στη προκειμένη περίπτωση, οι μετρικοί συντελεστές είναι :

$$h_\xi = d \sqrt{\frac{\xi^2 \mp \eta^2}{\xi^2 - 1}}, \quad h_\eta = d \sqrt{\frac{\xi^2 \mp \eta^2}{1 - \eta^2}}, \quad h_\phi = d \sqrt{(1 - \eta^2)(\xi^2 - 1)}, \quad (\text{A'.16})$$

όπου τα πρόσημα \mp μας δίνουν τις επιμήκεις και πεπλατυσμένες συντεταγμένες, αντίστοιχα. Βέβαια, στο συγκεκριμένο πρόβλημα θα χρησιμοποιήσουμε επιμήκεις σφαιροειδείς συντεταγμένες², γεγονός που επιβάλλεται από τη συμμετρία του συστήματός μας.

Αντικαθιστώντας τις (A'.7), (A'.5) και (A'.16) στην (3.14) έχουμε :

$$\begin{aligned} & \frac{\partial}{\partial \eta} \left((1 - \eta^2) \frac{\partial \psi(\mathbf{r})}{\partial \eta} \right) + \frac{\partial}{\partial \xi} \left((\xi^2 - 1) \frac{\partial \psi(\mathbf{r})}{\partial \xi} \right) \\ & + \frac{\xi^2 - \eta^2}{(\xi^2 - 1)(1 - \eta^2)} \frac{\partial^2 \psi(\mathbf{r})}{\partial \phi^2} + \left(\alpha c \xi - \beta c \eta - c^2 (\xi^2 - \eta^2) \right) \psi(\mathbf{r}) = 0 \end{aligned} \quad (\text{A'.17})$$

όπου εισάγαμε τις παραμέτρους $\{\alpha, \beta, c\} \in \mathbb{R}^3$:

$$c = R \sqrt{\frac{-E}{2}}, \quad \alpha = (Z_A + Z_B) \sqrt{\frac{-2}{E}}, \quad \beta = (Z_A - Z_B) \sqrt{\frac{-2}{E}}. \quad (\text{A'.18})$$

A'.2.1 Διαχωρισμός των μεταβλητών

Γράφοντας την ψ ως γινόμενο τριών ανεξάρτητων συναρτίσεων $\psi(\xi, \eta, \phi) = \Xi(\xi)H(\eta)\Phi(\phi)$ και αντικαθιστώντας στην (A'.17) καταλήγουμε σε 3 συνήθεις διαφορικές εξισώσεις :

$$\frac{d}{d\xi} \left((\xi^2 - 1) \frac{d\Xi(\xi)}{d\xi} \right) - \left(\lambda - \alpha c \xi + c^2 \xi^2 + \frac{\mu^2}{\xi^2 \mp 1} \right) \Xi(\xi) = 0 \quad (\text{A'.19α'})$$

$$\frac{d}{d\eta} \left((1 - \eta^2) \frac{dH(\eta)}{d\eta} \right) + \left(\lambda - \beta c \eta + c^2 \eta^2 - \frac{\mu^2}{1 - \eta^2} \right) H(\eta) = 0 \quad (\text{A'.19β'})$$

$$\frac{d^2 \Phi(\phi)}{d\phi^2} + \mu^2 \Phi(\phi) = 0 \quad (\text{A'.19γ'})$$

² Η εξ. (A'.17) μπορεί να μετατραπεί από επιμήκης σε πεπλατυσμένη μορφή με το μετασχηματισμό $c \longleftrightarrow ic$ και $\xi \longleftrightarrow i\xi$

³ Η ενέργεια για δέσμες καταστάσεις (όπως είναι ένα διατομικό μόριο) είναι αρνητική $E < 0$, και γι' αυτό τοποθετούμε αρνητικό πρόσημο στην υποόριξη ποσότητα ώστε τελικά να μίν έχουμε μιγαδικές παραμέτρους.

όπου λ και μ οι σταθερές διαχωρισμού. Η λύση της (Α'.19γ') είναι τετραμμένη $\Phi_\mu(\phi) = \frac{1}{\sqrt{2\pi}} e^{\pm i\mu\phi}$, $\mu = 0, 1, 2, \dots$, αφού πρέπει να ικανοποιεί τις συνοριακές συνθήκες $\Phi(\phi + 2\pi) = \Phi(\phi)^4$, το οποίο υπονοεί ότι το μ^5 είναι ακέραιο.

Η ακτινική (Α'.19α') και η γωνιακή εξίσωση (Α'.19β') αποτελούν παραδείγματα της γενικευμένης διαφορικής εξίσωσης σφαιροειδούς, η οποία έχει την παρακάτω γενική μορφή :

$$\frac{d}{dx} \left((1-x^2) \frac{dy(x)}{dx} \right) + \left(\lambda - \beta c x + c^2 x^2 - \frac{m^2}{1-x^2} \right) y(x) = 0 \quad (\text{Α'.20})$$

και διαφέρει από την διαφορική εξίσωση σφαιροειδούς μόνο κατά τον γραμμικό όρο $\beta c x$.

Αριθμός Κόμβων

Στη κατασκευή των κυματοσυναρτήσεων είναι χρήσιμο να ορίσουμε τους κβαντικούς αριθμούς κόμβων N_ξ και N_η , που καθορίζουν τον αριθμό των κόμβων (ή αλλιώς μηδενισμών) στην ακτινική $\Xi(\xi)$ και στη γωνιακή $H(\eta)$ συνάρτηση αντίστοιχα. Τότε η μοριακή κατάσταση καθορίζεται πλήρως από την τριάδα (N_ξ, N_η, m) . Φυσικά, στο όριο του "ενωμένου ατόμου" (united atom limit) $R \rightarrow 0$ χρησιμοποιείται η "κλασική" τριάδα κβαντικών αριθμών (n, l, m) , όπου το n είναι ο κύριος κβαντικός αριθμός που καθορίζει το ενεργειακό επίπεδο, l είναι ο δευτερεύων (ή αξιμουθιακός) κβαντικός αριθμός που καθορίζει τη

⁴ Οι συνοριακές συνθήκες αυτές συνεπάγονται από την απαίτηση η συνολική κυματοσυνάρτηση να είναι μια μονότιμη συνάρτηση του χώρου. Έτσι, η $\Phi(\phi)$ πρέπει να είναι μια περιοδική συνάρτηση με περίοδο 2π το οποίο μας οδηγεί στα εξής :

- α) $\mu \geq 0$, γιατί αλλιώς οι λύσεις της (Α'.19γ') θα έχουν εκθετική μορφή (που φυσικά δεν είναι περιοδική συνάρτηση), γι' αυτό και τοποθετούμε μ^2 .
- β) Συνεπώς οι λύσεις θα είναι της μορφής $\cos m\phi$, $\sin m\phi$ και σε συνδυασμό με τη συνθήκη $\Phi(\phi + 2\pi) = \Phi(\phi)$ οδηγούμαστε στο συμπέρασμα πως $\mu \in \mathbb{Z}$.

⁵ Το $\mu \equiv m$ ονομάζεται και αξιμουθιακός κβαντικός αριθμός.

στροφορμή και m ο μαγνητικός κβαντικός αριθμός που καθορίζει τη προβολή της στροφορμής στον άξονα των z (l_z). Οι δύο τριάδες αριθμών σχετίζονται ως εξής :

$$N_\xi = n - l - 1 \quad \text{και} \quad N_\eta = l - m \quad (\text{A'.21})$$

ή αντίστροφα

$$n = N_\xi + N_\eta + m \quad \text{και} \quad l = N_\eta + m \quad (\text{A'.22})$$

Α'.3 Μελέτη Διαφορικών Εξισώσεων ΟΕDM

Στο προηγούμενο κεφάλαιο καταλήξαμε στις γενικευμένες διαφορικές εξισώσεις σφαιροειδούς (Α'.19α') και (Α'.19β'), μετά από των διαχωρισμών με μεταβλητών της εξίσωσης Schrödinger (Α'.6). Για αυτό το λόγο, χρησιμοποιήσαμε κυματοσυναρτήσεις της μορφής $\psi(\xi, \eta, \phi) = \Xi(\xi)H(\eta)\Phi(\phi)$, όπου $\mu \in \mathbb{Z}$ είναι η ιδιοτιμή του τελεστή της στροφορμής $\hat{L}_z = -i\frac{\partial}{\partial\phi}$, ο οποίος μετατίθεται με την την Χαμιλτονιανή $[\hat{H}_{el}, \hat{L}_z] = 0$ και άρα η προβολή στον άξονα z διατηρείται αυτό, βέβαια, είναι απόρροια της κυλινδρικής συμμετρίας του συστήματος, γεγονός που ανταναλάται στο δυναμικό του, ότι δηλ. είναι από ανεξάρτητο από τη γωνία ϕ ⁶ :

$$\hat{V} = -\frac{1}{r_A} - \frac{1}{r_B} + \frac{1}{R} = -\frac{2}{R(\xi + \eta)} - \frac{2}{R(\xi - \eta)} + \frac{1}{R} = \hat{V}(\xi, \eta; R). \quad (\text{A'.23})$$

Έτσι, λοιπόν, οι δύο τελεστές θα έχουν κοινό σύστημα ιδιοσυναρτήσεων, οπότε μπορούμε να επιλέξουμε τις λύσεις $\psi(\xi, \eta, \phi)$ έτσι ώστε να είναι ιδιοσυναρτήσεις του \hat{L}_z · άρα τελικά λαμβάνουν τη μορφή :

$$\psi(\xi, \eta, \phi) = \Xi(\xi)H(\eta)\frac{1}{\sqrt{2\pi}}e^{\pm i\mu\phi} \quad (\text{A'.24})$$

⁶ Κοινώς το σύστημα έχει αξιμουθιακή συμμετρία· εδώ έχουμε μια ακόμα επιβεβαίωση του θεωρήματος της Noether :

Θεώρημα Α'.3.0.1. Απο συμμετρίας στη φύση παράγονται αρχές διατήρησης στη φυσική [32].

Οι κυματοσυναρτήσεις αυτές περιγράφουν τα μονοηλεκτρονιακά διατομικά μόρια (OEDM) και ονομάζονται μονοηλεκτρονιακά διατομικά τροχιακά (OEDO). Οι $\Xi(\xi)$, $H(\eta)$ είναι οι λύσεις των εξισώσεων (Α'.19α') και (Α'.19β'), τις οποίες στη παρούσα εργασία θα τις χρησιμοποιήσουμε με τις παρακάτω μορφές :

Γωνιακή κυματική εξίσωση:

$$\begin{aligned} \frac{d}{d\eta} \left((1-\eta^2) \frac{dH(\eta)}{d\eta} \right) + \left(\lambda - \beta c \eta + c^2 \eta^2 - \frac{\mu^2}{1-\eta^2} \right) H(\eta) &= 0 \xrightarrow{(A'.18)} \\ \frac{d}{d\eta} \left((1-\eta^2) \frac{dH(\eta)}{d\eta} \right) + \left(\lambda - R(Z_A - Z_B) \eta - \frac{ER^2}{2} \eta^2 - \frac{m^2}{1-\eta^2} \right) H(\eta) &= 0 \\ \frac{d}{d\eta} \left((1-\eta^2) \frac{dH(\eta)}{d\eta} \right) \\ + \left(\mathcal{C} - (ip)^2 (1-\eta^2) - R(Z_A - Z_B) \eta - \frac{m^2}{1-\eta^2} \right) H(\eta) &= 0 \quad (A'.25) \end{aligned}$$

όπου $p \equiv c = R \sqrt{\frac{-E}{2}}$, $p \in \mathbb{R}^+$ και $\mathcal{C} = -(ip)^2 - \xi$: η σταθερά διαχωρισμού. Συγκεκριμένα, στην περίπτωση ομοπυρηνικού μορίου, όπου η (Α'.25) γίνεται η γωνιακή σφαιροειδής διαφορική εξίσωση, θέτουμε $\mathcal{C} = \mathfrak{A}_l^m(ip)$, $l = m, m+1, \dots$ που είναι η ιδιοτιμή σφαιροειδούς (βλ. (3.20), σελ. 25). Εφόσον η E πρέπει να είναι πεπερασμένη, θα ισχύει στο όριο του ενιαίου ατόμου ότι $\lim_{R \rightarrow 0} p \rightarrow 0$, ενώ η (Α'.25) ανάγεται στην συναφή διαφορική εξίσωση Legendre (Α'.31) με ιδιοτιμή την $\xi = -\mathfrak{A}_l^m(0) = -l(l+1)$.

Ακτινική κυματική εξίσωση:

Η ακτινική εξίσωση παίρνει παρόμοια μορφή, ενώ η ύπαρξη της παραμέτρου p την συζευγνύει με την (Α'.25)

$$\begin{aligned} \frac{d}{d\xi} \left((\xi^2 - 1) \frac{d\Xi(\xi)}{d\xi} \right) \\ + \left(-\mathcal{C} + R(Z_A + Z_B) \xi + (ip)^2 (\xi^2 - 1) - \frac{m^2}{\xi^2 - 1} \right) \Xi(\xi) &= 0 \quad (A'.26) \end{aligned}$$

όπου \mathcal{C} ορίζεται όπως προηγουμένως.

Α'.3.1 Αναγωγή στην διαφορική εξίσωση σφαιροειδούς

- Όταν $Z_A, Z_B \rightarrow 0$ η εξίσωση *Schrödinger* (Α'.6) ανάγεται στην εξίσωση *Helmholtz* : $(\nabla^2 + k^2)\psi(\mathbf{r}) = 0$, ενώ οι εξισώσεις (Α'.19α') και (Α'.19β') ανάγονται στις δ.ε σφαιροειδούς :

$$\frac{d}{d\xi} \left((\xi^2 - 1) \frac{d\Xi(\xi)}{d\xi} \right) - \left(\lambda + c^2 \xi^2 + \frac{\mu^2}{\xi^2 \mp 1} \right) \Xi(\xi) = 0 \quad (\text{Α'.26α'})$$

$$\frac{d}{d\eta} \left((1 - \eta^2) \frac{dH(\eta)}{d\eta} \right) + \left(\lambda + c^2 \eta^2 - \frac{\mu^2}{1 - \eta^2} \right) H(\eta) = 0 \quad (\text{Α'.26β'})$$

- Όταν $Z_B \rightarrow 0$ η εξίσωση *Schrödinger* (Α'.6) ανάγεται στην εξίσωση που περιγράφει το άτομο του \mathbf{H} , για το οποίο οι λύσεις βρίσκονται σε κλειστή μορφή, ενώ η ακτινική και γωνιακή εξίσωση γίνονται ίδιες.
- Όταν $Z_A = Z_B = 1$ η εξίσωση *Schrödinger* (Α'.6) ανάγεται στην εξίσωση που περιγράφει το μοριακό ιόν H_2^+ , για το οποίο μόνο η γωνιακή εξίσωση (Α'.19β') ανάγεται στη γωνιακή εξίσωση σφαιροειδούς (3.19β').
- Όταν, επιλέον του $Z_A = Z_B = 1$, και $c \rightarrow 0^7$, τότε οι επιμήκειες σφαιροειδείς συντεταγμένες (ξ, η, ϕ) ανάγονται στις σφαιρικές πολικές συντεταγμένες (r, θ, ϕ) (σχ. Α'.5, σελ. 94).

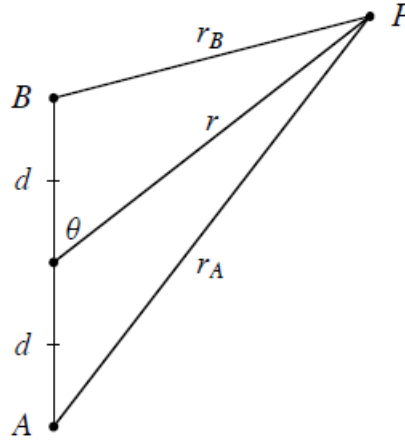
Απο το νόμο των συνημιτόνων έχουμε :

$$\Rightarrow \begin{cases} r_A^2 = r^2 + d^2 - 2rd \cos(\pi - \theta) = r^2 + d^2 + 2rd \cos \theta, \\ r_B^2 = r^2 + d^2 - 2rd \cos \theta \end{cases} \quad (\text{Α'.27})$$

ή

$$r_A^2 - r_B^2 = (r_A - r_B)(r_A + r_B) = 4dr \cos \theta \Rightarrow \quad (\text{Α'.28})$$

⁷ Το c ονομάζεται παράμετρος σφαιροειδούς και ισχύει ότι $c = kd$, όπου $2d = R$: η απόσταση μεταξύ των 2 εστίων της έλλειψης.



Σχήμα Α'5: Διαγραμματική απεικόνιση του συσχετισμού των σφαιρικών συντεταγμένων r, θ και των σφαιροειδών ακτίνων r_A, r_B .

$$\Rightarrow \begin{cases} \eta = \frac{r_A - r_B}{2d} = \frac{2r \cos \theta}{r_A + r_B}, \\ \xi = \frac{r_A + r_B}{2d} = \frac{2r \cos \theta}{r_A - r_B} \end{cases} \quad (\text{A'.29})$$

Στο σφαιρικό όριο $d \rightarrow 0 \Rightarrow r_A, r_B \rightarrow r$ οπότε

$$\Rightarrow \begin{cases} \eta \xrightarrow{d \rightarrow 0} 2 \cos \theta, \\ \xi d \xrightarrow{d \rightarrow 0} r \end{cases} \quad (\text{A'.30})$$

Η (Α'.19β') για $c \rightarrow 0$ γίνεται :

$$\frac{d}{d\eta} \left((1 - \eta^2) \frac{dH(\eta)}{d\eta} \right) + \left(\lambda - \frac{\mu^2}{1 - \eta^2} \right) H(\eta) = 0 \quad (\text{A'.31})$$

που είναι η συναφής διαφορική εξίσωση Legendre, με $\lambda = \kappa(\kappa + 1)$ και λύση την $H(\eta) = C_1 \mathcal{P}_\lambda^\mu(\eta) + C_2 \mathcal{Q}_\lambda^\mu(\eta)$, όπου $\mathcal{P}_\lambda^\mu, \mathcal{Q}_\lambda^\mu$: τα συναφή πολυώνυμα Legendre 1^{ου} και 2^{ου} είδους, αντίστοιχα. Άλλωστε η ανάπτυξη των γωνιακών σφαιροειδών συναρτήσεων σε πολυώνυμα Legendre βασίζεται στην παραπάνω αναγωγή.

Η (Α'19α') για $c \rightarrow 0$ γίνεται : Κάνοντας την αλλαγή μεταβλητής $z = c\xi$ και θέτοντας $g(\xi) = (\xi^2 - 1)^{\frac{\mu}{2}} f(z)$ έχουμε

$$(z^2 - c^2) \frac{d^2 f(z)}{dz^2} - 2(\mu + 1)z \frac{df(z)}{dz} + [z^2 + \mu(\mu + 1) - \lambda] f(z) = 0$$

$$\xrightarrow{c \rightarrow 0} z^2 \frac{d^2 f(z)}{dz^2} + 2(\mu + 1)z \frac{df(z)}{dz} + [z^2 + \mu(\mu + 1) - \lambda] f(z) = 0 \quad (\text{Α'.32})$$

και αντικαθιστώντας στην παραπάνω $f(z) = \frac{g(z)}{z^\mu}$ παίρνουμε τελικά

$$z^2 \frac{d^2 g(z)}{dz^2} + 2z \frac{dg(z)}{dz} + (z^2 - \lambda) g(z) = 0 \quad (\text{Α'.33})$$

Η (Α'.33) είναι η σφαιρική διαφορική εξίσωση *Bessel* με $\lambda = \nu(\nu + 1)$ και λύση την

$$\Xi(z) = C_1 j_\nu(z) + C_2 y_\nu(z) \quad (\text{Α'.34})$$

όπου j_ν, y_ν οι σφαιρικές συναρτήσεις *Bessel* οι οποίες αναπτύσσονται σε σειρά ως

$$j_\nu(z) = \sqrt{\frac{\pi}{2z}} \sum_{k=0}^{\infty} \frac{(-1)^k}{k!(k + \nu + \frac{1}{2})} \left(\frac{z}{2}\right)^{2k + \nu + \frac{1}{2}} \quad (\text{Α'.35α'})$$

$$y_\nu(z) = -\frac{(2\nu - 1)!!}{x^{\nu+1}} \sum_{k=0}^{\infty} \frac{(-1)^k}{k! \left(\frac{1}{2} - \nu\right)_k} \left(\frac{z}{2}\right)^{2k} \quad (\text{Α'.35β'})$$

ή

$$j_\nu(z) = \sqrt{\frac{\pi}{2z}} J_{\nu + \frac{1}{2}}(z) \quad (\text{Α'.36α'})$$

$$y_\nu(z) = \sqrt{\frac{\pi}{2z}} Y_{\nu + \frac{1}{2}}(z) \quad (\text{Α'.36β'})$$

όπου $J_{\nu + \frac{1}{2}}, Y_{\nu + \frac{1}{2}}$: οι συναρτήσεις *Bessel* πρώτου και δευτέρου είδους (ή *Neumann*) αντίστοιχα, που ορίζονται ως

$$J_\nu(z) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!(k + \nu)} \left(\frac{z}{2}\right)^{2k + \nu} \quad (\text{Α'.37α'})$$

$$Y_\nu(z) = \frac{\cos \nu\pi J_\nu(z) - J_{-\nu}(z)}{\sin \nu\pi} \quad (\text{Α'.38})$$

Παράρτημα Β΄

Συναρτήσεις Γενικευμένου Σφαιροειδούς

Β΄.1 Γενικευμένη Διαφορική Εξίσωση Σφαιροειδούς

Β΄.1.1 Εισαγωγή

Η γενική μορφή της διαφορικής εξίσωσης είναι :

$$x(x-x_0)\frac{d^2y}{dx^2} + (a_1 + a_2x)\frac{dy}{dx} + [p^2x(x-x_0) - 2bp(x-x_0) + a_3]y(x) = 0 \quad (\text{B'.1})$$

όπου a_1, a_2, a_3, p, b και x_0 είναι σταθερές, με διάστημα ορισμού τα $[0 \leq x \leq x_0]$ και $[x_0 \leq x < \infty)$. Η παραπάνω εξίσωση είναι μια δευτέρας τάξεως γραμμική διαφορική εξίσωση, τετάρτου βαθμού¹ με δύο κανονικά

¹ Ο βαθμός ορίζεται ως εξής :

Ορισμός Β΄.1.1.1. Βαθμός ενός τυπικού όρου $x^m y^{(n)}$ μίας διαφορικής εξίσωσης με πολυωνυμικούς συντελεστές ονομάζεται n διαφορά $d = m - n$.

Ο βαθμός ισούται δηλαδή με τον εκθέτη της δύναμης που πολλαπλασιάζει την παράγωγο μείον την τάξη της παραγώγου[14].

ανώμαλα σημεία και ένα ουσιώδες ανώμαλο σημείο² $x = 0$, $x = x_0$ και $x = \infty$, αντίστοιχα.

Αναγωγή στην (Β'.1) :

Για να καταλήξουμε στη γενική μορφή (Β'.1) από τις εξισώσεις (Α'.19α') και (Α'.19β') (παρ. Α', σελ.89),

$$\frac{d}{d\xi} \left((\xi^2 - 1) \frac{d\Xi(\xi)}{d\xi} \right) - \left(\lambda - \alpha c \xi + c^2 \xi^2 + \frac{m^2}{\xi^2 - 1} \right) \Xi(\xi) = 0 \quad (\text{Β'.2α'})$$

$$\frac{d}{d\eta} \left((1 - \eta^2) \frac{dH(\eta)}{d\eta} \right) + \left(\lambda - \beta c \eta - c^2 \eta^2 - \frac{m^2}{1 - \eta^2} \right) H(\eta) = 0 \quad (\text{Β'.2β'})$$

αρχικά γράφουμε $\Xi = (\xi^2 - 1)^{\frac{m}{2}} f(\xi)$ και $H = (1 - \eta^2)^{\frac{m}{2}} g(\eta)$, ώστε να απομακρύνουμε τους "προβληματικούς όρους" $\frac{m^2}{1 - \eta^2}$ και $\frac{m^2}{\xi^2 - 1}$ και να μειώσουμε το βαθμό της εξίσωσης κατά 1, οπότε έχουμε :

$$(\xi^2 - 1) \frac{d^2 f}{d\xi^2} + 2(m + 1)\xi \frac{df}{d\xi} + [c^2 \xi^2 + \alpha c \xi + m(m + 1) - \lambda] f(\xi) = 0 \quad (\text{Β'.3})$$

και

$$(1 - \eta^2) \frac{d^2 g}{d\eta^2} - 2(m + 1)\eta \frac{dg}{d\eta} - [c^2 \eta^2 + \beta c \eta + m(m + 1) - \lambda] g(\eta) = 0 \quad (\text{Β'.4})$$

² Στο $x = \infty$ έχουμε "Essential singularity" [15].

³ Η αντικατάσταση αυτή, βέβαια, δεν είναι αυθαίρετη αλλά πηγάζει από την απαίτηση η κυματοσυνάρτηση να είναι παντού πεπερασμένη. Έτσι, π.χ. για την γωνιακή εξίσωση, η συνάρτηση που την ικανοποιεί θα πρέπει να είναι πεπερασμένη και στους πόλους $\eta_0 = \pm 1$. Άρα θα πρέπει να μηδενίζεται εκεί ως $(1 \pm \eta)$ στα $\eta_0 = \mp 1$ αντίστοιχα, υψωμένα σε έναν κατάλληλο εκθέτη (κοινό λόγω συμμετρίας)[13]. Αντικαθιστώντας την $(1 - \eta^2)^s$ στην (Β'.2β') βρίσκουμε ότι $s = \pm \frac{|m|}{2}$, εκ των οποίων κρατάμε μόνο το $s = +\frac{|m|}{2}$, εφόσον έχουμε απειρισμό για $s = -\frac{|m|}{2}$ στα $\eta_0 = \mp 1$.

Για να καταλήξουμε στην (Β'.1) τοποθετούμε $x = \xi + 1$ στην (Β'.2α'), και $x = \eta + 1$ στην (Β'.2β') :

$$x(x-2)\frac{d^2f}{dx^2} + 2(m+1)(x-1)\frac{df}{dx} + [c^2x(x-2) + \alpha c(x-2) + c^2 + \alpha c + m(m+1) - \lambda]f(x) = 0 \quad (\text{Β'.5})$$

$$x(x-2)\frac{d^2g}{dx^2} + 2(m+1)(x-1)\frac{dg}{dx} + [c^2x(x-2) + \beta c(x-2) + c^2 + \beta c + m(m+1) - \lambda]g(x) = 0 \quad (\text{Β'.6})$$

Β'.1.2 Επίλυση εξίσωσης σφαιροειδούς μέσω δυναμοσειρών

Εάν εκφράσουμε τις λύσεις της (Β'.1) κοντά στους πόλους⁴ $x = 0$ και $x = x_0$ ως :

$$\lim_{x \rightarrow 0} y(x) = x^{k_1} \quad \text{και} \quad \lim_{x \rightarrow x_0} y(x) = (x - x_0)^{k_2} \quad (\text{Β'.7})$$

και αντικαταστήσουμε σ'αυτήν παίρνουμε $k_1 = 0, 1 + \frac{a_1}{x_0}$ και $k_2 = 0, 1 - a_2 - \frac{a_1}{x_0}$, ενώ συγκεκριμένα για τις (Β'.5) και (Β'.6) οι δείκτες γίνονται $k_1 = 0, -m$ και $k_2 = 0, -m$.

Μελετούμε τη συμπεριφορά της λύσης καθώς $x \rightarrow \infty$, με απώτερο σκοπό να μειώσουμε περεταίρω το βαθμό της διαφορικής εξίσωσης (Β'.1), από 4 σε 3, ώστε να καταλήξουμε τελικά σε μια αναδρομική σχέση τριών όρων.

$$\begin{aligned} (\text{Β'.1}) \xrightarrow{x \rightarrow \infty} x^2 \frac{d^2 y_\infty}{dx^2} + p^2 x^2 y_\infty(x) = 0 &\Rightarrow \\ \frac{d^2 y_\infty}{dx^2} + p^2 y_\infty(x) = 0 &\Rightarrow y_\infty(x) = e^{\pm ipx} \end{aligned} \quad (\text{Β'.8})$$

Συνεπώς, η λύση θα είναι της μορφής $y(x) = e^{\pm ipx} f(x)$, όπου $f(x)$ θα είναι κάποιο πολυώνυμο με άγνωστους συντελεστές, ενώ οι ρίζες του

⁴ Το σημείο α λέγεται **πόλος τάξης n** της συνάρτησης $f(x)$ αν $f(x) = \frac{g(x)}{(x-\alpha)^n}$ με $n \in \mathbb{N}^+$ και $g(x) \neq 0$ [15].

θα μας δίνουν τους κόμβους της κυματοσυνάρτησης. Αναπτύσσουμε τη λύση σε δυναμοσειρά γύρω από το $x = 0^5$ για $k_1 = k_2 = 0$

$$y(x) = e^{ipx} \sum_{n=0}^{\infty} c_n x^n \quad (\text{B'.9})$$

η οποία θα πρέπει να ικανοποιεί την (B'.1), οπότε μετά από κάποιες πράξεις:

$$\begin{aligned} & \sum_{n=0}^{\infty} \{ [c_n n(n-1) - 2ix_0 + ia_1 p + a_2 n + 2bpx_0 + a_3] \\ & + a_{n-1} [2ip(n-1) + ia_2 p - 2bp] \\ & + a_{n+1} [a_1(n+1) - x_0 n(n+1)] \} x^n = 0 \Rightarrow \end{aligned} \quad (\text{B'.10})$$

$$\Rightarrow \begin{cases} \alpha_0 c_1 + \beta_0 c_0 = 0, \\ \alpha_n c_{n+1} + \beta_n c_n + \gamma_n c_{n-1} = 0 \end{cases} \quad (\text{B'.11})$$

με $n = 1, 2, \dots$

Καταλήξαμε σε μια αναδρομική σχέση τριών όρων (B'.11) όπου :

$$\alpha_n = c_1 + (a_1 - x_0)n - x_0 n^2 \quad (\text{B'.12α'})$$

$$\beta_n = 2bpx_0 + ipa_1 + a_3 + (a_2 - 2ipx_0 - 1)n + n^2 \quad (\text{B'.12β'})$$

$$\gamma_n = ip(a_2 - 2) - 2bp + 2ipn \quad (\text{B'.12γ'})$$

Συνεχόμενα Κλάσματα

Η (B'.11), όπως όλες οι αναδρομικές σχέσεις τριών όρων, έχει δύο ανεξάρτητα σύνολα λύσεων $\{a_n : n = 1, 2, \dots\}$ και $\{b_n : n = 1, 2, \dots\}$. Άν επιπλέον, οι λύσεις κατέχουν και την ιδιότητα $\lim_{n \rightarrow \infty} \frac{a_n}{b_n} = 0$, τότε το σύνολο $\{a_n : n = 1, 2, \dots\}$ αναφέρεται ως “ελάχιστο” (minimal set) καθώς $n \rightarrow \infty$, ενώ κάθε μί ελάχιστο σύνολο (nonminimal set) $\{b_n : n = 1, 2, \dots\}$ ως “κυρίαρχο” (dominant set)⁶.

⁵ Κατ' αντιστοιχία μπορούμε να αναπτύξουμε τη λύση γύρω από το $x = x_0$ ως $y(x) = e^{ipx} \sum_{n=0}^{\infty} c_n (x - x_0)^n$.

⁶ Τα κυρίαρχα σύνολα λύσεων δεν είναι μοναδικά καθώς κάθε πολλαπλάσιο της “ελάχιστης” λύσης μπορεί να προστεθεί σε αυτά, χωρίς να καταστρέψει την “κυρίαρχη” ιδιότητα[21].

Για να αναλύσουμε τη συμπεριφορά των συντελεστών ανάπτυξης $\{c_n : n = 1, 2, \dots\}$, σε μεγάλες τιμές του n γράφουμε την (Β'.11) ως

$$\alpha_n \frac{c_{n+1}}{c_n} + \beta_n + \gamma_n \frac{c_{n-1}}{c_n} = 0, \quad (\text{B'.13})$$

διαιρούμε με n^2 και κρατάμε τους κυρίαρχους όρους :

$$-x_0 \frac{c_{n+1}}{c_n} + 1 + \frac{2ip}{n} \frac{c_{n-1}}{c_n} \approx 0. \quad (\text{B'.14})$$

Βλέπουμε πώς εάν $\lim_{n \rightarrow \infty} \frac{c_n}{c_{n-1}} \sim \frac{2ip}{n}$ τότε τα c_n είναι μελη του ελάχιστου συνόλου, ενώ εάν $\lim_{n \rightarrow \infty} \frac{c_n}{c_{n-1}} = \frac{1}{x_0}$ είναι μελη του κυρίαρχου⁷.

Τέλος, εκφράζουμε το λόγο των στοιχείων της ελάχιστης λύσης με τη μέθοδο των συνεχόμενων κλασμάτων :

$$\frac{c_{n+1}}{c_n} = \frac{-\gamma_{n+1}}{\beta_{n+1} - \frac{\alpha_{n+1}\gamma_{n+2}}{\beta_{n+2} - \frac{\alpha_{n+2}\gamma_{n+3}}{\beta_{n+3} - \dots}}} \quad (\text{B'.15})$$

η οποία για $n = 0$ δίνει

$$\frac{c_1}{c_0} = \frac{-\gamma_1}{\beta_1 - \frac{\alpha_1\gamma_2}{\beta_2 - \frac{\alpha_2\gamma_3}{\beta_3 - \dots}}} \quad (\text{B'.16})$$

Όμως από τη (Β'.11) έχουμε :

$$\frac{c_1}{c_0} = -\frac{\beta_0}{\alpha_0} \quad (\text{B'.17})$$

Προφανώς οι εξισώσεις (Β'.16) και (Β'.17) δεν μπορούν να ικανοποιούνται ταυτόχρονα για αυθαίρετες τιμές των $\alpha_n, \beta_n, \gamma_n$, ώστε να η λύση να είναι κυρίαρχο σύνολο, οπότε επιλέγουμε μια τιμή για το c_0 . Η δυναμοσειρά (Β'.9) συγκλίνει $\forall |x| < |x_0|$, αλλά αποκλίνει $\forall |x| \geq |x_0|$.

⁷ Εφόσον στη 1^η περίπτωση το όριο μηδενίζεται, ενώ στη 2^η όχι, σύμφωνα με την παραπάνω ιδιότητα.

Άν θέλουμε να μελετήσουμε τη συμπεριφορά της λύσης κοντά στο $x = x_0$, στο διάστημα $[x_0 \leq x < \infty)$, κάνουμε την αλλαγή μεταβλητής $z = x - x_0$ ⁸, οπότε η εξίσωση μετασχηματίζεται στην

$$z(z + x_0) \frac{d^2 y}{dz^2} + (a_1 + a_2 x_0 + a_2 z) \frac{dy}{dz} + [p^2 z(z + x_0) - 2bpz + a_3]y(z) = 0 \quad (\text{B'.18})$$

που είναι παρόμοια με την (B'.1), οπότε αναπτύσσουμε την λύση σε δυναμοσειρά γύρω από το 0. Δευτερεύουσες λύσεις, στην περίπτωση που τα $1 + \frac{a_1}{x_0}$ ή $1 - a_2 - \frac{a_1}{x_0} \in \mathbb{Z}$, μπορούν να βρεθούν με τη μέθοδο του Frobenius.

B'.1.3 Οι λύσεις των Hylleraas και Jaffé

Η ακτινική εξίσωση (B'.5) ορίζεται στο διάστημα $[x_0 \leq x < \infty)$, με $x_0 = 1$. Πρώτος ο Hylleraas[35], ανέπτυξε τη λύση $y(x)$ γύρω απ' το $x = x_0$ ⁹ σε σειρά πολυωνύμων Laguerre¹⁰ :

$$y_H(x) = e^{ipx} \sum_{n=0}^{\infty} \frac{n!c_n}{\Gamma(\frac{1}{2}a_2 + ib + \frac{a_1}{x_0} + 1 + n)} L_n^{a_2 + \frac{a_1}{x_0} - 1}(-2ip(x - x_0)) \quad (\text{B'.19})$$

όπου Γ : η συνάρτηση γάμμα, L_n, L_n^k : τα πολυώνυμα *Laguerre* και τα συναφή πολυώνυμα *Laguerre*, αντίστοιχα, που ορίζονται [10] ως

$$L_n(x) = \sum_{k=0}^n \frac{(-1)^k}{k!} \binom{n}{k} x^k, \quad (\text{B'.20α'})$$

$$L_n^k(x) = \sum_{m=0}^n (-1)^m \frac{(n+k)!}{(n-m)!(k+m)!m!} x^m \quad (\text{B'.20β'})$$

⁸ ώστε όταν $x \rightarrow x_0 \Rightarrow z = 0$

⁹ Το $x = x_0$ είναι κανονικό ανώμαλο σημείο[15] της (B'.1) εφόσον οι συναρτήσεις $(x - x_0)P(x) = \frac{a_1 + a_2 x}{x}$ και $(x - x_0)^2 Q(x) = (x - x_0) \frac{p^2 x(x - x_0) - 2bp(x - x_0) + a_3}{x}$ είναι αναλυτικές σ' αυτό.

¹⁰ Ο Hylleraas χρησιμοποίησε ως Ansatzes ιδιοσυναρτήσεις του ατόμου του **H**, όπου η ακτινική εξίσωση έχει ως αποδεκτές λύσεις τα συναφή πολυώνυμα Laguerre και η γωνιακή τις σφαιρικές αρμονικές[5]. Αυτή είναι μια λογική επιλογή εφόσον, όπως έδειξα παραπάνω, η γενικευμένη διαφορική εξίσωση σφαιροειδούς προκύπτει (και από την εξίσωση Schrödinger για το μοριακό ιόν του **H**. Συνεπώς και η λύση της θα σχετίζεται με ένα γραμμικό συνδυασμό των λύσεων του ατόμου του **H**.

όπου $\binom{n}{k} = \frac{n!}{(n-k)!k!}$: ο διωνυμικός συντελεστής και τέλος $c_n^{rad} \equiv c_n$: οι συντελεστές ανάπτυξης της ακτινικής συνάρτησης.

Αργότερα ο Jaffé[24] έκανε την αναδιάταξη των ιδιομορφων σημείων $0, x_0$ και ∞ κάνοντας την αλλαγή μεταβλητής¹¹ $z = \frac{x-x_0}{x}$, οπότε η (B'.1) γίνεται :

$$z(1-z)^2 \frac{d^2 y}{dz^2} + \left[a_1 \frac{(1-z)^2}{x_0} + a_2(1-z) \right] \frac{dy}{dz} + \left[p^2 x_0^2 \frac{z}{1-z^2} - 2bp x_0 \frac{z}{1-z} + a_3 \right] y(z) = 0 \quad (B'.21)$$

Εν συνεχεία εκφράζοντας τη λύση ως $y(x) = e^{ipx} x^{-\frac{1}{2}a_2 - ib} g(z)$ και αντικαθιστώντας στην (B'.21) έλαβε :

$$z(1-z)^2 \frac{d^2 g}{dz^2} + (\kappa_1 + \kappa_2 z + \kappa_3 z^2) \frac{dg}{dz} + (\kappa_4 - \kappa_5 z) g(z) = 0 \quad (B'.22)$$

με

$$\kappa_1 = a_2 + \frac{a_1}{x_0} \quad (B'.23\alpha')$$

$$\kappa_2 = -2[\kappa_1 + 1 + i(b - px_0)] \quad (B'.23\beta')$$

$$\kappa_3 = \kappa_1 + 2(1 + ib) \quad (B'.23\gamma')$$

$$\kappa_4 = -\kappa_5 - \frac{a_2}{2} \left(\frac{a_2}{2} - 1 \right) + b(i - b) + ipx_0 \kappa_1 + a_3 \quad (B'.23\delta')$$

$$\kappa_5 = \left(\frac{a_2}{2} + ib \right) \left(\frac{a_2}{2} + ib + 1 + \frac{a_1}{x_0} \right) \quad (B'.23\epsilon')$$

Τέλος, αναπτύσσεται η συνάρτηση $g(z)$ σε δυναμοσειρά γύρω από το 0 , $g(z) = \sum_{n=0}^{\infty} c_n z^n$, οπότε :

$$y_J(x) = e^{ipx} x^{-\frac{1}{2}a_2 - ib} \sum_{n=0}^{\infty} c_n \left(\frac{x-x_0}{x} \right)^n \quad (B'.24)$$

¹¹ Με αυτό το τρόπο ο Jaffé απέφυγε το πρόβλημα της σύγκλισης της δυναμοσειράς (εξ' αιτίας του 0) μεταξύ x_0 και ∞ . Έτσι, δεδομένου ότι η ακτίνα σύγκλισης, της δυναμοσειράς-λύσης μιας διαφορικής εξίσωσης γύρω από ένα κανονικό ανώμαλο σημείο, είναι ίση με την απόσταση από το σημείο αυτό έως το κοντινότερο ιδιόμορφο σημείο[14] (στη προκειμένη περίπτωση το 0) $|x - x'_0| = R$, "μετέφερε" το $x = x_0$ στο 0 (εφόσον $\lim_{x \rightarrow x_0} z = 0$) και το ∞ στο 1 (εφόσον $\lim_{x \rightarrow \infty} z = 1$).

Θέτοντας την (Β'.24) στην (Β'.22), που είναι τριβάθμια, καταλήγουμε στην αναδρομική σχέση τριών όρων¹² :

$$\Rightarrow \begin{cases} \alpha_0 c_1 + \beta_0 c_0 = 0, \\ \alpha_n c_{n+1} + \beta_n c_n + \gamma_n c_{n-1} = 0 \end{cases} \quad (\text{Β'.25})$$

με $n = 1, 2, \dots$ και

$$\alpha_n = (n+1) \left(n + a_2 + \frac{a_1}{x_0} \right) \quad (\text{Β'.26α'})$$

$$\beta_n = -2n^2 - 2 \left[a_2 + i(b - px_0) + \frac{a_1}{x_0} \right] n - \left(\frac{a_2}{2} + ib \right) \left(a_2 + \frac{a_1}{x_0} \right) + ip(a_1 + a_2 x_0) + a_3 \quad (\text{Β'.26β'})$$

$$\gamma_n = \left(n - 1 + \frac{a_2}{2} + ib \right) \left(n + \frac{a_2}{2} + ib + \frac{a_1}{x_0} \right) \quad (\text{Β'.26γ'})$$

Κατόπιν, εξετάζεται η σύγκλιση των (Β'.19) και (Β'.24) εφαρμόζοντας το όριο του λόγου δύο διαδοδοχικών συντελεστών¹³ για μεγάλα n . Έτσι, διαιρώντας την (Β'.25) με $n^2 c_n$, κρατώντας μόνο τους όρους $O(\frac{1}{n})$ έχουμε:

$$\lim_{n \rightarrow \infty} \frac{c_{n+1}}{c_n} = 1 + \frac{a}{\sqrt{n}} + \frac{b}{n} + \dots \quad (\text{Β'.27})$$

και μετά από πράξεις[21] καταλήγουμε στην :

$$\lim_{n \rightarrow \infty} a_n \approx n^u e^{2w \sqrt{n}} \quad (\text{Β'.28})$$

με $w^2 = -2ipx_0$ και $u = i(b - px_0) - \frac{3}{4}$. Αποδεικνύεται ότι για συστήματα όπως το H_2^+ η ανάπτυξη κατά Jaffé (Β'.24) είναι καταλληλότερη απ'ότι του Hylleraas (Β'.19), ενώ η ενέργεια ορίζεται ως $E = -\frac{\sigma^2}{2w}$, με $\sigma = -ip$.

¹² Οι συντελεστές c_n που προκύπτουν από τις (Β'.19) και (Β'.24) ταυτίζονται, οπότε υπακούουν και στην ίδια αναδρομική σχέση.

¹³ Για να συγκλίνει η σειρά το όριο $R = \lim_{n \rightarrow \infty} \left| \frac{c_n}{c_{n+1}} \right|$, πρέπει να υπάρχει και να είναι πεπερασμένο ή άπειρο, ενώ αν είναι μηδέν θα αποκλίνει. Η τιμή του παραπάνω ορίου ορίζει και την ακτίνα σύγκλισης[14].

Β.2 Ακτινική Εξίσωση

Η εξίσωση (Β.2α'),

$$\frac{d}{d\xi} \left((\xi^2 - 1) \frac{d\Xi(\xi)}{d\xi} \right) - \left(\lambda - \alpha c \xi + c^2 \xi^2 + \frac{m^2}{\xi^2 - 1} \right) \Xi(\xi) = 0$$

έχει λύση τη $\Xi(\xi) = (\xi^2 - 1)^{\frac{m}{2}} y_J(\xi)$, που βασίζεται στη δυναμοσειρά του Jaffé (Β.24), ενώ με τις κατάλληλες αντικαταστάσεις, $x = \xi + 1$, $x_0 = 2 \rightarrow \frac{x - x_0}{x} = \frac{\xi - 1}{\xi + 1}$, $a_2 = 2(m + 1)$, $b = \frac{\alpha}{2}$ και $c = -ip$, παίρνει τη μορφή:

$$\Xi(\xi) = (\xi^2 - 1)^{\frac{m}{2}} e^{-c\xi} (\xi + 1)^{\frac{\alpha}{2} - (m+1)} \sum_{n=0}^{\infty} g_n \left(\frac{\xi - 1}{\xi + 1} \right)^n \quad (\text{Β.29})$$

Β.3 Γωνιακή Εξίσωση

Η εξίσωση (Β.2β')

$$\frac{d}{d\eta} \left((1 - \eta^2) \frac{dH(\eta)}{d\eta} \right) + \left(\lambda - \beta c \eta + c^2 \eta^2 - \frac{m^2}{1 - \eta^2} \right) H(\eta) = 0$$

- Για $\beta = 0$ (δηλ. $Z_A = Z_B$) ανάγεται στην (απλή) εξίσωση σφαιροειδούς

$$\frac{d}{d\eta} \left((1 - \eta^2) \frac{dH(\eta)}{d\eta} \right) + \left(\lambda + c^2 \eta^2 - \frac{m^2}{1 - \eta^2} \right) H(\eta) = 0 \quad (\text{Β.30})$$

Παρατηρούμε πώς για $c = 0$ η (Β.30) γίνεται η συναφής εξίσωση Legendre (βλ. παρ. Α.3.1, σελ. 93) :

$$\frac{d}{d\eta} \left((1 - \eta^2) \frac{dH(\eta)}{d\eta} \right) + \left(\lambda_\kappa - \frac{m^2}{1 - \eta^2} \right) H(\eta) = 0 \quad (\text{Β.31})$$

Η παραπάνω ικανοποιείται από τα συναφή πολυώνυμα Legendre \mathcal{P}_l^m , \mathcal{Q}_l^m με $\lambda_\kappa = \kappa(\kappa + 1)$, τα οποία σχηματίζουν μια κατάλληλη βάση ώστε να αναπτύξουμε (σε γραμμικό συνδυασμό) τις γωνιακές σφαιροειδείς αρμονικές συναρτήσεις $S_l^m(c; \eta)$:

$${}^{(1)}S_l^m(c; \eta) = \sum_{j=0,1}^{\infty} 'd_j^{lm}(c) \mathcal{P}_{l+j}^m(\eta), \quad (\text{Β.32α}')$$

$${}^{(2)}S_l^m(c; \eta) = \sum_{j=-\infty}^{\infty} 'd_j^{lm}(c) \mathcal{Q}_{l+j}^m(\eta) \quad (\text{Β.32β}')$$

Τα $^{(1,2)}S_l^m(c; \eta)$ είναι οι γωνιακές σφαιροειδείς συναρτήσεις 1^{ου} και 2^{ου} είδους, αντίστοιχα. Ο τόνος στο άθροισμα υποδηλώνει ότι η άθροιση γίνεται μόνο πάνω στις άρτιες τιμές του j όταν $l-m = 2k$ (άρτιο) και μόνο πάνω στις περιττές τιμές του j όταν $l-m = 2k+1$ (περιττό).

Για να καθοριστούν πλήρως οι σφαιροειδείς συναρτήσεις επιβάλλονται οι παρακάτω οριακές συνθήκες[34] :

$$S_l^m(c; 0) = \begin{cases} \frac{(-1)^{\frac{l-m}{2}}(l+m)!}{2^{l-m} \left(\frac{l-m}{2}\right)! \left(\frac{l+m}{2}\right)!}, & (l-m) = 2k \\ 0, & (l-m) = 2k+1 \end{cases} \quad (\text{B'.33})$$

- Για $\eta \rightarrow \infty$ και d (ή R) $\rightarrow 0$ η (B'.30) γίνεται :

$$\begin{aligned} (\text{B'.30}) \xrightarrow{\eta \rightarrow \infty} \eta^2 \frac{d^2 H_\infty}{d\eta^2} + c^2 \eta^2 H_\infty(\eta) = 0 \Rightarrow \\ \frac{d^2 H_\infty}{d\eta^2} + c^2 H_\infty(\eta) = 0 \Rightarrow H_\infty(\eta) = e^{\pm ic\eta} \end{aligned} \quad (\text{B'.34})$$

Άρα τελικά η λύση για μικρά R είναι της μορφής :

$$S_l^m(c; \eta) = e^{\pm ic\eta} \sum_{j=0}^{\infty} d_j^{lm}(c) \mathcal{P}_{l+j}^m(\mp \eta), \quad \text{ή} \quad (\text{B'.35α'})$$

$$S_l^m(-ic; \eta) = e^{\mp c\eta} \sum_{j=0}^{\infty} d_j^{lm}(-ic) \mathcal{P}_{l+j}^m(\pm \eta) \quad (\text{B'.35β'})$$

οπου στη τελευταία έκανα χρήση του μετασχηματισμού $c \rightarrow -ic$ από επιμήκεις σε πεπλατυσμένες σφαιροειδείς συντεταγμένες.

- Για μεγάλα R η λύση της (B'.2β') γράφεται ως :

$$S_l^m(c; \eta) = (1 - \eta^2)^{\frac{m}{2}} \sum_{j=0}^{\infty} 'A_j^{lm}(c) \eta^j, \quad \text{ή} \quad (\text{B'.36α'})$$

$$S_l^m(-ic; \eta) = (1 - \eta^2)^{\frac{m}{2}} e^{\pm c\eta} \sum_{j=0}^{\infty} 'B_j^{lm}(-ic) (1 \mp \eta)^j \quad (\text{B'.36β'})$$

όπου οι $A_j^{lm}(c)$ και $B_j^{lm}(-ic)$ υπακούουν στις παρακάτω αναδρομικές σχέσεις[34] :

$$(j+1)(j+2)A_{j+2}^{lm}(c) - [j(j+2m+1) + m(m+1) - \lambda_\kappa(c)]A_j^{lm}(c) - c^2A_{j-2}^{lm}(c) = 0, \quad (\text{B'.37}\alpha')$$

$$2(j+1)(j+m+1)B_{j+1}^{lm}(-ic) - [j(j+2m+1+4c) + (m+1)(m+2c) - \lambda_\kappa(-ic) - c^2]B_j^{lm}(-ic) + 2(l+m)cB_{j-1}^{lm}(-ic) = 0, \quad (\text{B'.37}\beta')$$

με

$$\lim_{j \rightarrow \infty} \frac{A_{j+2}^{lm}(c)}{A_j^{lm}(c)} = 0 \quad \text{και} \quad \lim_{j \rightarrow \infty} \frac{B_{j+1}^{lm}(-ic)}{B_j^{lm}(-ic)} = 0 \quad (\text{B'.38})$$

Μέρος V
FORTRAN Subroutines

Παράρτημα Γ΄

Κεντρικά Προγράμματα

Γ.1 Κύριο πρόγραμμα

```
program Energy
implicit real*8 (a-h,o-z)
parameter (nelecp=7,np=2*nelecp,norbp=10,norbpE=10,aa0=0.d0,
1 aa1=1.d0,nlowerp=10,nconfp=10,kp=10) !physical dimension. (You can
c integer aip,ajq
parameter(n0=0,nFac=29)
real*8 ag(0:50),af(0:50)
dimension zza(norbp),zzb(norbp),NN(norbp),LL(norbp),MM(norbp)
dimension pp(2*norbp),xx(2*norbp,2*norbp)
dimension zamin(norbp),zbmin(norbp),zamax(norbp),zbmax(norbp)
character*3 ffile
common/cpcqaipajq/cp(nconfp),cq(nconfp,0:nlowerp),
1 aip(kp,nconfp),ajq(kp,nconfp,0:nlowerp),nconfE,nconf(0:nlowerp),
1 kE(kp),k(kp,0:nlowerp)
common/cus/isl(nelecp,kp,nconfp),isr(nelecp,kp,nconfp,0:nlowerp),
1 iul(nelecp,kp,nconfp),iur(nelecp,kp,nconfp,0:nlowerp)
common/czszdfgnfng/zs(norbp,0:nlowerp),zd(norbp,0:nlowerp),
1 Rjump(norbp,0:nlowerp),p(norbp,0:nlowerp),
```



```

1 f(0:50,norbp,0:nlowerp),g(0:50,norbp,0:nlowerp),
1 nf(norbp,0:nlowerp),ng(norbp,0:nlowerp),
2 N(norbp,0:nlowerp),L(norbp,0:nlowerp),M(norbp,0:nlowerp)
common/czazbG/za(norbp,0:nlowerp),zb(norbp,0:nlowerp)
common/czold/zaold(norbp,0:nlowerp),zbold(norbp,0:nlowerp)

common/cicen/icen(norbp,0:nlowerp),lstart(norbp,0:nlowerp)
!excited
common/czoldE/zaoldE(norbp),zboldE(norbp)!excited
common/czazbE/zaE(norbp),zbE(norbp)!excited
common/cicenE/icenE(norbp),lstartE(norbp)!excited
common/czszdfgnfngE/zsE(norbp),zdE(norbp),RjumpE(norbp),pE(norbp),
1 fE(0:50,norbp),gE(0:50,norbp),nfE(norbp),ngE(norbp),
2 NE(norbp),LE(norbp),ME(norbp)
common/cnorbE/norbE,nlower
common/cnorbG/norb(0:nlowerp)
common/ciu/iu40,iu80
common/choose/smv,ener,ichoice,imchf,jvirt,icrea,jvorb,jvconf
1 ,jvelec
common/cRzpLm/R2, zi,zk,zj,zl2 ,pi,pk,pj,pl2, Lamda,mu
common/cRzpLm1/R1, pl1,pm1, zzs,zzd, zl1,zm1, zzl,zzm
common/cRzpLm0/R, zl,zm, pl,pm
common/cxi/xii,xif
common/ceta/etai,etaf
common/cnelec/nelec
common/cESlower/HFi(0:nlowerp),SFi(0:nlowerp),
1 SqFi(0:nlowerp),EFi(0:nlowerp)
common/cLamda/maLamda
common/com1/ftol,ztol,zmin(2*norbpE),zmax(2*norbpE)
common/cmsgns/msgn1(nelecp,kp,nconfp),
1 msgnr(nelecp,kp,nconfp,0:nlowerp)

```

```
external func
```

```
common /cfac/fac(n0:nfac)
```

```
character ffac*504,c3*200
```

```
write (ffac,'(a)')"1.d0,1.d0,2.d0,6.d0,24.d0,120.d0,  
-720.d0,5040.d0,40320.d0,362880.d0,3.6288d006,3.99168d007,  
-4.790016d008,6.2270208d009,8.71782912d010,1.307674368d012,  
-2.0922789888d013,3.55687428096d014,6.402373705728d015,  
-1.21645100408832d017,2.43290200817664d018,5.109094217170944d019,  
-1.1240007277776077d021,2.585201673888498d022,  
-6.204484017332394d023,1.5511210043330986d025,  
1 4.0329146112660565d026,1.0888869450418352d028,  
1 3.0488834461171387d029,8.841761993739702d030"  
read(ffac,*) fac
```

```
C all angular functions are integrated with qromb_DQDAGS_2
```

```
C all internal radial functions are integrated with qromb_DQDAGS_1
```

```
C all external radial functions are integrated with qromb_DQDAGS
```

```
xii=1.d0
```

```
xif=100.d0
```

```
etai=-1.d0
```

```
etaf= 1.d0
```

```
print *, 'read *,R,Za,Zb,number of electrons,number of state'
```

```
read *,R,ZRa,ZRb,nelec,nlower
```

```
R2=R
```

```
R1=R
maLamda=0

zZs=ZRa+ZRb
zZd=ZRa-ZRb

    icoice=1
open(500)
c ,file='G:\Hector\Hete_L-S\ExpValue\heddiag.txt')
read(500,*) nconfE
do iconf=1,nconfE
read(500,*) kE(iconf)
do idet=1,kE(iconf)
read(500,*) aip(idet,iconf),(iul(ii,idet,iconf),
1 msgnl(ii,idet,iconf),isl(ii,idet,iconf),ii=1,nelec)
end do
end do
close(500)

open(600,file=
c 1 'F:\Ektor\Matrix Elements\Hete_L-S\ExpValue\ezazbnlm.dat')
1 'G:\Hector\Hete_L-S\ExpValue\ezazbnlm.dat')
do jvirt=1,norbp
zaoldE(jvirt)=aa0
zboldE(jvirt)=aa0
read(600,*,err=4) zaE(jvirt),zbE(jvirt),
1 NE(jvirt),LE(jvirt),ME(jvirt),
1 zamin(jvirt),zbmin(jvirt),zamax(jvirt),zbmax(jvirt)
end do
4 norbE=jvirt-1
```

```
do jvirt=1,norbE
if(zaE(jvirt).ne.zaoldE(jvirt).or.zbE(jvirt).ne.zboldE(jvirt))
1 call make_pgf(jvirt)
end do

do ilower=0,nlower
iulower=nlower*1000
open(500+iulower)
c 1 ,file='G:\Hector\Hete_L-S\ExpValue\heddiag.txt')
read(500+iulower,*) nconf(ilower)
do iconf=1,nconf(ilower)
read(500+iulower,*) k(iconf,ilower)
do idet=1,k(iconf,ilower)
read(500+iulower,*) ajq(idet,iconf,ilower),
1 (iur(ii,idet,iconf,ilower),
1 msgnr(ii,idet,iconf,ilower),
1 isr(ii,idet,iconf,ilower),ii=1,nelec)
end do
end do
close(500+iulower)

if (ilower.ne.0) read (iulower,*) SqFi(ilower),EFi(ilower),
1 norb(ilower)
if(ilower.eq.0) norb(ilower)=norbE

c 1'G:\Hector\Hete_L-S\ExpValue\gzazbnlm.dat')
do jvirt=1,norb(ilower)
iu80=ilower*1000+800+jvirt
iu40=ilower*1000+400+jvirt
```

```
read (iu80,*,err=1) za(jvirt,ilower),zb(jvirt,ilower),
1 N(jvirt,ilower),L(jvirt,ilower),M(jvirt,ilower),
1 p(jvirt,ilower),cscr,sigmascr
1 ,Rjump(jvirt,ilower), g(0,jvirt,ilower),g(1,jvirt,ilower)
ag(0)=abs(g(0,jvirt,ilower))
ag(1)=abs(g(1,jvirt,ilower))
do i=2,50
read (iu80,*) iscr,g(i,jvirt,ilower)
ag(i)=abs(g(i,jvirt,ilower))
if (ag(i).lt.1.d-17) go to 2
if (i.ge.3)then
if (ag(i).gt.ag(i-1).and.
1 ag(i-1).lt.ag(i-2).and.ag(i-2).lt.ag(i-3)) goto 2
end if
end do
2 g(i,jvirt,ilower)=aa0
ng(jvirt,ilower)=i-1
close(iu80)

read (iu40,*,err=1)
1 pscr,cscr,mscr,zdifscr,icen(jvirt,ilower),igoscr,
1 lstart(jvirt,ilower),
1 RZscr,INVERSscr,
1 f(0,jvirt,ilower),f(1,jvirt,ilower)
af(0)=abs(f(0,jvirt,ilower))
af(1)=abs(f(1,jvirt,ilower))
do i=2,50
read (iu40,*) iscr,f(i,jvirt,ilower)
af(i)=abs(f(i,jvirt,ilower))
if (af(i).lt.1.d-17.or.af(i).gt.af(1)) go to 3
if (i.ge.3)then
```

```
if (af(i).gt.af(i-1).and.  
1 af(i-1).lt.af(i-2).and.af(i-2).lt.af(i-3)) go to 3  
end if  
end do  
  3 f(i,jvirt,ilower)=aa0  
    nf(jvirt,ilower)=i-1  
close (iu40)  
zs(jvirt,ilower)=za(jvirt,ilower)+zb(jvirt,ilower)  
zd(jvirt,ilower)=za(jvirt,ilower)-zb(jvirt,ilower)  
  
end do !jvirt=1,norbp  
  1 norb(ilower)=jvirt-1  
close (iu80)  
close (iu40)  
enddo !ilower=0,nlower  
  
do i=1,norbE  
do j=1,norbE  
xx(j,i)=aa0  
end do  
end do  
do jvirt=1,norbE  
xx(jvirt,jvirt)=aa1  
pp(2*jvirt-1)=zaE(jvirt)  
pp(2*jvirt)=zbE(jvirt)  
  
zmin(2*jvirt-1)=zamin(jvirt)  
zmin(2*jvirt)=zbmin(jvirt)  
  
zmax(2*jvirt-1)=zamax(jvirt)  
zmax(2*jvirt)=zbmax(jvirt)
```

```

end do
ftol=1.d-9 !9(ftol)=2*5(ztol)-1
ztol=1.d-5
call min_zz(pp,2*norbE,iter,fret,func)

nlower=nlower+1
iulower=nlower*1000
write (iulower,*) sqrt(SFi(0)),HFi(0)/SFi(0),norbE
do jvirt=1,norbE
iu40=400+jvirt
iu80=800+jvirt
iu40out=nlower*1000+400+jvirt
iu80out=nlower*1000+800+jvirt
call readwrite(iu40,iu40out)
call readwrite(iu80,iu80out)
end do

end

```

Γ.2 Βελτιστοποίηση των z^*

```

subroutine min_zz(p,n,iter,fret,func)
c Set the initial variation direction and call powell
implicit real*8(a-h,o-z)
parameter(nph=35)
parameter(i10=10,z0d0=0.d0,z1d0=1.d0,z2d0=2.d0)
dimension p(nph),xi(nph,nph)
common/com1/ftol,ztol,zmin(nph),zmax(nph)
EXTERNAL func

```

```

common/cifail/ifail

c From this routine up to subroutine BRACKET, we modified the Powell
c method to minimize n-dimensional function (c.f. Numerical Recipes
c edition,pp406) under the specific bracketed region
np = np
ctab1a from ctab1a to ctab1b give initial direction xi
do i=1,np
do j =1,np
if(i.eq.j) then
xi(i,j) = z1d0
else
xi(i,j) = z0d0
end if
end do
end do
ctab1b
call ppowell(p,xi,n,np,iter,fret,func)
if (ifail.eq.1) return
write (*,*) 'powell finished'
      write (*,'(<n+1>f21.15)') fret,(p(i),i=1,n)
c print '(a,7f10.5)', 'z,f', (p(i),i=1,n), fret
end

```

```

      SUBROUTINE ppowell(p,xi,n,np,iter,fret,f)
c N dimmesinal bracketed minimization
      implicit real*8(a-h,o-z)
parameter(np=35)
      INTEGER iter,n,np,NMAX,ITMAX
      REAL*8 fret,p(np),xi(np,np)

```



```
EXTERNAL f
parameter (i10=10, z0d0=0.d0, z1d0=1.d0, z2d0=2.d0)
PARAMETER (NMAX=nph, ITMAX=200)
INTEGER i, ibig, j
REAL*8 del, fp, fptt, t, pt (NMAX), ptt (NMAX), xit (NMAX)
common/com1/ftol, ztol, zmin (nph), zmax (nph)
common/cbord/ibord
common/cifail/ifail
ibord=0
fret=f (p)
do 11 j=1, n
pt (j)=p (j)
11 continue
iter=0
1 iter=iter+1
write (*, *) 'iter=', iter
fp=fret
ibig=0
del=z0d0
do 13 i=1, n
do 12 j=1, n
xit (j)=xi (j, i)
12 continue
fptt=fret
c call linmin (p, xit, n, fret)
call lgmin (p, xit, n, f, fret)
if (ifail.eq.1) return
if (ibord.eq.1) return
if (dabs (fptt-fret) .gt. del) then
del=dabs (fptt-fret)
ibig=i
```

```

endif
  13 continue
if(z2d0*dabs(fp-fret).le.ftol*(dabs(fp)+dabs(fret)))return
if(iter.eq.ITMAX) pause 'powell exceeding maximum iterations'
do 14 j=1,n
ptt(j)=z2d0*p(j)-pt(j)
if (ptt(j).lt.zmin(j) ) ptt(j)=zmin(j) !we made this bracketing
if (ptt(j).gt.zmax(j) ) ptt(j)=zmax(j) !we made this bracketing
xit(j)=p(j)-pt(j)
pt(j)=p(j)
  14 continue
fptt=f(ptt)
if(fptt.ge.fp)goto 1
t=z2d0*(fp-z2d0*fret+fptt)*(fp-fret-del)**2-del*(fp-fptt)**2
if(t.ge.z0d0)goto 1
c      call linmin(p,xit,n,fret)
call lgmin(p,xit,n,f,fret)
if (ibord.eq.1) return
do 15 j=1,n
xi(j,ibig)=xi(j,n)
xi(j,n)=xit(j)
  15 continue
      goto 1
      END

```

```

subroutine lgmin(p,xi,n,f,fret)
c Bracketed minimization in a certain direction
implicit real*8(a-h,o-z)
parameter(nph=35)
dimension p(nph),xi(nph),xmin(nph),a(nph), b(nph)

```

```

EXTERNAL f
common/com1/ftol, ztol, zmin(nph), zmax(nph)
common/cifail/ifail
if (ifail.eq.1) return
write (*,*) 'another direction'
call bracket(p, xi, a, b, n)
call ggolden(a, p, b, f, ztol, xmin, n, fret)
do j=1, n
p(j) = xmin(j)
end do
write (*, '(a, <n+1>f21.15)') '--->', fret, (p(i), i=1, n)
c write (26, '( <n+1>f11.5)') (p(i), i=1, n), fret
end

```

```

subroutine ggolden(ax, bx, cx, f, tol, xmin, n, fret)
c N dimensional golden rule search
implicit real*8(a-h, o-z)
parameter(nph=35)
parameter(i10=10, z0d0=0.d0, z1d0=1.d0)
REAL*8 fret, ax(nph), bx(nph), cx(nph), tol, xmin(nph), f, R, C
EXTERNAL f
PARAMETER (R=.61803399d0, C=z1d0-R)
REAL*8 f1, f2, x0(nph), x1(nph), x2(nph), x3(nph)
logical lx1, lx2
common/com1/ftol, ztol, zmin(nph), zmax(nph)
common/cbord/ibord
common/cizeq/izeq
common/cifail/ifail
call addvec(x0, z1d0, ax, z0d0, bx, n) !proceed along this direction
call addvec(x3, z1d0, cx, z0d0, bx, n)

```

```
if (dist (cx, bx, n) .gt. dist (bx, ax, n) ) then
call addvec (x1, z1d0, bx, z0d0, bx, n)
call addvec (x2, z1d0-c, bx, c, cx, n)
else
call addvec (x2, z1d0, bx, z0d0, bx, n)
call addvec (x1, z1d0-c, bx, c, ax, n)
endif
f1=f (x1)
f2=f (x2)
1 if (dist (x3, x0, n) .gt. tol* (dist (x1, ax, n)
1      +dist (x2, ax, n) ) ) then
if (f2.lt.f1) then
call addvec (x0, z1d0, x1, z0d0, ax, n)
call addvec (x1, z1d0, x2, z0d0, ax, n)
call addvec (x2, r, x1, c, x3, n)
f1=f2
f2=f (x2)
else
call addvec (x3, z1d0, x2, z0d0, ax, n)
call addvec (x2, z1d0, x1, z0d0, ax, n)
call addvec (x1, r, x2, c, x0, n)
f2=f1
f1=f (x1)
endif
if (ifail.eq.1) return
ibord= 0
tolx = ztol
if (izeq.eq.1) return ! the equal z
do in=1, n
lx1=abs (x1 (in) -zmin (in) ) .lt. tolx .or. abs (x1 (in) -zmax (in) ) .lt. tolx
lx2=abs (x2 (in) -zmin (in) ) .lt. tolx .or. abs (x2 (in) -zmax (in) ) .lt. tolx
```

```
c print *, lx1, lx2
if(lx1.eq..true..or.lx2.eq..true.) then
ibord=1
print *, in, ' in ggolden goes to the border'
write(26,*) in, ' in ggolden goes to the border'
if (lx1.eq..true.) call addvec(xmin,z1d0,x1,z0d0,ax,n)
if (lx2.eq..true.) call addvec(xmin,z1d0,x2,z0d0,ax,n)
c print *, 'x1, x2',x1,x2,xmin
return
end if
end do
goto 1
endif
if(f1.lt.f2)then
fret = f1
call addvec(xmin,z1d0,x1,z0d0,ax,n)
else
fret = f2
call addvec(xmin,z1d0,x2,z0d0,ax,n)
endif
return
      END
```

```
function dist(x,y,n)
c Distance of two points in n dimensions
implicit real*8(a-h,o-z)
parameter(i10=10,z0d0=0.d0,z1d0=1.d0)
dimension x(n),y(n)
xy2 = z0d0
do i=1,n
```

```
xy2 = xy2+(x(i)-y(i))**2
end do
xy   = dsqrt(xy2)
dist = xy
end
```

```
subroutine addvec(x,a,y,b,z,n)
c Linear combination of two vectors
implicit real*8(a-h,o-z)
dimension x(n),y(n),z(n)
do i=1,n
x(i) = a*y(i) +b*z(i)
end do
end
```

```
subroutine bracket(p,pn,a,b,n)
c calculate the bracket a and b along direction pn passing from p
implicit real*8(a-h,o-z)
parameter(nph=35)
dimension p(nph),pn(nph),a(nph),b(nph),pmin(nph),pmax(nph)
common/com1/ftol,ztol,zmin(nph),zmax(nph)
cmin = 1.d16
cmax = -1.d16
do i=1,n
if(pn(i).gt.1.d-5) then           ! Pn(i) > 0
pmax(i) = (zmin(i)-p(i))/pn(i)
pmin(i) = (zmax(i)-p(i))/pn(i)
else if(pn(i).lt.-1.d-5) then    ! Pn(i) < 0
pmin(i) = (zmin(i)-p(i))/pn(i)
```

```

pmax(i) = (zmax(i)-p(i))/pn(i)
else if(abs(pn(i)).lt.1.d-5) then ! Pn(i) = 0
pmin(i) = cmin
pmax(i) = cmax
end if
cmax = dmax1(cmax,pmax(i))
cmin = dmin1(cmin,pmin(i))
end do
do i= 1,n
a(i) = p(i) + cmax*pn(i)
b(i) = p(i) + cmin*pn(i)
c print '(i3,3f10.5)', i, a(i), p(i), b(i)
end do
end

```

Γ.2.1 Συνάρτηση που καλεί η Γ.2

```

function Func(pp)
implicit real*8 (a-h,o-z)
parameter (nelecp=7,np=2*nelecp,norbp=10,norbpE=10,aa0=0.d0,
1 nlowerp=10,nconfp=10,kp=10) !physical dimension. (You can increase
common/cnorbE/norbE,nlower
dimension pp(2*norbE)

print(' (1p20g15.6) '),pp
call newZ(pp)
call Psi0_PsiK
Func=Functional(nlower)
print*, 'F=',Func
write(1,' (d25.15,1p20g15.6) ') Func,pp
end

```

Παραγωγή OEDM βάσει των νέων z^*

```

subroutine newZ(pp)
implicit real*8(a-h,o-z)
parameter (nelecp=7,np=2*nelecp,norbp=10,norbpe=10,aa0=0.d0,
1 nlowerp=10,nconfp=10,kp=10) !physical dimension. (You can increase
real*8 ag(0:50),af(0:50)
common/cnorbe/norbE,nlower
dimension pp(2*norbE)
common/czazbG/za(norbp,0:nlowerp),zb(norbp,0:nlowerp)

common/czszdfgnfng/zs(norbp,0:nlowerp),zd(norbp,0:nlowerp),
1 Rjump(norbp,0:nlowerp),p(norbp,0:nlowerp),
1 f(0:50,norbp,0:nlowerp),g(0:50,norbp,0:nlowerp),
1 nf(norbp,0:nlowerp),ng(norbp,0:nlowerp),
2 N(norbp,0:nlowerp),L(norbp,0:nlowerp),M(norbp,0:nlowerp)
common/cicen/icen(norbp,0:nlowerp),lstart(norbp,0:nlowerp)
common/czoldE/zaoldE(norbp),zboldE(norbp) !excited
common/czazbE/zaE(norbp),zbE(norbp) !excited
common/cicenE/icenE(norbp),lstartE(norbp) !excited
common/czszdfgnfngE/zsE(norbp),zdE(norbp),RjumpE(norbp),pE(norbp),
1 fE(0:50,norbp),gE(0:50,norbp),nfE(norbp),ngE(norbp),
2 NE(norbp),LE(norbp),ME(norbp)

do jvirt=1,norbE
zaE(jvirt)=pp(2*jvirt-1)
zbE(jvirt)=pp(2*jvirt)
za(jvirt,0)=pp(2*jvirt-1)
zb(jvirt,0)=pp(2*jvirt)
end do

```



```
ilower=0
do jvirt=1,norbE
if (zaE(jvirt).ne.zaoldE(jvirt).or.zbE(jvirt).ne.zboldE(jvirt)) then
call make_pgf(jvirt)

iu80=ilower*1000+800+jvirt
iu40=ilower*1000+400+jvirt
read (iu80,*,err=1) za(jvirt,ilower),zb(jvirt,ilower),
1 N(jvirt,ilower),L(jvirt,ilower),M(jvirt,ilower),
   1 p(jvirt,ilower),cscr,sigmascr
1 ,Rjump(jvirt,ilower), g(0,jvirt,ilower),g(1,jvirt,ilower)
ag(0)=abs(g(0,jvirt,ilower))
ag(1)=abs(g(1,jvirt,ilower))
do i=2,50
read (iu80,*) iscr,g(i,jvirt,ilower)
ag(i)=abs(g(i,jvirt,ilower))
if (ag(i).lt.1.d-17) go to 2
if (i.ge.3) then
if (ag(i).gt.ag(i-1).and.
1 ag(i-1).lt.ag(i-2).and.ag(i-2).lt.ag(i-3)) goto 2
end if
end do
   2 g(i,jvirt,ilower)=aa0
   ng(jvirt,ilower)=i-1
close(iu80)

read (iu40,*,err=1)
1 pscr,cscr,mscr,zdifscr,icen(jvirt,ilower),igoscr,
   1 lstart(jvirt,ilower),
1 RZscr,INVERSscr,
```

```

1 f(0,jvirt,ilower),f(1,jvirt,ilower)
af(0)=abs(f(0,jvirt,ilower))
af(1)=abs(f(1,jvirt,ilower))
  do i=2,50
    read (iu40,*) iscr,f(i,jvirt,ilower)
af(i)=abs(f(i,jvirt,ilower))
if (af(i).lt.1.d-17.or.af(i).gt.af(1)) go to 3
if (i.ge.3)then
if(af(i).gt.af(i-1).and.
1 af(i-1).lt.af(i-2).and.af(i-2).lt.af(i-3))go to 3
end if
end do
  3 f(i,jvirt,ilower)=aa0
    nf(jvirt,ilower)=i-1
close (iu40)
zs(jvirt,ilower)=za(jvirt,ilower)+zb(jvirt,ilower)
zd(jvirt,ilower)=za(jvirt,ilower)-zb(jvirt,ilower)
end if
end do !jvirt=1,norbpE
  1 close (iu80)
close (iu40)

end

```

Παραγωγή των p, g, f

```

subroutine make_pgf(jvirt)
implicit real*8 (a-h,o-z)
parameter (norbp=10,aa0=0.d0)
real*8 ag(0:50),af(0:50)
common/czazbE/zaE(norbp),zbE(norbp) !excited
common/cicenE/icenE(norbp),lstartE(norbp) !excited

```

```

common/czszdfgnfngE/zsE(norbp),zdE(norbp),RjumpE(norbp),pE(norbp),
1 fe(0:50,norbp),gE(0:50,norbp),nfE(norbp),ngE(norbp),
2 NE(norbp),LE(norbp),ME(norbp)
common/ciu/iu40,iu80
common/czoldE/zaoldE(norbp),zboldE(norbp)
common/cRzpLm0/R, z1,zm, pl,pm

```

```

ichoice=1
if (ichoice.eq.1) then
iu40=400+jvirt
iu80=800+jvirt
write (iu80,*) zaE(jvirt),zbE(jvirt),
1 NE(jvirt),LE(jvirt),ME(jvirt)
close (iu80)
zaa=aa0
zbb=aa0
call oedm(zaa,zbb,R)
do i=0,50
gE(i,jvirt)=aa0
end do
read (iu80,*) zaE(jvirt),zbE(jvirt),NE(jvirt),LE(jvirt),
1 ME(jvirt),pE(jvirt),cscr,sigmascr
1 ,RjumpE(jvirt), gE(0,jvirt),gE(1,jvirt)
ag(0)=abs(gE(0,jvirt))
ag(1)=abs(gE(1,jvirt))
do i=2,50
read (iu80,*) iscr,gE(i,jvirt)
ag(i)=abs(gE(i,jvirt))
if (ag(i).lt.1.d-17) go to 2
if (i.ge.3)then
if (ag(i).gt.ag(i-1).and.

```

```
1 ag(i-1).lt.ag(i-2).and.ag(i-2).lt.ag(i-3)) goto 2
end if
end do
  2 gE(i,jvirt)=aa0
    ngE(jvirt)=i-1
close(iu80)
do i=0,50
fE(i,jvirt)=aa0
end do
read(iu40,*)
1 pscr,cscr,mscr,zdifscr,icenE(jvirt),igoscr,lstartE(jvirt),RZscr,
1 INVERSscr,
1 fE(0,jvirt),fE(1,jvirt)
af(0)=abs(fE(0,jvirt))
af(1)=abs(fE(1,jvirt))
do i=2,50
read(iu40,*) iscr,fE(i,jvirt)
af(i)=abs(fE(i,jvirt))
if (af(i).lt.1.d-17.or.af(i).gt.af(1)) go to 3
if (i.ge.3)then
if(af(i).gt.af(i-1).and.
1 af(i-1).lt.af(i-2).and.af(i-2).lt.af(i-3))go to 3
end if
end do
  3 fE(i,jvirt)=aa0
    nfE(jvirt)=i-1
close(iu40)

zsE(jvirt)=zaE(jvirt)+zbE(jvirt)
zdE(jvirt)=zaE(jvirt)-zbE(jvirt)
end if
```

```

c zaoldE(jvirt)=zaE(jvirt)
c zboldE(jvirt)=zbE(jvirt)
return
end

```

Πρόγραμμα Powel

```

subroutine oedm(za, zb, rstart)
C PROGRAM OEDM (ONE-ELECTRON DIATOMIC MOLECULES)
C AUTHOR - JAMES D. POWER, CHEMISTRY DIVISION, NATIONAL RESEARCH
C COUNCIL OF CANADA, OTTAWA, CANADA K1A0R6
      IMPLICIT REAL*8 (A-H, O-Z)
C VARIABLES WHOSE NAMES BEGIN WITH A-H OR O-Z ARE DOUBLE PRECISION
      INTEGER*2 CENTER(2) / ' A ', ' B ' / , EOUT(2) / ' W ', ' E ' /
      INTEGER FCHAIN, GCHAIN
common/choose/smv, ener, ichoice, imchf, jvirt, icrea, jvorb, jvconf
1 , jvelec
common/crjumpoedm/rjump
common/ciu/iu40, iu80

      DATA IIN, IOUT, IPCH/5, 6, 7/
C INPUT IS FROM UNIT 5, PRINTED OUTPUT ON 6, PUNCHED OP ON 7
      LOGICAL*1 PT1, PT2, PT3
      INTEGER IACDFT(4) /2*11, 2*5/, IACMAX(4) /2*13, 2*8/, IACMIN(4) /2*4, 2
c      DATA IXTRAP, ITMAX/2*10/
      DATA IXTRAP, ITMAX/2*50/
      DIMENSION PP(999), CSEP(999), RR(999), IAC(4), AC(4)
C THESE DIMENSIONS ALLOW FOR UP TO 999 POINTS ON THE POTENTIAL CURVE
      EQUIVALENCE (AC(1), PAC), (AC(2), CAC), (AC(3), ACCIN), (AC(4), ACCOUT)
c common/pcs/p, c, fnew, pnew, cnew
character*3 ffile

```

```
c encode (3, '(i3)', ffile) 400+jvirt
c open (40, file=ffile, status='unknown')
c encode (3, '(i3)', ffile) 800+jvirt
c open (80, file=ffile, status='unknown')
if (ichoice.eq.1) read (iu80, *) zascr, zbscr, n, l, m !, pscr, cscr, sigma
c usful type *, 'read (80, *) zascr, zbscr, n, l, m, pscr, cscr, sigma'
c usful type *, zascr, zbscr, n, l, m, pscr, cscr, sigma
c type *, za, zb, n, l, m !, p, c, sigma
if (za.eq.0.d0) za=zascr
if (zb.eq.0.d0) zb=zbscr
  1      WRITE(IOUT, 600)
600    FORMAT(' 1-ELECTRON DIATOMIC MOLECULES, EXACT SOLUTIONS')
C   ENTER DATA
if (ichoice.ne.1) then
print*, 'read(iin, *) za, zb, n, l, m '
print*, 'read(iin, 5021) rstart, iac, ietot, irisk, iprint, ipunch, rj'
read(iin, *) za, zb, n, l, m
  read(iin, *) rstart, iac, ietot, irisk, iprint, ipunch, rj
!   read(iin, 5021) rstart, iac, ietot, irisk, iprint, ipunch, rj
      end if
  if(za*zb.le.0.d0) go to 15
!5021 format (f10.0, 8i2, f10.0)
c      READ(IIN, 501) ZA, ZB, N, L, M, IETOT, IRISK, IPRINT, IPUNCH, RJ
501    FORMAT(2F5.0, 7I2, F10.0)
C   ZA, ZB   NUCLEAR CHARGES           N, L, M   UA QUANTUM NUMBERS
C   IETOT=0 GIVES OUTPUT OF ELECTRONIC ENERGY W=E-ZA*ZB/R, WHERE R IS
C   THE INTERNUCLEAR DISTANCE.  A NONZERO IETOT GIVES OUTPUT OF THE
C   TOTAL ENERGY, E.
C   A NON-ZERO IRISK SLIGHTLY SPEEDS UP THE RUN BY
C   MODIFYING THE CONVERGENCE TEST AS DESCRIBED LATER.
C   IF IPUNCH IS NON-ZERO, THE EIGENPARAMETERS ARE PUNCHED ON CARDS.
```

```
C IPRINT CONTROLS PRINTED OUTPUT, AS FOLLOWS%
C =0  DEFAULT, MINIMUM PRINTED.  SHOULD BE SUFFICIENT IF NO TROUBLE
C =-1  VALUES OF P, C AND THEIR CHANGES PRINTED, TO FOLLOW ITERATIO
C      ALSO GIVES NUMBER OF CONVERGENTS REQUIRED FOR CTD. FRNS.
C =-2  ABOVE, PLUS VALUES OF F AND G AND THEIR DERIVATIVES, AND
C      VALUES OF THE CUTOFF PARAMETERS FOR F AND G.
C =-3  ABOVE, PLUS CONTINUED FRACTION APPROXIMATES SO THAT THE
C      CONVERGENCE OF THE CTD FRN CAN BE EXAMINED.  WARNING, THIS
C      CAN PRODUCE VOLUMINOUS OUTPUT.
C =+J  ESSENTIALLY SETS IPRINT=-2 BEGINNING WITH THE JTH R VALUE.
C      THUS, IF THE PROGRAM RAN OK FOR THE FIRST J-1 VALUES AND THE
C      HAD TROUBLE, ONE CAN BEGIN PRINTING OUT EXTRA DETAILS AT THE
C      DESIRED POINT.
C IF RJ IS NON-ZERO, THE VALUE OF RJUMP CALCULATED BELOW IS REPLACED
C BY RJ.  THIS OPTION IS USEFUL IN DEALING WITH QUASI-CROSSINGS.
c   IF(ZA.LE.0.D0) GO TO 15
      PT1=.FALSE.
      PT2=.FALSE.
      PT3=.FALSE.
      IF(IPRINT.LE.-1) PT1=.TRUE.
      IF(IPRINT.LE.-2) PT2=.TRUE.
      IF(IPRINT.LE.-3) PT3=.TRUE.
      IGO=1
      IF(ZA.EQ.ZB) IGO=2
      IF(M.LT.0) M=-M
      LSTART=M
      IF(L-M .NE. 2*((L-M)/2)) LSTART=M+1
INVERS=1
      IF(ZA.GE.ZB) GO TO 2
C SOME OF THE SUBROUTINES REQUIRE THAT ZA.GE.ZB AND THAT M.GE.0 .
INVERS=-1
```

```

      Z=ZA
      ZA=ZB
      ZB=Z
2      WRITE(IOUT,601) ZA,ZB,N,L,M,IETOT,IRISK,IPRINT,IPUNCH,RJ
601    FORMAT(' ZA,ZB,N,L,M,IETOT,IRISK,IPRINT,IPUNCH,RJ'
1 /1X,2F5.2,7I3,F10.4)
call corroedm(za,zb,n,l,m,ns,ks,icen)
c this finds the dissociation products of this oedm state.
sign=(-1)**(icen-1)
zz=za*(2-icen)+zb*(icen-1)
zp=za*(icen-1)+zb*(2-icen)
idelta=n-l-1-ks
c for a heteronuclear oedm, as soon as r.gt.rjump, we switch from th
c small r to the large r expansion for y(eta).
rjump=2*ns*ns*(1+2*dsqrt(dabs(zp/zz)))/dabs(zz)
c usful
write(iout,604) rjump
write(iout,*) 'icen =',icen
rab=rstart
rstart=0
if (rab.ge.rjump/2.) rstart=rjump+.1
c      READ(IIN,502) RSTART,IAC
502    FORMAT(F10.0,4I2)
      DO 21 I=1,4
          IF(IAC(I).LT.IACMIN(I) .OR. IAC(I).GT.IACMAX(I)) IAC(I)=IACDFT
21    AC(I)=1.D0/10.D0**IAC(I)
C PAC AND CAC CONTROL THE ACCURACY OF THE ITERATION FOR THE EIGEN-
C PARAMETERS P AND C, I.E., (P(I)-P(I-1))/(1+P(I)) .LE. PAC .
C ACCIN AND ACCOUT CONTROL THE TRUNCATION OF THE CONTINUED FRACTIONS
C FOR THE INNER AND OUTER EQUATIONS. FOR INSTANCE, WITH ACCIN=10**
C WE CAN EXPECT THE INNER EQUATION CONTINUED FRACTION TO HAVE APPROX

```



```
C MATELY K SIGNIFICANT DIGITS. SINCE THE NUMBER OF ACCURATE
C DIGITS IN P AND C IS ROUGHLY DOUBLED WITH EACH ITERATION, THERE IS
C NO SENSE IN CALCULATING THE FUNCTION VALUES MORE ACCURATELY THAN C
C EVER BE UTILIZED, I.E., FOR 10 DIGIT FINAL ACCURACY, A MAXIMUM OF
C ABOUT 5 DIGITS COULD BE GAINED IN THE 'BEST' ITERATION.
C A BLANK CARD ASSIGNS DEFAULT VALUES TO THESE PARAMETERS.
C CHOOSE A LOWER LIMIT FOR THE ACCURACY PARAMETERS AT LEAST A FEW
C DIGITS LESS THAN THE PRECISION OF YOUR COMPUTER. I HAVE USED
C A MAXIMUM ACCURACY OF 13 DIGITS, DEFAULT OF 11, FOR P AND C.
c WRITE(IOUT,602) RSTART,PAC,CAC,ACCIN,ACCOUT
602 FORMAT(' RSTART,PAC,CAC,ACCIN,ACCOUT ',1PD11.4,1P4D8.1)
Z=ZA+ZB
c NSAVE=NOPTS
NOPTS=1
RR(1)=RSTART
C ENTER VALUES OF R AT WHICH THE EIGENPARAMETERS WILL BE CALCULATED
c3 READ(IIN,503) NPT,RINCR
503 FORMAT(I2,F15.0)
c IF(NPT.EQ.0) GO TO 5
c WRITE(IOUT,608) NPT,RINCR
608 FORMAT(' NPT,RINCR ',I3,F12.8)
c ISTART=NOPTS+1
c NOPTS=NOPTS+NPT
c DO 4 I=ISTART,NOPTS
c4 RR(I)=RR(I-1)+RINCR
c GO TO 3
stepmin=0.5d0
frstincr=stepmin
if (rstart.eq.0) frstincr=0.95d0*n/z
c usful type '(a,2f10.3)', ' frstincr, 0.95*n/z',frstincr,0.95*n/z
nopts=2
```

```
if (rab.lt.rjump+.1) then
rdif=rab-rstart
if (abs(rdif).gt.frstincr) then
istart=3
nopts=5
rr(2)=rstart+dsign(frstincr,rdif)
rdif2=rab-rr(2)
mst=abs(rdif2)/stepmin
nop2=nopts-2
if (mst.lt.nop2) then
nopts=mst+3
rincr=dsign(stepmin,rdif)
else
rincr=(rdif2)/nop2
end if
do 4 i=istart,nopts-1
  4 rr(i)=rr(i-1)+rincr
end if
else
rstart=rab+.1
rr(1)=rstart
end if
rr(nopts)=rab
c usful write(iout,602) rstart,pac,cac,accin,accout
C IF THE INITIAL INCREMENT IS TOO LARGE, WE CANNOT GET RELIABLE
C GUESSES FOR P AND C AT RR(2), IF STARTING AT R=0.
5 IF(NOPTS.EQ.1) NOPTS=NSAVE
C IF NO POINTS HAVE BEEN ENTERED, USE THOSE FROM THE PREVIOUS RUN.
IF(RSTART.EQ.0.D0 .AND.RR(2)*Z/N .GT.1.D0) GO TO 14
IF(RSTART.NE.0.D0 .AND. RR(NOPTS).LT.0.01D0) GO TO 18
c I put the following calculation of rjump in the beginning, to defin
```

```

c      CALL CORR(ZA,ZB,N,L,M,NS,KS, ICEN)
cC    THIS FINDS THE DISSOCIATION PRODUCTS OF THIS OEDM STATE.
c      SIGN=(-1)**(ICEN-1)
c      ZZ=ZA*(2-ICEN)+ZB*(ICEN-1)
c      ZP=ZA*(ICEN-1)+ZB*(2-ICEN)
c      IDELTA=N-L-1-KS
cC    FOR A HETERONUCLEAR OEDM, AS SOON AS R.GT.RJUMP, WE SWITCH FROM T
cC    SMALL R TO THE LARGE R EXPANSION FOR Y(ETA).
c      RJUMP=2*NS*NS*(1+2*DSQRT(DABS(ZP/ZZ)))/DABS(ZZ)
cC    WE COULD ITERATE TO IMPROVE THIS ESTIMATE OF RJUMP, BUT THE ABOVE
cC    SUFFICIENTLY ACCURATE.
c up to here
      CALL WRLG1(ZZ,ZP,NS, IDELTA,M,1.D0,W,W0,W1,W2,W3,W4,W5,W6)
cC    THIS INITIALIZES THE CONSTANTS IN THE LARGE R ASYMPTOTIC EXP. OF W
cC    SET PARAMETER VALUES AT R(1)=RSTART
c      WRITE(IOUT,604) RJUMP
604   FORMAT(' THE 1/R ASYMPTOTIC EXPANSION OF W IS VALID FOR R.GT.RJ
      X=',F6.2)
c usful      WRITE(IOUT,605) W0,W1,W2,W3,W4,W5,W6
605   FORMAT(' COEF OF THE 1/R EXPANSION OF W'/1P4D18.10/1P3D18.10)
      IF(RSTART.NE.0.D0 .AND. RSTART.LT.RJUMP) GO TO 19
      IF(RJ.EQ.0.D0) GO TO 39
      RJUMP=RJ
      WRITE(IOUT,624) RJ
624   FORMAT(' SWITCH TO LARGE R Y(ETA) EXPANSION FOR R.GE.',F10.4)
39     IF(RSTART.NE.0.D0) GO TO 40
      PP(1)=0
      CSEP(1)=L*(L+1)
      ISTART=2
      GO TO 41
40     ISTART=1

```

```

        CALL WRLG2 (RSTART, W, W0, W1, W2, W3, W4, W5, W6)
        PP (1) =RSTART*DSQRT (-W/2)
        CSEP (1) =CRLG1 (N, L, M, PP (1), RSTART*Z)
C
C BEGIN THE DO-LOOP OVER THE R VALUES
C
41      DO 10 I=ISTART, NOPTS
        R=RR (I)
        IF (IGO.EQ.3) IGO=1
        IF (IGO.EQ.1 .AND. R.GE.RJUMP) IGO=3
C IGO=1, 2, OR 3 RESPECTIVELY FOR THE HETERONUCLEAR SMALL R, HOMONUCLEAR
C OR HETERONUCLEAR LARGE R EXPANSIONS FOR Y (ETA) .
        IF (I.EQ.IPRINT) PT2=.TRUE.
        IF (PT2) PT1=.TRUE.
        RZ=R*Z
        RZDIF=R* (ZA-ZB) *SIGN
C SIGN=+1 FOR A CENTRE A SA STATE, -1 FOR A CENTRE B SA STATE.
C THE SIGN OF RZDIF DETERMINES WHETHER WE HAVE THE CENTRE A OR B
C LARGE R EXPANSION FOR Y (ETA) GIVING THE CONTINUED FRACTION.
        IF (I.GT.2) GO TO 6
        IF (I.EQ.1) GO TO 7
C ESTIMATES FOR P AND C AT R (2)
        IF (RSTART.NE.0.D0) GO TO 51
        PP (2) =0.5D0*R*Z/N
C MIXED MODE ARITHMETIC (INTEGER AND REAL) AS IN THE ABOVE STATEMENT
C O.K. WITH THE FORTRAN COMPILER ON MY COMPUTER, AND I HAVE USED
C IT EXTENSIVELY IN THIS MAIN PROGRAM AND THE SUBROUTINES.
        ZK=RZDIF*0.5D0/PP (2)
        IF (L.EQ.0) DC=2* (1-ZK*ZK) /3*PP (2) **2
        IF (L.NE.0) DC=2* ( ( (L+1) **2-M*M) * ( (L+1) **2-ZK*ZK) / ( (L+1) * (L+L+3)
X          - (L*L-M*M) * (L*L-ZK*ZK) / (L* (L+L-1) ) ) / (L+L+1) *PP (2) **2

```

```

      CSEP (2) =CSEP (1) +DC
      GO TO 7
51    CALL WRLG2 (RR (2) , W, W0, W1, W2, W3, W4, W5, W6)
      PP (2) =R*DSQRT (-W/2)
      CSEP (2) =CRLG2 (PP (2) , RR (2) *Z)
      GO TO 7
C    EXTRAPOLATE FOR P AND C AT CURRENT R
6      NPT=MIN0 (I, IXTRAP)
C    USE DATA FROM THE IXTRAP(=10) PREVIOUS POINTS, IF AVAILABLE.
      J=I+1-NPT
      CALL EXTRAP (RR (J) , PP (J) , NPT)
      CALL EXTRAP (RR (J) , CSEP (J) , NPT)
7      P=PP (I) * (1.000123456789)
      C=CSEP (I)
C    THESE ARE THE STARTING GUESSES FOR P AND C, OBTAINED FROM
C    SUBROUTINE EXTRAP.  IN ORDER TO ENSURE CONVERGENCE TO THE DESIRED
C    ROOT, THESE INITIAL GUESSES MUST BE CLOSER TO THIS EXACT ROOT THAN
C    TO ANY OTHER EXACT ROOT.  IF W(R) IS CHANGING LINEARLY WITH R, THE
C    EXTRAPOLATION WILL BE VERY ACCURATE.  HOWEVER, IF W(R) IS CURVING,
C    AS NEAR A MINIMUM OR AN AVOIDED CROSSING (QUASI-CROSSING), THE
C    INCREMENTS IN R MUST BE SUFFICIENTLY SMALL IN ORDER FOR THE
C    EXTRAPOLATION TO FOLLOW THIS CHANGE.  IF THE INCREMENTS ARE TOO
C    LARGE, THE INITIAL GUESSES CAN BE VERY POOR, AND THE ITERATION
C    CAN CONVERGE TO A ROOT CORRESPONDING TO A DIFFERENT STATE.
      CALL CHAIN (FCHAIN, GCHAIN, IGO, RZ, RZDIF, N, L, M, KS, P, C)
C    DETERMINE THE CHAINING VALUES FOR THE CONTINUED FRACTIONS
      IF (PT1) WRITE (IOUT, 620) R, FCHAIN, GCHAIN, P, C
620  FORMAT ('0R, FCHAIN, GCHAIN, INITIAL P, C=', F10.5, 2I3, 1P2D20.12)
      KOUNT=0
C    INITIALIZE THE ITERATION COUNT, BEGIN ITERATING FOR P AND C.
      FCUT=0

```

```

GCUT=0
c usful type *, ' R=',R
8 CALL CTDFRN(IGO,P,C,RZ,RZDIF,n,l,M,FCHAIN,LSTART,ACCIN,FCUT,F,
X DFDC,DFDP,IOUT,PT3,NCVIN,icen,INVERS,za,zb)
C
C EVALUATE THE TWO CONTINUED FRACTIONS
CALL CTDFRN(4,P,C,RZ,RZDIF,n,l,M,GCHAIN,LSTART,ACCOU,GCUT,G,
X DGDC,DGDP,IOUT,PT3,NCVOUT,icen,INVERS,za,zb)
C IGO=4 GIVES THE CONTINUED FRACTION FOR THE OUTER EQUATION. THUS F
C AND ITS DERIVATIVES CORRESPOND TO THE INNER EQN., G TO THE OUTER.
C THE FUNCTIONS F AND G ARE BOTH 0 WHEN P AND C ARE ROOTS. DFDC AND
C DFDP ARE THE FIRST PARTIAL DERIVATIVES OF F WITH RESPECT TO C AND
DENOM=DFDC*DGDP-DFDP*DGDC
DELTAP=(F*DGDC-G*DFDC)/DENOM
DELTAC=(G*DFDP-F*DGDP)/DENOM
C
C USE THE 2-DIMENSIONAL NEWTON-RAPHSON ITERATION FUNCTION TO GET
C IMPROVED ESTIMATES FOR P AND C.
P=P+DELTAP
C=C+DELTAC
IF(PT1) WRITE(IOUT,621) P,C,DELTAP,DELTAC,NCVIN,NCVOUT
621 FORMAT(' P,C,DELTAP,DELTAC=',1P2D20.12,1P2D16.8/' CTD. FRN. C
XERGENTS INNER=',0PI3,' OUTER=',I3)
IF(PT2) WRITE(IOUT,622) F,DFDC,DFDP,G,DGDC,DGDP
622 FORMAT(' F,DFDC,DFDP,G,DGDC,DGDP=',1P6D17.8)
C TEST FOR THE CONVERGENCE OF THE VALUES OF P AND C.
IF(DABS(DELTAP)/(1+P).LT.PAC .AND. DABS(DELTAC)/(1+DABS(C)).LT.
X) GO TO 9
C INSTEAD OF TESTING THE MAGNITUDE OF DELTAP/P, WE TEST DELTAP/(1+P)
C WHICH AVOIDS PROBLEMS WHEN P IS SMALL.
KOUNT=KOUNT+1

```

```

        IF(KOUNT.GT.ITMAX) GO TO 16
C THE ITERATION WILL EASILY CONVERGE IN 3-4 CYCLES FOR ISOLATED ROOT
        FCUT=0.5D0*DABS(DENOM)*DMAX1(PAC*(1+P)/DABS(DGDC),CAC*(1+DABS(C
X /DABS(DGDP))
        GCUT=0.5D0*DABS(DENOM)*DMAX1(PAC*(1+P)/DABS(DFDC),CAC*(1+DABS(C
X /DABS(DFDP))
C SET THE VALUES OF THE CUTOFF PARAMETERS. THE DERIVATIVE VALUES
C CHANGE ONLY SLIGHTLY AS THE ITERATION PROGRESSES, WHEREAS F AND G
C GO TOWARDS ZERO. WE THUS MAY BE ABLE TO SAY THAT IF THE VALUE
C OF F IS LESS THAN SOME CUTOFF IN MAGNITUDE, IT WILL NOT CHANGE THE
C SUBSEQUENT ITERATED VALUES OF P AND C TO WITHIN THE ACCURACIES
C SPECIFIED BY PAC AND CAC. WE PASS THIS INFORMATION ON TO THE
C CONTINUED FRACTION SUBROUTINE, WHICH ADJUSTS THE TRUNCATION
C CONTROL SO THAT WE DO NOT WASTE EFFORT FINDING, SAY, THAT
C F=1.2345D-11, WHEN F=1.2D-11 IS SUFFICIENT. THE MAXIMUM
C ACCURACY OF THE CONTINUED FRACTION TRUNCATION TEST IS SPECIFIED BY
C THE VALUES OF ACCIN AND ACCOUT. FCUT AND GCUT DECREASE THESE
C ACCURACIES, WHEN APPROPRIATE.
        IF(PT2) WRITE(IOUT,623) FCUT,GCUT
623  FORMAT(' FCUT,GCUT=',1P2D14.5)
        IF(IRISK.EQ.0) GO TO 8
C IF IRISK.NE.0, WE TAKE A 'RISK' WITH THE CONVERGENCE TEST, AND
C POSSIBLY SAVE ONE ITERATION CYCLE. IF THE LATEST VALUES OF
C F AND G ARE SMALL ENOUGH, WE GAMBLE THAT THE NEXT ITERATION
C CYCLE WILL ONLY PROVE CONVERGENCE AND NOT MAKE A MEANINGFUL
C CHANGE IN THE VALUES OF P AND C, AND THUS GO TO THE NEXT R.
        IF(DABS(F)*ACCIN/FCUT.LT.1.D0 .AND. DABS(G)*100*ACCOUT/GCUT.LT.
X 1.D0) GO TO 9
C ACCIN IS A RELIABLE INDICATOR OF THE NUMBER OF SIGNIFICANT DIGITS
C IN THE VALUE OF F, BUT THE VALUE OF G MAY HAVE SEVERAL DIGITS LESS
C ACCURACY THAN GIVEN BY ACCOUT BECAUSE OF THE SLOW CONVERGENCE OF T

```

```
C   TAIL OF ITS CONTINUED FRACTION, SO WE HAVE THE FUDGE FACTOR OF 100
      GO TO 8
9     CSEP(I)=C
10    PP(I)=P
C
C   THE DO-LOOP OVER R VALUES HAS BEEN COMPLETED
C
C   STORE ENERGIES IN THE VECTOR PP
101   IF(RSTART.NE.0.D0) GO TO 11
      PP(1)=-0.5D0*(Z/N)**2
11    DO 12 I=ISTART,NOPTS
12    PP(I)=-2*(PP(I)/RR(I))**2
C   BEGIN PRINTING OUT RESULTS
      WRITE(IOUT,603) ZA,ZB,N,L,M
603   FORMAT('0ZA=',F5.2,' ZB=',F5.2,'      UA QUANTUM NUMBERS  N=',I2,
X  L=',I2,' M=',I2)
      IF(IGO.NE.2) WRITE(IOUT,609) NS,KS,CENTER(ICEN)
609   FORMAT(' SA QUANTUM NUMBERS NS=',I2,' KS=',I2,' FOR CENTRE',A)
      IF(IGO.EQ.2) WRITE(IOUT,610) NS,KS
610   FORMAT(' SA QUANTUM NUMBERS  NS=',I2,' KS=',I2/)
      JJ=1
      IF(IETOT.EQ.0) GO TO 13
C   IF IETOT IS NON-ZERO, PRINT OUT THE TOTAL INSTEAD OF ELECTRONIC EN
      JJ=2
      ZAB=ZA*ZB
      DO 17 I=ISTART,NOPTS
17    PP(I)=PP(I)+ZAB/RR(I)
      IF(ISTART.EQ.2) PP(1)=1.D30
13    WRITE(IOUT,612) EOUT(JJ)
612   FORMAT('0',5X,'R',14X,A2,18X,'C'/)
      WRITE(IOUT,606) (RR(I),PP(I),CSEP(I),I=1,NOPTS)
```



```
606  FORMAT(1H ,0PF9.4,2X,1P2D19.9)
      IF(IPUNCH.EQ.0) GO TO 1 5 !5 added by me
      WRITE(IPCH,701) ZA,ZB,N,L,M,NS,KS,ICEN,IETOT,NOPTS
701  FORMAT(2F5.0,7I2,I3)
      WRITE(IPCH,702) (RR(I),PP(I),CSEP(I),I=1,NOPTS)
702  FORMAT(2(0PF8.4,1P2D16.9))
C   IF ONE WANTS TO PRINT OUT OR PUNCH THE RESULTS WITH MORE THAN 10
C   DIGITS, FORMATS 606 AND 702 HAVE TO BE CHANGED.
      GO TO 1 5 !5 added by me
14   WRITE(IOUT,607)
607  FORMAT(' RR(2)*Z/N .GT. 1, SKIP TO THE NEXT RUN')
      GO TO 1 5 !5 added by me
16   WRITE(IOUT,611) I,R
611  FORMAT('0*** MAX. NO. OF ITERATIONS EXCEEDED AT R(',I3,') =',F10.6,
      X)
C   PRINT OUT ANY RESULTS WHICH WERE OBTAINED.
      NOPTS=I-1
      GO TO 101
18   WRITE(IOUT,613) RR(NOPTS)
613  FORMAT(' R=',F10.6,' NEGATIVE OR TOO CLOSE TO ZERO, SKIP RUN')
      GO TO 1 5 !5 added by me
19   WRITE(IOUT,614) RSTART,RJUMP
614  FORMAT(' RSTART=',F10.6,' .LT. RJUMP=',F6.2,' SKIP RUN')
      GO TO 1 5 !5 added by me
c15  STOP
15   continue
rstart=rr(nopts)
if (invers.eq.-1) then
zt=za
za=zb
zb=zt
```

```
end if
close (iu40)
close (iu80)
return
      END

C
      SUBROUTINE CORROedm (ZA, ZB, N, L, M, NS, K, ICEN)
      REAL*8 ZA, ZB
C   CORRELATES UA AND SA STATES USING ANALYTICAL FORMULAE OF J. POWER,
C   1973 PHIL.TRANS.ROY.SOC.LOND. (TO BE PUBLISHED).
C   ASSUMES THAT ZA.GE.ZB
      IF (ZA.NE.ZB) GO TO 1
      ICEN=0
C   ZA=ZB, HOMONUCLEAR OEDM
      ICEN=0
C   THE CHOICE OF CENTER IS IRRELEVANT
      K=(L-M)/2
      NS=N-L+M+K
      RETURN
C   HETERONUCLEAR OEDM
1     NU= ((1+N)*ZB-ZA*(M+N-1-L))/(ZA+ZB)
      IF (NU.LT.0) NU=0
      NUTEST=(N*ZB-ZA*(M+N-1-L))/(ZA+ZB)
      IF (NU.EQ.0 .OR. NUTEST.EQ.NU) GO TO 2
C   DISSOCIATES WITH THE ELECTRON LOCALIZED ON CENTRE B
      ICEN=2
      K=NU-1
      NS=N-L+M+K
      RETURN
C   DISSOCIATES WITH THE ELECTRON LOCALIZED ON CENTRE A
2     ICEN=1
```

```

K=L-M-NU
NS=N-NU
RETURN
END

SUBROUTINE CTDFRN(IGO,P,C,RZ,RZDIF,n,l,M,LCHAIN,LSTART,ACCY,
X CUTOFF,F,DFDC,DFDP,IOUT,PT3,ICLK,icen,INVERS,za,zb)
C EVALUATES THE CONTINUED FRACTIONS AND THEIR DERIVATIVES FOR THE IN
C AND OUTER EQUATIONS. IGO=1,2,OR 3 CHOOSES BETWEEN RESPECTIVELY T
C HETERONUCLEAR SMALL R, HOMONUCLEAR, OR HETERONUCLEAR LARGE R
C EXPANSIONS FOR Y(ETA). IGO=4 GIVES THE CTND FRACTION FOR X(XI).
IMPLICIT REAL*8(A-H,O-Z)
LOGICAL*1 TEST,PT3
common/crjumpoedm/rjump
common/ciu/iu40,iu80
DATA SCALE/1.D-37/
TEST=.FALSE.
ICLK=0
c usful type 601, igo,p,c,lchain
IF(PT3) WRITE(IOUT,601) IGO,P,C,LCHAIN
601 FORMAT(' BEGIN CTD FRN, IGO=',I2,' P,C=',1P2D20.12,' CHAIN AT'
XPI3)
ACY=ACCY
MM=M*M
P2=2*P*P
JINCR=1
C INITIALIZE VARIABLES FOR THE NUMERATORS. A CONTAINS THE CURRENT
C NUMERATOR, AO THE PREVIOUS ONE, DADC THE PARTIAL DERIVATIVE OF A
C WITH RESPECT TO C, ETC. IN THE FORWARD METHOD, WE ONLY NEED THE
C NUMERATORS.
AO=1

```

```
      DAODC=0
      DAODP=0
      DADC=1
      GO TO (1,2,3,4), IGO
1     J=M
      RZDIF2=RZDIF*RZDIF
      A=C-MM-M
      DADP=0
      ASSIGN 6 TO JGO
      GO TO 40
2     J=LSTART
      P4=P2*P2/4
      JJ=J*J
      ASAVE=(MM+JJ+J-1.D0)/(4*(JJ+J)-3)
      A=C-JJ-J-P2*ASAVE
      DADP=-4*P*ASAVE
      JINCR=2
C    AN INCREMENT OF 2 IS NEEDED FOR THE HOMONUCLEAR Y(ETA) FUNCTION.
      ASSIGN 7 TO JGO
      GO TO 40
3     J=0
      ZK=0.5D0*RZDIF/P
      S1=C-MM-M-2*P*(M+1-ZK)
      S2=-4*P-M-M-1
      DS1=-2*(M+1)
      A=S1
      DADP=DS1
      ASSIGN 8 TO JGO
      GO TO 40
4     J=0
      SIGMA=0.5D0*RZ/P-M-1
```

```

S1=C-RZ-(M+1)*(M+SIGMA-P-P)
S2=4*P-2*SIGMA
DSIGMA=-0.5D0*RZ/P**2
DS1=(M+1)*(2-DSIGMA)
DS2=4-2*DSIGMA
A=S1
DADP=DS1
ASSIGN 9 TO JGO
GO TO 40
5   J=J+JINCR
    ICHK=ICHK+1
C   ICHK GUARDS AGAINST AN INFINITE LOOP IF THE CONTINUED FRACTION IS
C   UNEXPECTEDLY FAILING TO CONVERGE.
    IF(ICHK.EQ.1000) STOP 1000
C   THE FOLLOWING SECTION EVALUATES THE ELEMENTS OF THE CONTINUED FRACTION
C   AT THE CURRENT VALUE OF J, FOR BOTH THE FORWARD AND BACKWARD METHODS.
    JJ=J*J
    GO TO JGO, (6, 7, 8, 9)
6   ASAVE=(JJ-MM)/(4*JJ-1.D0)
    AJ=(2*P2*JJ-RZDIF2)*ASAVE
    BJ=C-JJ-J
    DAJ=8*P*JJ*ASAVE
    DBJ=0.D0
if (j.eq.m+1) then
iu=iu40
go=1.d0
jm1=j-1
gamajm1=(jm1+1+m)*(2*p*(jm1+1)-rzdif)/(2*jm1+3.)
betajm1=c-jm1*(jm1+1)
gj=-betajm1/gamajm1
rewind iu

```

```

write (iu, '(2D25.15,i5,D25.15,3i5,D25.15,i5)')
1 p,c,m,rzdif,icen,igo,lstart,RZ,INVERS
write (iu, '(2D25.15)') go,gj
end if
alfaj=- (j-m) * (2*p*j+rzdif) / (2*j-1.)
betaj=c-j*(j+1)
gama j=(j+1+m) * (2*p*(j+1)-rzdif) / (2*j+3.)
gn=- (alfaj*go+betaj*gj) / gama j
      GO TO 10
7     AJ=-P4*(JJ-J-J+1-MM) * (JJ-MM) / ((4*JJ-8*J+3) * (4*JJ-1))
      ASAVE=(MM+JJ+J-1.D0) / (4*(JJ+J)-3)
      BJ=C-JJ-J-P2*ASAVE
      DAJ=4*AJ/P
      DBJ=-4*P*ASAVE
if (j.eq.lstart+2) then
iu=iu40
go=1.d0
jm1=j-2
gama jm1=p*p*(jm1+2+m) * (jm1+1+m) / (2*jm1+3.) / (2*jm1+5.)
betajm1=(jm1+1-m) * (jm1+1+m) / (2*jm1+1.) / (2*jm1+3.) +
1 (jm1*jm1-m*m) / (4*jm1*jm1-1.)
betajm1=c-jm1*(jm1+1)+p*p*(betajm1-1)
gj=-betajm1/gama jm1
rewind iu
write (iu, '(2D25.15,i5,D25.15,3i5,D25.15,i5)')
1 p,c,m,rzdif,icen,igo,lstart,RZ,INVERS
write (iu, '(2D25.15)') go,gj
end if
alfaj=p*p*(j-1-m) * (j-m) / (2*j-3.) / (2*j-1.)
betaj=(j+1-m) * (j+1+m) / (2*j+1.) / (2*j+3.) + (j*j-m*m) / (4*j*j-1.)
betaj=c-j*(j+1)+p*p*(betaj-1)

```

```

gama j=p*p*(j+2+m)*(j+1+m)/(2*j+3.)/(2*j+5.)
gn=- (alfaj*go+betaj*gj)/gama j
      GO TO 10
8      ASAVE=-4*J*(J+M)
      DAJ=ASAVE*(J+M)
      AJ=P*ASAVE*(J+M-ZK)
      BJ=-JJ+J*S2+S1
      DBJ=DS1-4*J
if (j.eq.1) then
iu=iu40
go=1.d0
gj=p+m/2.-((rzdif)+c)/(m+1.)/2.
rewind iu
write (iu,'(2D25.15,i5,D25.15,3i5,D25.15,i5)')
1 p,c,m,rzdif,icen,igo,lstart,RZ,INVERS
write (iu,'(2D25.15)') go,gj
end if
alfaj=2*p*(j+m)-(rzdif)
betaj=(rzdif)+c-(2*j+m+1)*(m+2*p)-j*(j+1)
gama j=(j+1)*(j+1+m)*2
gn=- (alfaj*go+betaj*gj)/gama j
      GO TO 10
9      ASAVE=-J*(J+M)
      AJ=ASAVE*(J-1-SIGMA)*(J-1-SIGMA-M)
      BJ=S1+J*S2+2*JJ
      DAJ=ASAVE*(M+2*(1-J+SIGMA))*DSIGMA
      DBJ=DS1+J*DS2
if (j.eq.1) then
iu=iu80
go=1.d0
gj=s1/(m+1)

```

```
rewind iu
  write (iu, '(2D25.15,3i5,3D25.15)') za, zb, n, l, m, p, c, sigma
write (iu, '(3D25.15)') rjump, go, gj
end if
alfaj=aj/asave
betaj=-bj
gama j=(j+1)*(j+1+m)
gn=-(alfaj*go+betaj*gj)/gama j
10  CONTINUE
c write (iu, '(i4,1p4E20.9)') j, alfaj, betaj, gama j, gn
write (iu, '(i5,D25.15)') j+1, gn
go=gj
gj=gn
      IF(TEST) GO TO 12
C  WHEN TEST IS TRUE, DO THE BACKWARD EVALUATION
C  FORWARD EVALUATION
      ASAVE=DADC
      DADC=A+BJ*DADC+AJ*DAODC
      DAODC=ASAVE
      ASAVE=DADP
      DADP=BJ*DADP+AJ*DAODP+DBJ*A+DAJ*AO
      DAODP=ASAVE
      ASAVE=A
      A=BJ*A+AJ*AO
      AO=ASAVE
40  IF(J.LT.LCHAIN) GO TO 5
      TEST=.TRUE.
C  WE HAVE FINISHED THE FORWARD EVALUATION, NOW INITIALIZE VARIABLES
C  THE BACKWARD EVALUATION.
      A=A/AO
      IF(PT3) WRITE(IOUT,602) A
```



```

602  FORMAT(' VALUE FROM FORWARD CALC.=' ,1PD20.12,/, ' J, A(J)/B(J) '
      DADP=(DADP-A*DAODP)/AO
      DADC=(DADC-A*DAODC)/AO
      AO=1
      DAODC=0
      DAODP=0
      BO=0
      DBODC=0
      DBODP=0
      B=1
      DBDC=0
      DBDP=0
      RATOLD=A
      DIFOLD=1
      GO TO 5
C   BACKWARD EVALUATION
12   AC=DADC
      DADC=A+BJ*DADC+AJ*DAODC
      BC=DBDC
      DBDC=B+BJ*DBDC+AJ*DBODC
      AP=DADP
      DADP=BJ*DADP+AJ*DAODP+DBJ*A+DAJ*AO
      BP=DBDP
      DBDP=BJ*DBDP+AJ*DBODP+DBJ*B+DAJ*BO
      ASAVE=A
      A=BJ*A+AJ*AO
      BSAVE=B
      B=BJ*B+AJ*BO
      RATNEW=A/B
      IF (PT3) WRITE (IOUT, 603) J, RATNEW
603  FORMAT(1H , I3, 1PD20.12)

```

```
DENOM=DABS (RATNEW) +SCALE
C  ADDING SCALE (1.D-37) AVOIDS POSSIBLE DIVISION BY ZERO
    DIFNEW=DABS (RATNEW-RATOLD) /DENOM
    IF (DIFNEW.GT.0.1D0) GO TO 13
C  WHEN THE ABOVE TEST IS FALSE, THE CONTINUED FRACTION IS BEGINNING
C  CONVERGE, AND RATNEW HAS NEARLY 2 SIGNIFICANT DIGITS.  THUS, MAKE
C  THE MORE STRINGENT CONVERGENCE TEST, COMPARING THE VALUES OF THREE
C  SUCCESSIVE APPROXIMATES.
    IF (DIFNEW+DIFOLD .LE.ACY) GO TO 15
    ACP=CUTOFF/DENOM
    IF (ACP.GT.1.D-2) ACP=1.D-2
    IF (ACY.LT.ACP) ACY=ACP
C  WE HAVE USED THE VALUE OF CUTOFF AND THE CURRENT ESTIMATE OF F TO
C  MAKE A POSSIBLE ADJUSTMENT IN THE CONVERGENCE CONTROL, SO THAT WE
C  NOT CALCULATE F TO MORE SIGNIFICANT DIGITS THAN CAN BE USED.
13  DIFOLD=DIFNEW
    RATVO=RATOLD
    RATOLD=RATNEW
    IF (DABS (B) .GT.1.D30) GO TO 14
    AO=ASAVE
    BO=BSAVE
    DAODC=AC
    DAODP=AP
    DBODC=BC
    DBODP=BP
    GO TO 5
C  B IS GETTING LARGE, RESCALE ALL TERMS TO AVOID EXPONENT OVERFLOWS
14  A=A*SCALE
    B=B*SCALE
    AO=ASAVE*SCALE
    BO=BSAVE*SCALE
```

```
DADC=DADC*SCALE
DBDC=DBDC*SCALE
DAODC=AC*SCALE
DBODC=BC*SCALE
DADP=DADP*SCALE
DBDP=DBDP*SCALE
DAODP=AP*SCALE
DBODP=BP*SCALE
GO TO 5
C THE APPROXIMATES OF THE INFINITE CONTINUED FRACTION HAVE CONVERGED
C TO SUFFICIENT ACCURACY
15 F=RATNEW
    IF (DABS (F-RATOLD) .Le .1.D-12*DABS (F) ) GO TO 16
C USE THE AITKEN DEL-SQUARED EXTRAPOLATION ON THE LAST THREE
C APPROXIMATES, WHICH MAY ADD AN ADDITIONAL SIGNIFICANT DIGIT TO
C THE ESTIMATE FOR F.
    F=RATVO- (RATOLD-RATVO) **2/ (F-RATOLD-RATOLD+RATVO)
C ONE COULD OF COURSE SAVE, SAY, THE LAST TEN APPROXIMATES AND USE
C A BETTER EXTRAPOLATION, SUCH AS SUBROUTINE 'DTEAS' FROM THE IBM SS
C LIBRARY. I HAVE NOT JUDGED THIS TO BE WORTHWHILE IN MY WORK, BUT
C IF ONE HAS A COMPUTER SUCH AS CDC MACHINES WITH MUCH HIGHER POSSIBLE
C ACCURACY, AND WANTS TO GET P AND C ACCURATE TO 20 DIGITS, THE SLOW
C ASYMPTOTIC CONVERGENCE OF THE OUTER EQUATION'S CONTINUED FRACTION
C WOULD BECOME RATE DETERMINING, AND SUCH AN EXTRAPOLATION MIGHT
C PROVE VALUABLE. IN THIS CASE ONE COULD ATTEMPT TO ACCELERATE
C CONVERGENCE BY EXTRAPOLATING THE APPROXIMATES AND TESTING THE
C CONVERGENCE OF THE EXTRAPOLATES, NOT THE APPROXIMATES.
16 DFDC= (DADC-F*DBDC) /B
    DFDP= (DADP-F*DBDP) /B
    RETURN
END
```

```
      SUBROUTINE EXTRAP (X,F,N)
C   GIVEN VALUES OF X FOR I=1,2,...,N AND F FOR I=1,2,...,N-1,
C   EXTRAPOLATE TO GET AN ESTIMATE OF F FOR I=N.
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(1),F(1),DELTA(8)
      NM2=N-2
C   FORM THE DIVIDED DIFFERENCES
      DO 1 I=1,NM2
1     DELTA(I)=(F(I+1)-F(I))/(X(I+1)-X(I))
C   THE FUNCTIONS WE ARE EXTRAPOLATING ARE APPROXIMATELY LINEAR IN X
C   SO THE DIVIDED DIFFERENCES WILL BE NEARLY CONSTANT. WE FIRST
C   EXTRAPOLATE TO GET AN ESTIMATE OF THE DIVIDED DIFFERENCE
C   CORRESPONDING TO I=N.
      J=NM2
2     J=J-1
      IF(J.EQ.0) GO TO 4
      DO 3 I=1,J
3     DELTA(I)=DELTA(I+1)-DELTA(I)
      GO TO 2
4     SUM=DELTA(NM2)
      J=NM2
5     J=J-1
      IF(J.EQ.0) GO TO 6
C   EACH SUCCESSIVE TERM SHOULD BE AN ORDER OF MAGNITUDE SMALLER THAN
C   THE PREVIOUS TERM. IF NOT, TRUNCATE.
      IF(DABS(DELTA(J)).GT.0.2D0*DABS(DELTA(J+1))) GO TO 6
      SUM=SUM+DELTA(J)
      GO TO 5
C   SUM IS THE ESTIMATE FOR DELTA(N-1)
6     F(N)=F(N-1)+(X(N)-X(N-1))*SUM
      RETURN
```

```

END
FUNCTION CRLG1 (N, L, M, P, RZ)
C ASYMPTOTIC LARGE R EXPANSION OF C IN POWERS OF 1/P
IMPLICIT REAL*8 (A-H, O-Z)
NU=2* (N-L-1) +M+1
MM=M*M
NN=NU*NU
C1=-0.5D0* (NN+1-MM)
C2=NU* (NN+1-MM)
C3=MM-1-3*NN
C4=6*NN*MM- (1-MM) **2-5*NN*NN-10*NN
C5=NU* (20* (NN+1) -12*MM)
C6=8* (MM-1) -24*NN
C7=NU* (NN* (33*NN+114-46*MM) +37+MM* (13*MM-50) )
C8=NN* (138*MM-342-165*NN) -37+MM* (50-13*MM)
C9=NU* (284*NN+292-156*MM)
C10=NN* (MM* (230-39*MM+100*NN) -239-NN* (340+63*NN) ) -14+MM* (30-MM*
X (18-2*MM) )
C11=NU* (NN* (1360+378*NN-400*MM) +478+MM* (78*MM-460) )
C12=NN* (630*MM-1810-845*NN) -209+MM* (250-41*MM)
C13=NU* (860*NN+900-460*MM)
C14=NU* (NN* (5221+MM* (465*MM-939*NN-3750) +NN* (4139+527*NN) ) +1009
X -MM* (1591-MM* (635-53*MM) ) )
C15=NN* (MM* (11250-1395*MM+4695*NN) -15663-NN* (20695+NN*3689) ) -10
X +MM* (1591-MM* (635-53*MM) )
C16=NU* (14072+NN* (37640-9520*MM+10128*NN) -MM* (11640-1440*MM) )
C17=NN* (9780*MM-30140-13750*NN) -3630+MM* (4140-510*MM)
C18=NU* (9520*NN+10080-5040*MM)
ENTRY CRLG2 (P, RZ)
S=RZ/ (2*P)
CRLG1=2*P* (S-NU) +NU*S+C1+ (C2+S* (C3+2*NU*S) ) / (8*P) + (C4+S* (C5+S*

```

```

X 8*NU*S)) / (64*P*P) + (C7+S* (C8+S* (C9+S* (8*C6+40*NU*S))) / (512*P*
X + (C10+S* (C11+S* (C12+S* (C13+S* (16*C6+56*NU*S)))) / (1024*P**4) +
X +S* (C15+S* (C16+S* (C17+S* (C18+S* (128*C6+336*NU*S)))))) / (8192*P*
RETURN
END
SUBROUTINE WRLG1 (Z, ZP, N, D, M, R, W, W0, W1, W2, W3, W4, W5, W6)
C ASYMPTOTIC LARGE R EXPANSION OF W IN POWERS OF 1/R
IMPLICIT REAL*8 (A-H, O-Z)
INTEGER D, DD
RATIO=ZP/Z
NN=N*N
DD=D*D
MM=M*M
W0=-0.5D0*Z*Z/NN
W1=-ZP
W2=1.5D0*N*D*RATIO
W3=-NN*(6.*DD+1.-NN)*RATIO/(2.*Z)
W4=N*NN*D*(109.*DD-39.*NN-9.*MM+59.)*RATIO/16.+NN*NN*
1 (3.*DD+9.*MM-17.*NN
X -19.)*RATIO**2/16.
W5=NN*NN*(594.*NN*DD-1065.*DD*DD-1230.*DD+234.*MM*DD
1 -9.*MM*MM-33.*NN*NN
X +18.*MM*(1.+NN)-105+138.*NN)*RATIO/64.
1 +N*NN*NN*D*(111.*NN-21.*DD-63.*MM
X +189.)*RATIO**2/16.
W6=N*NN*NN*D*(2727.*DD*DD-2076.*NN*DD+5544.*DD
1 -1056.*DD*MM+93.*MM*MM
X +273.*NN*NN-78.*NN*MM-450.*MM+1533.-1470.*NN)*RATIO/64.
1 +NN*NN*NN*(207.*
X DD*DD-1044.*NN*DD-2436.*DD+576.*MM*DD+42.*NN
1 -371.+162.*MM-42.*NN*MM

```

X +89.*NN*NN-15.*MM*MM)*RATIO**2/32.

1 +NN*NN*NN*N*D*(69.*NN-3.*DD+117.

X +33.*MM)*RATIO**3/32.

W4=W4/Z**2

W5=W5/Z**3

W6=W6/Z**4

ENTRY WRLG2(R,W,W0,W1,W2,W3,W4,W5,W6)

T=1/R

W=W0+T*(W1+T*(W2+T*(W3+T*(W4+T*(W5+T*W6))))

RETURN

END

SUBROUTINE CHAIN(FCHAIN,GCHAIN,IGO,RZ,RZDIF,N,L,M,KS,P,C)

C DETERMINES THE VALUES OF THE CONTINUED FRACTION INDICES AT WHICH

C CHAIN. LET THE SERIES COEFFICIENTS BE G(J), J=0,1,2,... . OUR

C CONTINUED FRACTIONS COME FROM THE EQUATION ALPHA(J)*G(J-1)/G(J)+

C BETA(J)+GAMMA(J)*G(J+1)/G(J)=0 FOR A PARTICULAR CHOICE OF J. THE

C ARE SEVERAL INTERDEPENDENT CONSIDERATIONS IN CHOOSING A SUITABLE

C AT WHICH TO CHAIN. WE DO NOT WANT THE RATIOS G(J+1)/G(J) OR

C G(J-1)/G(J) TO BE TOO LARGE, AND ESPECIALLY MUST AVOID HAVING G(J)

C MINIMUM VALUES OF FCHAIN AND GCHAIN ARE SPECIFIED BY THE NUMBER OF

C NODES IN X(XI) AND Y(ETA) AND FROM CONSIDERATION OF THE LIMITING

C FORMS OF THE SERIES EXPANSIONS AT R=0 AND INFINITY. LET A(I) AND

C B(I) DENOTE THE NUMERATORS AND DENOMINATORS OF THE CONTINUED

C FRACTIONS. AT THE END OF THE FORWARD EVALUATION, THE VALUE IS =

C A(J-1)/A(J). WHEN WE BEGIN THE BACKWARD EVALUATION, WE HAVE B(0)=

C AND B(1)=BETA(J+1)*A(J-1), AND WE DO NOT WANT THIS OR SUBSEQUENT

C DENOMINATORS TO BE ZERO.

IMPLICIT REAL*8(A-H,O-Z)

INTEGER FCHAIN,GCHAIN

C THE CONTINUED FRACTION FOR X(XI) -

J=N-L-1

```

        SIGMA=0.5D0*RZ/P-M-1
        A=2*P*SIGMA+(M+1)*(M+SIGMA)-C
        B=2*(SIGMA-P-P)
1      J=J+1
        ALPHA=(J-1-SIGMA)*(J-1-SIGMA-M)
        BETA=-2*J*J+J*B+A
        IF(DABS(ALPHA).GT.DABS(BETA)) GO TO 1
        GCHAIN=J-1
C     ALTHOUGH THE COEF. RATIOS G(J+1)/G(J) FOR THE OUTER EQUATION EVEN-
C     TULLY BECOME VERY CLOSE TO 1 AND ASYMPTOTIC CONVERGENCE IS
C     EXCRUCIATINGLY SLOW, INITIAL CONVERGENCE IS VERY RAPID. I HAVE
C     FOUND THAT FOR J IN THE CHAINING REGION, JUST THE LEAD TERM FROM T
C     CONTINUED FRACTION G(J+1)/G(J)=-ALPHA(J+1)/BETA(J+1)-... GIVES A
C     REASONABLE APPROXIMATION. WE CHAIN WHEN THIS RATIO BECOMES LESS
C     THAN 1, INDICATING THE ONSET OF CONVERGENCE. SINCE X(XI) HAS N-L-
C     NODES, WE NEED AT LEAST THIS MANY TERMS IN THE EXPANSION.
        IF(IGO.NE.3) GO TO 3
        J=KS
C     HETERONUCLEAR LARGE R EXPANSION FOR Y(ETA) -
        A=C-2*P*(M+1)+RZDIF-M*(M+1)
2      J=J+1
        ALPHA=2*P*(J+M)-RZDIF
        BETA=A-4*P*J-J*(J+1+M+M)
        IF(DABS(ALPHA).GT.DABS(BETA)) GO TO 2
        FCHAIN=J-1
        RETURN
3      J=L
        IF(IGO.EQ.1) GO TO 5
C     HOMONUCLEAR EXPANSION FOR Y(ETA)
4      J=J+2
        ALPHA=P*P*(J-1-M)*(J-M)/((J+J-3)*(J+J-1))

```



```

        BETA=C-J*(J+1)-2*P*P*(M*M+J*(J+1)-1)/((J+J+3)*(J+J-1))
        IF(DABS(ALPHA).GT.DABS(BETA)) GO TO 4
        FCHAIN=J-2
C     THE RATIO OF ALPHA/BETA IS NOT AS ACCURATE AN APPROX. FOR THE COEFF
C     RATIO IN THIS CASE AS IT IS IN THE TWO ABOVE CASES, BUT IT REALLY
C     DOES NOT MATTER IF IT IS OFF BY A FACTOR OF 2 OR SO FOR R LARGE AND
C     WE CHAIN AT J=6 INSTEAD OF 8. IF WE KNEW P AND C ACCURATELY, WE
C     COULD USE FORWARD RECURRENCE. THIS HOWEVER BREAKS DOWN AT SMALL R
C     OUR ESTIMATES OF P AND C ARE NOT HIGHLY ACCURATE.
        RETURN
C
C     HETERONUCLEAR SMALL R EXPANSION FOR Y(ETA).
C     THE APPROX. COEF. RATIO FOUND USING JUST ALPHA(J+1)/BETA(J+1) IS
C     INACCURATE IN THIS CASE EXCEPT FOR P SMALL, BECAUSE THE NEXT TERM
C     IN THE DENOMINATOR, -GAMMA(J+1)*ALPHA(J+2)/BETA(J+2), CAN BE MUCH
C     LARGER IN MAGNITUDE THAN BETA(J+1). THEY CAN BE OF OPPOSITE SIGNS
C     SO WE COULD HAVE A ZERO DENOMINATOR. WE HAVE TO CHAIN AT J=L FOR
C     R SMALL, BECAUSE ONLY THE COEF. OF P(L,M) REMAINS NON-ZERO AS R GOES
C     TO ZERO. IF (1) J*(J+1).GT.C AND (2) J.GT.R*(ZA-ZB)/(2*P), THEN
C     BETA(J) AND GAMMA(J-1)*ALPHA(J) ARE BOTH NEGATIVE, AND WE CANNOT GET
C     A ZERO DENOMINATOR, B. HOWEVER, WE CANNOT IN GENERAL HAVE BOTH J=L
C     AND SATISFY (2) FOR R SMALL, AND I HAVE CHOSEN TO CHAIN SO AS TO
C     AVOID HAVING B(1)=0. ONE COULD ACCIDENTALLY GET A ZERO DENOMINATOR
C     IN THIS APPROACH. THIS WOULD PRODUCE A DIVIDE CHECK MESSAGE, BUT
C     IT WOULD BE ACCEPTABLE PROVIDED IT DID NOT OCCUR BECAUSE B WAS CONVERG
C     GING TO 0, I.E., TWO SUCCESSIVE 0'S.
5     J=J+1
        TEST=J*J
        IF(TEST.LT.C) GO TO 5
        FCHAIN=J-1
        RETURN

```

END

Γ.3 CI

```

subroutine Psi0_PsiK
implicit real*8(a-h,o-z)
parameter (nelec=7,np=2*nelec,norbp=10,norbE=10,aa0=0.d0,
1 aa1=1.d0,nlower=10,nconf=10,kp=10) !physical dimension. (You can
c integer aip,ajq
dimension H(nconf,nconf),S(nconf,nconf),
1 Hpq(0:nlower),Spq(0:nlower)
c INTEGER NOUT
c REAL*8 EVAL(nconf), EVEC(nconf,nconf), DGPISP, PI
c EXTERNAL DGPISP, DGVCSP, UMACH, WRRRN
common/cpcqaipajq/cp(nconf),cq(nconf,0:nlower),
1 aip(kp,nconf),ajq(kp,nconf,0:nlower),nconfE,nconf(0:nlower),
1 kE(kp),k(kp,0:nlower)
common/cESlower/HFi(0:nlower),SFi(0:nlower),
1 SqFi(0:nlower),EFi(0:nlower)
common/cnorbE/norbE,nlower
common/cp012/p0new(nelec,nelec,0:nlower),
1 p1new(nelec,nelec,0:nlower),
1 p2new(nelec,nelec,nelec,nelec,0:nlower)

do ilower=0,nlower
do ielec=1,norbE
do jelec=1,norbE
do kelec=1,norbE
do lelec=1,norbE

```

```
p0new(ielec, jelec, ilower)=aa0
p1new(ielec, jelec, ilower)=aa0
p2new(ielec, jelec, kelec, lelec, ilower)=aa0
end do
end do
end do
end do
end do

ilower=0
HHpq=aa0
SSpq=aa0
do ip=1, nconfE
do jq=1, ip
do i=1, kE(ip)
do j=1, kE(jq)
call Hleftright(i, ip, j, jq, ilower, Hij, Sij)
ie=2-Kdelta(ip, jq)
HHpq=HHpq+aip(i, ip)*ajq(j, jq, ilower)*Hij*ie
SSpq=SSpq+aip(i, ip)*ajq(j, jq, ilower)*Sij*ie
print*, '{p, q, i, j}=',
1 '{', ip, ', ', jq, ', ', i, ', ', j, '}'
print*, 'Hij=', Hij
print*, 'Sij=', Sij
end do
end do
Hpq(0)=HHpq
Spq(0)=SSpq

H(ip, jq)=Hpq(0)
```

```
S(ip, jq)=Spq(0)
H(jq, ip)=Hpq(0)
S(jq, ip)=Spq(0)

print*, 'p, q, Hpq=', ip, jq, Hpq
print*, 'p, q, Spq=', ip, jq, Spq
write(77, *) 'p, q, Hpq=', ip, jq, Hpq
write(77, *) 'p, q, Spq=', ip, jq, Spq
end do !jq=1, ip
end do !ip=1, nconfE
C iroot=1
C call DesiredRoot(nconfp, nconfE, H, S, iroot, E, cp, cn)
c normalize the coefficients, such as <Psi|Psi>=1
C do i=1, nconfE
C cq(i, 0)=cp(i)
C end do
c find <Psi|Psi>
sumS=aa0
do ip=1, nconfE
do jq=1, ip
ie=2-Kdelta(ip, jq)
sumS=sumS+cp(ip)*cq(jq, 0)*S(ip, jq)*ie
end do
end do
C anorm=aa1/sqrt(sumS)
C do ip=1, nconfE
C cp(ip)=cp(ip)*anorm
C end do
C do jq=1, nconfE
C cq(jq, 0)=cq(jq, 0)*anorm
C end do
```

```
sumH=aa0
do ip=1,nconfE
do jq=1,ip
ie=2-Kdelta(ip,jq)
sumH=sumH+cp(ip)*cq(jq,0)*H(jq,ip)*ie
end do
end do
c an prothesw ta pragmatika cp tha vgalei to E
HFi(ilower)=sumH
SFi(ilower)=sumS
SqFi(ilower)=sqrt(SFi(ilower))
c ilower=0 is the excited
print*, 'ilower, H=<Psi|H|Psi>=', ilower, HFi(ilower)
print*, 'ilower, S=<Psi|Psi>=', ilower, SFi(ilower)

return
```

```
c do ilower=1,nlower
c sumPH=aa0
c sumPS=aa0
c do ip=1,nconfE
```

```
c sumQH=aa0
c sumQS=aa0
c do jq=1,nconf(ilower)
c sumIH=aa0
c sumIS=aa0
c do i=1,kE(ip)
c sumJH=aa0
c sumJS=aa0
c do j=1,k(jq,ilower)
c call Hleftright(i,ip,j,jq,ilower,Hij,Sij)
c sumJH=sumJH+aip(i,ip)*ajq(j,jq,ilower)*Hij
c sumJS=sumJS+aip(i,ip)*ajq(j,jq,ilower)*Sij
c print*,'{p,q,i,j}=',
c 1 '{',ip,',',',jq,',',',i,',',',j','}'
c print*,'Hij=',Hij
c print*,'Sij=',Sij
c end do
c sumIH=sumIH+sumJH
c sumIS=sumIS+sumJS
c end do
c Hpq(ilower)=sumIH
c Spq(ilower)=sumIS
c
c if (ilower.eq.0) then
c
c H(ip,jq)=Hpq(0)
c S(ip,jq)=Spq(0)
c H(jq,ip)=Hpq(0)
c S(jq,ip)=Spq(0)
c
c end if
```

```
c print*, 'p, q, Hpq=', ip, jq, Hpq
c print*, 'p, q, Spq=', ip, jq, Spq
c write(77, *) 'p, q, Hpq=', ip, jq, Hpq
c write(77, *) 'p, q, Spq=', ip, jq, Spq
c sumQH=sumQH+cp(ip)*cq(jq, ilower)*Hpq(ilower)
c sumQS=sumQS+cp(ip)*cq(jq, ilower)*Spq(ilower)
c end do !jq=1, ip
c sumPH=sumPH+sumQH
c sumPS=sumPS+sumQS
c end do !ip=1, nconfE
c if (ilower.eq.0)then
c iroot=1
c call DesiredRoot(nconfp, nconfE, H, S, iroot, E, c, cn)
cc normalize the coefficients, such as <Psi|Psi>=1
c do i=1, nconfE
c cp(i, 0)=c(i)
c cq(i, 0)=c(i)
c end do
cc find <Psi|Psi>
c sumPS=aa0
c do ip=1, nconfE
c do jq=1, ip
c sumQS=sumQS+cp(ip)*cq(jq, ilower)*Spq(ilower)
c sumPS=sumPS+sumQS
c end do
c end do
c end if
c
c HFi(ilower)=sumPH
c SFi(ilower)=sumPS
c SqFi(ilower)=sqrt(SFi(ilower))
```

```

cc ilower=0 is the excited
c print*, 'ilower, H=<Psi|H|Psi>=', ilower, HFi(ilower)
c print*, 'ilower, S=<Psi|Psi>=', ilower, SFi(ilower)
c
c return
c end do !tou ilower

end

```

Γ.3.1 Αναμενόμενη τιμή και επικάλυψη

```

C nelec=number of electrons, nn=numer of sums=2*nelec
c-----
subroutine Hleftright(ii, ip, jj, jq, ilower, Hij, Sij) !(Nn do-loops <= Np)
implicit real*8 (a-h, o-z)
parameter (nelecp=7, np=2*nelecp, norbp=10, norbpE=10, aa0=0.d0,
1 nlowerp=10, nconfp=10, kp=10) !physical dimension. (You can increase
dimension t(np), ic(np) !, il(nelecp), ir(nelecp)
common/cp012/p0new(nelecp, nelecp, 0:nlowerp),
1 p1new(nelecp, nelecp, 0:nlowerp),
1 p2new(nelecp, nelecp, nelecp, nelecp, 0:nlowerp)
character*3 ffile
common/cus/isl(nelecp, kp, nconfp), isr(nelecp, kp, nconfp, 0:nlowerp),
1 iul(nelecp, kp, nconfp), iur(nelecp, kp, nconfp, 0:nlowerp)
common/cuss/isll(norbp), isrr(norbp),
1 il(nelecp), ir(nelecp), iull(norbp),
1 iurr(norbp)
common/cmsgns/msgnl(nelecp, kp, nconfp),
1 msgnr(nelecp, kp, nconfp, 0:nlowerp)
common/cmsgnss/msgnll(norbp), msgnr(norbp)

```



```

common/czszdfgnfngG2/zs_2 (norbp) , zd_2 (norbp) , Rjump_2 (norbp) ,
1p_2 (norbp) , f_2 (0:50, norbp) , g_2 (0:50, norbp) , nf_2 (norbp) , ng_2 (norbp) ,
    1 N_2 (norbp) , L_2 (norbp) , M_2 (norbp) !excited
common/czszdfgnfngE2/zs_2E (norbp) , zd_2E (norbp) , Rjump_2E (norbp) , !2e, ex
1p_2E (norbp) , f_2E (0:50, norbp) , g_2E (0:50, norbp) , nf_2E (norbp) ,
    1 ng_2E (norbp) , N_2E (norbp) , L_2E (norbp) , M_2E (norbp)
common/czszdfgnfngG1/zs_1 (norbp) , zd_1 (norbp) , Rjump_1 (norbp) , !1e, groun
1p_1 (norbp) , f_1 (0:50, norbp) , g_1 (0:50, norbp) , nf_1 (norbp) , ng_1 (norbp) ,
    1 N_1 (norbp) , L_1 (norbp) , M_1 (norbp) !excited
common/czszdfgnfngE1/zs_1E (norbp) , zd_1E (norbp) , Rjump_1E (norbp) , !1e, ex
1p_1E (norbp) , f_1E (0:50, norbp) , g_1E (0:50, norbp) , nf_1E (norbp) ,
    1 ng_1E (norbp) , N_1E (norbp) , L_1E (norbp) , M_1E (norbp)
common/czszdfgnfngE/zsE (norbp) , zdE (norbp) , RjumpE (norbp) , pE (norbp) , !ov
1 fE (0:50, norbp) , gE (0:50, norbp) , nfE (norbp) , ngE (norbp) ,
2 NE (norbp) , LE (norbp) , ME (norbp)
common/choose/smv, ener, ichoice, imchf, jvirt, icrea, jvorb, jvconf
1 , jvelec
common/czszdfgnfngG/zs_0 (norbp) , zd_0 (norbp) ,
1 Rjump_0 (norbp) , p_0 (norbp) , !ovlp, ground
1 f_0 (0:50, norbp) , g_0 (0:50, norbp) , nf_0 (norbp) , ng_0 (norbp) , !and lower
2 N_0 (norbp) , L_0 (norbp) , M_0 (norbp) !excited
common/czazbG/za_0 (norbp) , zb_0 (norbp) !ground
common/cicenG0/icen_0 (norbp) , lstart_0 (norbp) !ground
common/cicenG1/icen_1 (norbp) , lstart_1 (norbp) !ground
common/cicenG2/icen_2 (norbp) , lstart_2 (norbp) !ground

common/czszdfgnfng/zs (norbp, 0:nlowerp) , zd (norbp, 0:nlowerp) ,
1 Rjump (norbp, 0:nlowerp) , p (norbp, 0:nlowerp) , !ovlp, ground
1 f (0:50, norbp, 0:nlowerp) , g (0:50, norbp, 0:nlowerp) ,
    1 nf (norbp, 0:nlowerp) , ng (norbp, 0:nlowerp) , !and lower than
2 N (norbp, 0:nlowerp) , L (norbp, 0:nlowerp) , M (norbp, 0:nlowerp)

```

```

common/czazbG/za(norbp,0:nlowerp),zb(norbp,0:nlowerp) !ground
common/cicen/icen(norbp,0:nlowerp),lstart(norbp,0:nlowerp)
!ground
common/cnorbg/norb(0:nlowerp)
common/cnorbe/norbE,nlower
common/czazbE/zaE(norbp),zbE(norbp) !excited
common/czoldE/zaoldE(norbp),zboldE(norbp) !excited
common/cicenE/icenE(norbp),lstartE(norbp) !excited
common/cnelec/nelec
common/ciu/iu40,iu80

```

```

real*8 ag(0:50),af(0:50)
character iu*(3*np)
dimension fact(nelec)
data fact/1.d0,2.d0,6.d0,24.d0,120.d0,720.d0,
1 5040.d0/!,40320.d0,362880.d0,3628800.d0

```

c lower:is the ground or excited state below the excited state in que
c if lower=0 -->ground state,lower=1 -->1st excited and so on!

```

do ielec=1,nelec
iull(ielec)=iul(ielec,ii,ip)
isll(ielec)=isl(ielec,ii,ip)
msgnll(ielec)=msgnl(ielec,ii,ip)
end do
do jelec=1,nelec
iurr(jelec)=iur(jelec,jj,jq,ilower)
isrr(jelec)=isr(jelec,jj,jq,ilower)
msgnrr(jelec)=msgnr(jelec,ii,jq,ilower)
end do

```

```
zs_2E=zsE
zd_2E=zdE
Rjump_2E=RjumpE
p_2E=pE
f_2E=fE
g_2E=gE
nf_2E=nfE
ng_2E=ngE
N_2E=NE
L_2E=LE
M_2E=ME
```

```
zs_1E=zsE
zd_1E=zdE
Rjump_1E=RjumpE
p_1E=pE
f_1E=fE
g_1E=gE
nf_1E=nfE
ng_1E=ngE
N_1E=NE
L_1E=LE
M_1E=ME
```

```
do iorb=1,norb(ilower)
zs_0(iorb)=zs(iorb,ilower)
zd_0(iorb)=zd(iorb,ilower)
Rjump_0(iorb)=Rjump(iorb,ilower)
```

```
p_0(iorb)=p(iorb,ilower)
nf_0(iorb)=nf(iorb,ilower)
ng_0(iorb)=ng(iorb,ilower)
do k=0,nf_0(iorb)
f_0(k,iorb)=f(k,iorb,ilower)
enddo
do k=0,ng_0(iorb)
g_0(k,iorb)=g(k,iorb,ilower)
enddo
N_0(iorb)=N(iorb,ilower)
L_0(iorb)=L(iorb,ilower)
M_0(iorb)=M(iorb,ilower)

icen_0(iorb)=icen(iorb,ilower)
lstart_0(iorb)=lstart(iorb,ilower)

zs_1(iorb)=zs(iorb,ilower)
zd_1(iorb)=zd(iorb,ilower)
Rjump_1(iorb)=Rjump(iorb,ilower)
p_1(iorb)=p(iorb,ilower)
nf_1(iorb)=nf(iorb,ilower)
ng_1(iorb)=ng(iorb,ilower)
do k=0,nf_1(iorb)
f_1(k,iorb)=f(k,iorb,ilower)
enddo
do k=0,ng_1(iorb)
g_1(k,iorb)=g(k,iorb,ilower)
enddo
N_1(iorb)=N(iorb,ilower)
L_1(iorb)=L(iorb,ilower)
M_1(iorb)=M(iorb,ilower)
```

```
icen_1(iorb)=icen(iorb, ilower)
lstart_1(iorb)=lstart(iorb, ilower)
```

```
zs_2(iorb)=zs(iorb, ilower)
zd_2(iorb)=zd(iorb, ilower)
Rjump_2(iorb)=Rjump(iorb, ilower)
p_2(iorb)=p(iorb, ilower)
nf_2(iorb)=nf(iorb, ilower)
ng_2(iorb)=ng(iorb, ilower)
do k=0, nf_2(iorb)
f_2(k, iorb)=f(k, iorb, ilower)
enddo
do k=0, ng_2(iorb)
g_2(k, iorb)=g(k, iorb, ilower)
enddo
N_2(iorb)=N(iorb, ilower)
L_2(iorb)=L(iorb, ilower)
M_2(iorb)=M(iorb, ilower)
```

```
icen_2(iorb)=icen(iorb, ilower)
lstart_2(iorb)=lstart(iorb, ilower)
end do
```

```
c
```

```
c Variable-dimensional nested do-loops: (nn do-loops) nn <= np logical
```

```
c N.C. Bacalis, 2004
```

```
c National Hellenic Research Foundation, Athens, Greece
```

```
c
```

```

c do i1=1,m1 (or t1=ti1,tf1,ts1)
c do i2=1,m2 (or t2=ti2,tf2,ts2)
c ...
c do i_nn = 1, m_nn
c [...(nn do-loops)...] output(i1,i2,...i_nn)=f(t1,t2,...t_nn)
c end do
c ...
c end do
c end do
c
c The central idea is that in the innermost heart of the nested do-loops
c the {i_1, i_2, ..., i_n} indexes have some values, e.g. {5, 3, ..., 1},
c and our job depends, of course, on these values, but does not care
c we got these values, either by nested do-loops or by other means.
c So, instead of doing nested do-loops, we produce all sets of values
c in increasing order (by Nbasn) and we pass each set of index values to
c Nbasn. Nbasn produces all possible sets of indexes blindly. If a set is not
c i.e. out of our range, we just disregard it and get the next set of
c indexes.
c In order to treat variable range, do x=xi,xf,xs (instead of do i=0,n)
c we apply a linear transformation from [0...i...n] <--> [xi...x...xf]
c  $x=xi+(xf-xi)/(n-0)*(i-0)$  or inversely  $i=0+(n-0)/(xf-xi)*(x-xi)$ 
c
c define m=max(m1,m2,...,m_nn), produce the maximum, and disregard un-
c
c We want say nn= 3 do-loops
c do k1=1,nk1 (say k1=1,5)
c   do k2=1,nk2 (say k2=1,2)
c     do k3=1,nk3 (say k3=1,7)
c   call f(k1,k2,k3)
c     enddo !k3
c   enddo !k2

```

```

c enddo !k1
c So, nn=3 do-loops, m=Max[5,2,7]=7
c We store {5,2,7} in array nkj(i),i=1,nn(=3)
c ic is a triad: {ic(1), ic(2), ic(3)}
c
c-----
! 10 continue
c print *, 'read *, nelec'
c 1 !(where nn=how many do-loops (each: k_2=0,m_2), m=Max[m_1...,m_nn])
c read *, nelec
mi=nelec
nn=nelec*2
print *, 'i,ic,m values,nn do-loops',m,nn
nmax=mi**nn
Hij=aa0
Sij=aa0
do ni=0,nmax-1
c first produce the values of the indices i1,i2,...i_nn at each step
call nbasm(ni,mi,i,np,nn,ic) ! array ic contains the present indices
print '(3i6,14i3)', nmax,i,ni,(ic(j)+1,j=nn,1,-1)
c setup the function of indices f(i1,i2,...i_nn) of this step (i.e. c
c Job, the user-supplied subroutine, produces the abscissae in array
write(iu,'(<np>i3)') (ic(j)+1,j=nn,1,-1)
read(iu,*) (il(j),j=1,nelec),(ir(j),j=1,nelec)
LvCl=Levi_Civita(np,nelec,il)
if (LvCl.eq.0) goto 1
LvCr=Levi_Civita(np,nelec,ir)
if (LvCr.eq.0) goto 1

call CI(nelec,ilower,termhg,Pzero)
print*, 'ni, LvCl, LvCr, termhg, Pzero', ni, LvCl, LvCr, termhg, Pzero

```

```
Hij=Hij+LvCl*LvCr*termhg
Sij=Sij+LvCl*LvCr*Pzero
print*, 'Hij=', Hij
print*, 'Sij=', Sij
print*, 'left orbs:', iull
print*, 'right orbs:', iurr
  1 continue
end do
Hij=Hij/fact(nelec)
Sij=Sij/fact(nelec)
print*, 'Hij=', Hij
print*, 'Sij=', Sij
print*, 'left orbs:', iull
print*, 'right orbs:', iurr
! goto 10
return
stop
end

subroutine nbasm(nn,m,i,np,ni,ic)
dimension ic(np)
parameter (null=0)
do 1 i=1,ni
  1 ic(i)=null
i=0
n=nn
  2 continue
mr=mod(n,m)
n=(n-mr)/m
c print *, n,mr, '      new n,remainder'
```



```

i=i+1
ic(i)=mr
if (n.ne.0) goto 2
end

```

Levi-Civita

```

function Levi_Civita(np,nelec,ip)
c n=nelec,ip=permutation of n electrons

```

```

integer ip(np)
n=nelec
iopt=0
LvC=1
call perm_cycle ( n, iopt, ip, isgn, ncycle,LvC )
print*, ip
print*, isgn
Levi_Civita=isgn*LvC
end

```

```

subroutine perm_cycle ( n, iopt, p, isgn, ncycle,LvC )

```

```

c*****

```

```

c

```

```

cc PERM_CYCLE analyzes a permutation.

```

```

c

```

```

c Discussion:

```

```

c

```

```

c The routine will count cycles, find the sign of a permutation,

```

```
c   and tag a permutation.
c
c   Example:
c
c   Input:
c
c       N = 9
c       IOPT = 1
c       P = 2, 3, 9, 6, 7, 8, 5, 4, 1
c
c   Output:
c
c       NCYCLE = 3
c       ISGN = +1
c       P = -2, 3, 9, -6, -7, 8, 5, 4, 1
c
c   Licensing:
c
c   This code is distributed under the GNU LGPL license.
c
c   Modified:
c
c       25 January 2007
c
c   Author:
c
c       Original FORTRAN77 version by Albert Nijenhuis, Herbert Wilf.
c       This FORTRAN77 version by John Burkardt.
c
c   Reference:
c
```

```
c   Albert Nijenhuis, Herbert Wilf,
c   Combinatorial Algorithms for Computers and Calculators,
c   Academic Press, 1978,
c   ISBN: 0-12-519260-6,
c   LC: QA164.N54.
c
c Parameters:
c
c   Input, integer N, the number of objects being permuted.
c
c   Input, integer IOPT, requests tagging.
c   0, the permutation will not be tagged.
c   1, the permutation will be tagged.
c
c   Input/output, integer P(N). On input, P describes a
c   permutation, in the sense that entry I is to be moved to P(I).
c   If IOPT = 0, then P will not be changed by this routine.
c   If IOPT = 1, then on output, P will be "tagged". That is,
c   one element of every cycle in P will be negated. In this way,
c   a user can traverse a cycle by starting at any entry I1 of P
c   which is negative, moving to I2 = ABS(P(I1)), then to
c   P(I2), and so on, until returning to I1.
c
c   Output, integer ISGN, the "sign" of the permutation, which is
c   +1 if the permutation is even, -1 if odd. Every permutation
c   may be produced by a certain number of pairwise switches.
c   If the number of switches is even, the permutation itself is
c   called even.
c
c   Output, integer NCYCLE, the number of cycles in the permutation.
c
```

```
implicit none

integer n

integer base
integer i
integer i1
integer i2
integer LvC
integer iopt
integer is
integer isgn
integer ncycle
integer p(n)

base = 1
call perm_check ( n, p, base, LvC )
if (LvC.eq.0) return

is = 1
ncycle = n

do i = 1, n

    i1 = p(i)

10    continue

    if ( i .lt. i1 ) then
        ncycle = ncycle - 1
        i2 = p(i1)
```

```

        p(i1) = -i2
        i1 = i2
        go to 10
    end if

    if ( iopt .ne. 0 ) then
        is = -sign ( 1, p(i) )
    end if

    p(i) = sign ( p(i), is )

end do

isgn = 1 - 2 * mod ( n - ncycle, 2 )

return
end

subroutine perm_check ( n, p, base, LvC )

c*****
c
cc PERM_CHECK checks that a vector represents a permutation.
c
c Discussion:
c
c The routine verifies that each of the integers from BASE to
c to BASE+N-1 occurs among the N entries of the permutation.
c

```

```
c      Set the input quantity BASE to 0, if P is a 0-based permutation,  
c      or to 1 if P is a 1-based permutation.  
c  
c      Licensing:  
c  
c      This code is distributed under the GNU LGPL license.  
c  
c      Modified:  
c  
c      03 June 2009  
c  
c      Author:  
c  
c      John Burkardt  
c  
c      Parameters:  
c  
c      Input, integer N, the number of entries.  
c  
c      Input, integer P(N), the array to check.  
c  
c      Input, integer BASE, the index base.  
c  
c      Output, integer IERROR, error flag.  
c      0, the array represents a permutation.  
c      nonzero, the array does not represent a permutation. The smallest  
c      missing value is equal to IERROR.  
c  
c      implicit none  
  
c      integer n
```

```
integer LvC
    integer base
    integer find
    integer ierror
    integer p(n)
    integer seek

    ierror = 0

    do seek = base, base + n - 1

        ierror = 1

        do find = 1, n
            if ( p(find) .eq. seek ) then
                ierror = 0
                go to 10
            end if
        end do

10    continue

        if ( ierror .ne. 0 ) then
            write ( *, '(a)' ) ' '
            write ( *, '(a)' ) 'PERM_CHECK - Fatal error!'
            write ( *, '(a)' ) ' The input array does not represent'
            write ( *, '(a)' ) ' a proper permutation.'
            LvC=0
        end if

    end do
```

```

return
end

```

Ανάπτυξη της Slater

```

subroutine CI(nelec, ilower, termhg, Pzero)
implicit real*8 (a-h, o-z)
parameter (nelecp=7, np=2*nelecp, norbp=10, norbpE=10, aa0=0.d0,
1 nlowerp=10, nconfp=10, kp=10) !physical dimension. (You can increase
external ovlp, e_1, e_2
dimension

1 p0old(nelecp, nelecp, 0:nlowerp),
   1 p1old(nelecp, nelecp, 0:nlowerp),
1 p2old(nelecp, nelecp, nelecp, nelecp, 0:nlowerp)
common/cp012/p0new(nelecp, nelecp, 0:nlowerp),
   1 p1new(nelecp, nelecp, 0:nlowerp),
   1 p2new(nelecp, nelecp, nelecp, nelecp, 0:nlowerp)
common/cuss/isl(norbp), isr(norbp),
1 il(nelecp), ir(nelecp), iul(norbp),
   1 iur(norbp)
common/czoldE/zaoldE(norbp), zboldE(norbp) !excited
common/czazbE/zaE(norbp), zbE(norbp) !excited
common/czazbG/za(norbp, 0:nlowerp), zb(norbp, 0:nlowerp) !ground
common/czold/zaold(norbp, 0:nlowerp), zbold(norbp, 0:nlowerp)
!ground
common/cilower/iilower
termhg=aa0

do ielec=1, nelec

```



```

isi=isl(il(ielec))+isr(ir(ielec))
if (isi.eq.0) then
termhg=aa0
Pzero=aa0
return
end if
end do
c ypologismos:ovlp*ovlp(twn ypoleipwn e)
c ypologismos:1e*ovlp(twn ypoleipwn e)
Pzero=aa0
Pone=aa0
do ielec=1,nelec
if( zaE(iul(il(ielec))).eq.zaoldE(iul(il(ielec)))
1 .and.zbE(iul(il(ielec))).eq.zboldE(iul(il(ielec)))
1 )then
if(p0old(iul(il(ielec)),iur(ir(ielec)),ilower).eq.aa0)then
p0=ovlp(iul(il(ielec)),iur(ir(ielec)))
p0new(iul(il(ielec)),iur(ir(ielec)),ilower)=p0
p0old(iul(il(ielec)),iur(ir(ielec)),ilower)=p0
else
p0new(iul(il(ielec)),iur(ir(ielec)),ilower)=
1 p0old(iul(il(ielec)),iur(ir(ielec)),ilower)
print*, 'auto antigrafthke iul,iur,ilower=',
1 iul(il(ielec)),iur(ir(ielec)),ilower
p0=p0new(iul(il(ielec)),iur(ir(ielec)),ilower)
end if
if(p1old(iul(il(ielec)),iur(ir(ielec)),ilower).eq.aa0)then

```

```
p1=e_1(iul(il(ielec)),iur(ir(ielec)))
p1new(iul(il(ielec)),iur(ir(ielec)),ilower)=p1
p1old(iul(il(ielec)),iur(ir(ielec)),ilower)=p1
else

p1new(iul(il(ielec)),iur(ir(ielec)),ilower)=
1 p1old(iul(il(ielec)),iur(ir(ielec)),ilower)
print*, 'auto antigrafthke iul,iur,ilower=',
1 iul(il(ielec)),iur(ir(ielec)),ilower

p1=p1new(iul(il(ielec)),iur(ir(ielec)),ilower)
end if
else
if(p0new(iul(il(ielec)),iur(ir(ielec)),ilower).eq.aa0)then
p0=ovlp(iul(il(ielec)),iur(ir(ielec)))
p0new(iul(il(ielec)),iur(ir(ielec)),ilower)=p0
p0old(iul(il(ielec)),iur(ir(ielec)),ilower)=p0
else
p0=p0new(iul(il(ielec)),iur(ir(ielec)),ilower)
end if

if(p1new(iul(il(ielec)),iur(ir(ielec)),ilower).eq.aa0)then
p1=e_1(iul(il(ielec)),iur(ir(ielec)))
p1new(iul(il(ielec)),iur(ir(ielec)),ilower)=p1
p1old(iul(il(ielec)),iur(ir(ielec)),ilower)=p1
else
p1=p1new(iul(il(ielec)),iur(ir(ielec)),ilower)
end if
end if
print*, 'iul,iur,p0=', iul(il(ielec)),iur(ir(ielec)),p0
```

```

print*, 'iul, iur, p1=', iul(il(ielec)), iur(ir(ielec)), p1
do jelec=1, nelec
  if (jelec.ne.ielec) then
    if( zaE(iul(il(jelec))).eq.zaoldE(iul(il(jelec)))
    1 .and.zbE(iul(il(jelec))).eq.zboldE(iul(il(jelec)))

        1 )then
      if(p0old(iul(il(jelec)), iur(ir(jelec)), ilower).eq.aa0) then
        pij=ovlp(iul(il(jelec)), iur(ir(jelec)))
        p0new(iul(il(jelec)), iur(ir(jelec)), ilower)=pij
        p0old(iul(il(jelec)), iur(ir(jelec)), ilower)=pij
      else
        p0new(iul(il(jelec)), iur(ir(jelec)), ilower)=
        1 p0old(iul(il(jelec)), iur(ir(jelec)), ilower)
        print*, 'auto antigrafthke iul, iur, ilower=',
        1 iul(il(jelec)), iur(ir(jelec)), ilower

        pij=p0new(iul(il(jelec)), iur(ir(jelec)), ilower)
      end if
    else
      if(p0new(iul(il(jelec)), iur(ir(jelec)), ilower).eq.aa0) then
        pij=ovlp(iul(il(jelec)), iur(ir(jelec)))
        p0new(iul(il(jelec)), iur(ir(jelec)), ilower)=pij
        p0old(iul(il(jelec)), iur(ir(jelec)), ilower)=pij
      else
        pij=p0new(iul(il(jelec)), iur(ir(jelec)), ilower)
      end if
    end if
    print*, 'iul, iur, ilower, pij=', iul(il(jelec)), iur(ir(jelec))
    1 , ilower, pij
    p0=p0*pij

```

```

print*, 'ovlp(1)*ovlp(2)=' ,
1 '<il1(i)|ir1(i)><il2(j)|ir2(j)>=' , p0

p1=p1*pij
print*, 'h(1)*ovlp(2)=' ,
1 '<il1(i)|h(i)|ir1(i)><il2(j)|ir2(j)>=' , p1
print*, 'ielec, jelec=' , ielec, jelec
end if
end do !jelec
Pzero=p0
Pone=Pone+p1
end do !ielec

c  apo edw kai katw ypologismos:2e*ovlp
Ptwo=aa0
do ielec=1,nelec-1
do jelec=ielec+1,nelec

if( zaE(iul(il(ielec))).eq.zaoldE(iul(il(ielec)))
1 .and.zbE(iul(il(ielec))).eq.zboldE(iul(il(ielec)))
1 .and.zaE(iul(il(jelec))).eq.zaoldE(iul(il(jelec)))
1 .and.zbE(iul(il(jelec))).eq.zboldE(iul(il(jelec)))

1 )then
if(p2old(iul(il(ielec)),iul(il(jelec)),
1 iur(ir(ielec)),iur(ir(jelec)),ilower).eq.aa0.and.
1 p2old(iul(il(jelec)),iul(il(ielec)),iur(ir(jelec)),
1 iur(ir(ielec)),ilower).eq.aa0)then
p2=e_2(iul(il(ielec)),iul(il(jelec)),

```

```

1 iur(ir(ielec),iur(ir(jelec)))
p2new(iul(il(ielec)),iul(il(jelec))),
1 iur(ir(ielec),iur(ir(jelec)),ilower)=p2
p2old(iul(il(ielec)),iul(il(jelec))),
1 iur(ir(ielec),iur(ir(jelec)),ilower)=p2
p2new(iul(il(jelec)),iul(il(ielec))),
1 iur(ir(jelec),iur(ir(ielec)),ilower)=p2
      p2old(iul(il(jelec)),iul(il(ielec))),
1 iur(ir(jelec),iur(ir(ielec)),ilower)=p2
p2new(iur(ir(ielec)),iur(ir(jelec))),
1 iul(il(ielec)),iul(il(jelec)),ilower)=p2
p2old(iur(ir(ielec)),iur(ir(jelec))),
1 iul(il(ielec)),iul(il(jelec)),ilower)=p2
else

p2new(iul(il(ielec)),iul(il(jelec))),
1 iur(ir(ielec),iur(ir(jelec)),ilower)=
1 p2old(iul(il(ielec)),iul(il(jelec))),
1 iur(ir(ielec),iur(ir(jelec)),ilower)
print*,'auto antigrafthke iulI,iulJ,iurK,iurL,ilower=',
1 iul(il(ielec)),iul(il(jelec)),
1 iur(ir(ielec)),iur(ir(jelec)),ilower

p2=p2new(iul(il(ielec)),iul(il(jelec))),
1 iur(ir(ielec),iur(ir(jelec)),ilower)
end if
else
if(p2new(iul(il(ielec)),iul(il(jelec))),
1 iur(ir(ielec),iur(ir(jelec)),ilower).eq.aa0)then
p2=e_2(iul(il(ielec)),iul(il(jelec))),

```

```

1 iur(ir(ielec),iur(ir(jelec)))
p2new(iul(il(ielec)),iul(il(jelec))),
1 iur(ir(ielec),iur(ir(jelec)),ilower)=p2
p2old(iul(il(ielec)),iul(il(jelec))),
1 iur(ir(ielec),iur(ir(jelec)),ilower)=p2
p2new(iul(il(jelec)),iul(il(ielec))),
1 iur(ir(jelec),iur(ir(ielec)),ilower)=p2
      p2old(iul(il(jelec)),iul(il(ielec))),
1 iur(ir(jelec),iur(ir(ielec)),ilower)=p2
p2new(iur(ir(ielec)),iur(ir(jelec))),
1 iul(il(ielec)),iul(il(jelec)),ilower)=p2
p2old(iur(ir(ielec)),iur(ir(jelec))),
1 iul(il(ielec)),iul(il(jelec)),ilower)=p2
else
c      if(p2new(iul(il(ielec)),iul(il(jelec))),
c 1 iur(ir(ielec)),iur(ir(jelec)),ilower).ne.aa0)then
      p2=p2new(iul(il(ielec)),iul(il(jelec))),
1 iur(ir(ielec)),iur(ir(jelec)),ilower)
end if
end if
print*, 'g(i, j), i, j=', p2, ielec, jelec
print*, 'iulI, iulJ, iulK, iulL=', iul(il(ielec)), iul(il(jelec)),
1 iur(ir(ielec)), iur(ir(jelec))

do kelec=1, nelec
if (kelec.ne.ielec.and.kelec.ne.jelec) then
if( zaE(iul(il(kelec))).eq.zaoldE(iul(il(kelec))))
1 .and.zbE(iul(il(kelec))).eq.zboldE(iul(il(kelec)))
c 1 .and.za(iur(ir(kelec)),ilower).eq.zaold(iur(ir(kelec)),ilower)
c 1 .and.zb(iur(ir(kelec)),ilower).eq.zbold(iur(ir(kelec)),ilower)
1      )then

```

```

if (p0old(iul(il(kelec)), iur(ir(kelec)), ilower).eq.aa0) then
pij=ovlp(iul(il(kelec)), iur(ir(kelec)))
p0new(iul(il(kelec)), iur(ir(kelec)), ilower)=pij
p0old(iul(il(kelec)), iur(ir(kelec)), ilower)=pij
else

p0new(iul(il(kelec)), iur(ir(kelec)), ilower)=
1 p0old(iul(il(kelec)), iur(ir(kelec)), ilower)
print*, 'auto antigrafthke iul, iur, ilower=',
1 iul(il(kelec)), iur(ir(kelec)), ilower !edw

pij=p0new(iul(il(kelec)), iur(ir(kelec)), ilower)
end if
else
if (p0new(iul(il(kelec)), iur(ir(kelec)), ilower).eq.aa0) then
pij=ovlp(iul(il(kelec)), iur(ir(kelec)))
p0new(iul(il(kelec)), iur(ir(kelec)), ilower)=pij
p0old(iul(il(kelec)), iur(ir(kelec)), ilower)=pij
else

pij=p0new(iul(il(kelec)), iur(ir(kelec)), ilower)
end if
end if
print*, 'pij, <il3(k) | ir3(k)>=',
1 pij, iul(il(kelec)), iur(ir(kelec))
p2=p2*pij
print*, 'g(i, j)*ovlp(k) =',
1 '<il1(i) il2(j) | 1/r(i, j) | ir1(i) ir2(j)>* <il3(k) | ir3(k)>=', p2
print*, 'iulI, iulJ, iulK, iulL, pij =', iul(il(ielec)), iul(il(jelec)),
1 iur(ir(ielec)), iur(ir(jelec)), iul(il(kelec)), iur(ir(kelec))
end if

```

```

end do !kelec
Ptwo=Ptwo+p2
end do !jelec
end do !ielec
termhg=Pone+Ptwo
print*, '[<il1(i) | h(i) | ir1(i) > <il2(j) | ir2(j) > ... +',
1 ' <il1(i) il2(j) | 1/r(1,2) | ir1(i) ir2(j) > <il3(j) | ir3(j) > ...] =', termhg
print*, 'Pzero=', Pzero
print*, 'Pone=', Pone
print*, 'Ptwo=', Ptwo
print*, 'termhg=', termhg
end

```

Γ.3.2 Διαγωνοποίηση

```

c implicit real*8(a-h,o-z)
c parameter(np=10)
c dimension s(np,np),h(np,np) !overlap and hamiltonian
c 1 ,c(np),cn(np) !desired eigenvector un-normalized and normalized
c n=4
c print *, 'read *, iroot'
c read *, iroot
c s(1, 1)=0.783333d0
c s(1, 2)=-0.183333d0
c s(1, 3)=-0.15d0
c s(1, 4)=-0.116667 d0
c s(2, 1)=-0.183333d0
c s(2, 2)=0.883333 d0
c      s(2, 3)=-0.05 d0
c s(2, 4)= 0.0166667 d0
c s(3, 1)=-0.15 d0

```



```
c s(3, 2) = -0.05 d0
c s(3, 3) = 1.05 d0
c s(3, 4) = 0.15 d0
c s(4, 1) = -0.116667 d0
c s(4, 2) = 0.0166667 d0
c s(4, 3) = 0.15 d0
c s(4, 4) = 1.28333 d0

c h(1, 1) = -2.05 d0
c h(1, 2) = .816667 d0
c h(1, 3) = .183333 d0
c h(1, 4) = -0.283333 d0
c h(2, 1) = .816667 d0
c h(2, 2) = -2.45 d0
c h(2, 3) = -0.216667 d0
c h(2, 4) = -0.816667 d0
c h(3, 1) = .183333 d0
c h(3, 2) = -0.216667 d0
c h(3, 3) = -2.61667 d0
c h(3, 4) = -1.35 d0
c h(4, 1) = -0.283333 d0
c h(4, 2) = -0.816667 d0
c h(4, 3) = -1.35 d0
c h(4, 4) = -2.88333 d0

c
c call DesiredRoot(np,n,h,s,iroot,e,c,cn)
c Print *, 'iroot, e =', iroot,e
c print '(/<n>f10.5)', (c(i),i=1,n)
c print '(<n>f10.5)', (cn(i),i=1,n)
c end
```

```

subroutine DesiredRoot(np,n,h,s,iroot,e,c,cn)
implicit real*8(a-h,o-z)
c generalized eigenvalue nxn problem Hx=eSx. (e,c(n)) = (desired energy, eigenvector)
c iroot is input: 1=the lowest, 2=the 2nd root etc.
dimension s(np,np),h(np,np) !overlap and hamiltonian
1 ,d(np),x(np,np) !scratch arrays and unsorted eigenvalues, eigenvectors
1 ,indx(np),c(np),cn(np) !sorting index and desired eigenvector un-normalized

call geneig(h,s,n,np,d,x)
print '( <n>f10.5/)',(d(i),i=1,n)
do i=1,n
print '( <n>f10.5)',(x(i,j),j=1,n)
enddo
c print '( /<np>f10.5/)',d
c print '( /<np>f10.5)',x

call indexx(n,d,indx)
c print '( /<n>f10.5/)',(d(indx(i)),i=1,n)
e=d(indx(iroot))
do i=1,n
c print '( <n>f10.5)',(x(i,indx(j)),j=1,n)
cn(i)=x(i,indx(iroot))
enddo
c cn = c-normalized
sq=0
do i=1,n
sq=max(sq,cn(i)**2)
enddo
srt=sqrt(sq)
do i=1,n
c(i)=cn(i)/srt

```

```
c print *, c(i)
enddo
c print '(//<np>f10.5/)', c
end

subroutine geneig(b, a, n, np, d, x)
c Generalized eigenvalue problem bx = dax
implicit real*8(a-h, o-z)
parameter(ndim=10) !27
dimension a(np, np), p(ndim), b(np, np), c(ndim, ndim), x(np, np),
1          at(ndim, ndim), v(ndim, ndim), d(np), ax(ndim, ndim)
common/cifail/ifail
parameter(i10=10, z0d0=0.d0, z1d0=1.d0, z2d0=2.d0)

c do i=1, n
c write(111, *), (b(i, j), j=1, n)
c enddo
c print '(a)', ''
c do i=1, n
c write(111, *), (a(i, j), j=1, n)
c enddo
c rewind(111)
c do i=1, n
c read(111, *), (b(i, j), j=1, n)
c enddo
c print '(a)', ''
c do i=1, n
c read(111, *), (a(i, j), j=1, n)
c enddo

c      get the Cholesky decomposition matrix L   rf.NM.P89
```

```
call choldc(a,n,np,p)
if (ifail.eq.1) return
cc    get the inverse of L,L-1 rf. NR.P91
do i =1,n
a(i,i) = z1d0/p(i)
c p(i) = 1.d0/p(i)
do j = i+1,n
sum = z0d0
do k = i,j-1
sum = sum - a(j,k)*a(k,i)
end do
  a(j,i) = sum/p(j)
  a(i,j) = 0
end do
end do
cc    get the inverse of L,L-1 rf. NR.P91
c11   get the transpose of (L-1)T, rf. NR.P91

do i = 1,n
do j = 1,n
at(i,j) = a(j,i)
end do
end do
c write (26,*) 'a', ((a(i,j),j=1,n),i=1,n)
c write (26,*) 'at', ((at(i,j),j=1,n),i=1,n)

c    calculate c=(L-1)*b(n,n)*(L-1)T  Rf.NM.P455.(11.0.21)
do i = 1, n
do j = 1,i
ap = z0d0
do k = 1,n
```

```
bp =z0d0
do m = 1,n
  bp = bp + a(i,k)*b(k,m)*at(m,j)
end do
  ap = ap + bp
end do
c(i,j) = ap
c(j,i) = c(i,j)
end do
end do

! print *, 'call jacobi from geneig'
call jacobi(c,n,np,d,v,nrot,i)
c find the original eigenvectors  $LT^*X = v$  get  $x$   $X = (LT)^{-1}*V = (L^{-1})T$ 
do i = 1, n
do j = 1,n
as = z0d0
do k = 1,n
as = as + at(i,k)*v(k,j)
end do
ax(i,j) = as
end do
end do

c      Normalize the eigenvector
do j = 1,n
asf = z0d0
do i = 1,n
asf = asf + ax(i,j)**2
end do
do i = 1,n
x(i,j) = ax(i,j)/dsqrt(asf)
```

```
end do
end do
end

      SUBROUTINE choldc(a,n,np,p)
c Cholesky decomposition
      implicit real*8(a-h,o-z)
parameter(z0=0.d0,dm4=1.d-4,i0=0)
c      INTEGER n,np
      REAL*8 a(np,np),p(n)
c      INTEGER i,j,k
c      REAL sum
      common/cifail/ifail
! kount=kount+1
! print *, kount
ifail=i0
      do 13 i=1,n
        do 12 j=i,n
          sum=a(i,j)
          do 11 k=i-1,1,-1
            sum=sum-a(i,k)*a(j,k)
11          continue
          if(i.eq.j)then
c            if(sum.le.0.d0)pause 'choldc failed'
            if(sum.le.dm4) print*,
1 'Caution ! Two orbitals may approach each other'
            if(sum.le.z0) then
ifail=1
            print *,'choldc failed'
write(26, *) 'choldc failed'
return
```

```

end if
        p(i)=dsqrt(sum)
    else
        a(j,i)=sum/p(i)
    endif
12     continue
13     continue
    return
END

SUBROUTINE jacobi(a,n,np,d,v,nrot,i)
INTEGER n,np,nrot,NMAX
REAL*8 a(np,np),d(np),v(np,np)
PARAMETER (NMAX=500)
INTEGER i,ip,iq,j
REAL*8 c,g,h,s,sm,t,tau,theta,tresh,b(NMAX),z(NMAX)
parameter(i10=10,z0d0=0.d0,z1d0=1.d0,z2d0=2.d0)

! print *, a(1,1),a(1,2),a(2,1),a(2,2),'in jacobi'
do 12 ip=1,n
    do 11 iq=1,n
        v(ip,iq)=z0d0
11     continue
        v(ip,ip)=z1d0
12     continue
do 13 ip=1,n
    b(ip)=a(ip,ip)
    d(ip)=b(ip)
    z(ip)=z0d0
13     continue
nrot=0

```

```
do 24 i=1,50
  sm=z0d0
  do 15 ip=1,n-1
    do 14 iq=ip+1,n
      sm=sm+abs(a(ip,iq))
14    continue
15  continue
  if(sm.eq.z0d0) return
  if(i.lt.4) then
    tresh=0.2d0*sm/n**2
  else
    tresh=z0d0
  endif
  do 22 ip=1,n-1
    do 21 iq=ip+1,n
      g=100.d0*abs(a(ip,iq))
      if((i.gt.4).and.(abs(d(ip))+
1 g.eq.abs(d(ip)).and.(abs(d(iq))+g.eq.abs(d(iq)))) then
        a(ip,iq)=z0d0
      else if(abs(a(ip,iq)).gt.tresh) then
        h=d(iq)-d(ip)
        if(abs(h)+g.eq.abs(h)) then
          t=a(ip,iq)/h
        else
          theta=0.5d0*h/a(ip,iq)
          t=z1d0/(abs(theta)+dsqrt(z1d0+theta**2))
          if(theta.lt.z0d0)t=-t
        endif
        c=z1d0/dsqrt(z1d0+t**2)
        s=t*c
        tau=s/(z1d0+c)
```



```
h=t*a(ip,iq)
z(ip)=z(ip)-h
z(iq)=z(iq)+h
d(ip)=d(ip)-h
d(iq)=d(iq)+h
a(ip,iq)=z0d0
do 16 j=1,ip-1
  g=a(j,ip)
  h=a(j,iq)
  a(j,ip)=g-s*(h+g*tau)
  a(j,iq)=h+s*(g-h*tau)
16 continue
do 17 j=ip+1,iq-1
  g=a(ip,j)
  h=a(j,iq)
  a(ip,j)=g-s*(h+g*tau)
  a(j,iq)=h+s*(g-h*tau)
17 continue
do 18 j=iq+1,n
  g=a(ip,j)
  h=a(iq,j)
  a(ip,j)=g-s*(h+g*tau)
  a(iq,j)=h+s*(g-h*tau)
18 continue
do 19 j=1,n
  g=v(j,ip)
  h=v(j,iq)
  v(j,ip)=g-s*(h+g*tau)
  v(j,iq)=h+s*(g-h*tau)
19 continue
nrot=nrot+1
```

```
endif
21 continue
22 continue
do 23 ip=1,n
    b(ip)=b(ip)+z(ip)
    d(ip)=b(ip)
    z(ip)=z0d0
23 continue
24 continue
stop 'too many iterations in jacobi'
return
END

SUBROUTINE indexx(n,arr,indx)
INTEGER n,indx(n),M,NSTACK
REAL*8 arr(n)
PARAMETER (M=7,NSTACK=50)
INTEGER i,indx(i),ir,itype,j,jstack,k,l,istack(NSTACK)
REAL*8 a
do 11 j=1,n
    indx(j)=j
11 continue
jstack=0
l=1
ir=n
1 if(ir-l.lt.M)then
do 13 j=l+1,ir
    indx(j)=j
a=arr(indx(j))
do 12 i=j-1,1,-1
    if(arr(indx(i)).le.a)goto 2
```

```
        indx(i+1)=indx(i)
12      continue
        i=0
2       indx(i+1)=indx(i)
13     continue
        if(jstack.eq.0) return
        ir=istack(jstack)
        l=istack(jstack-1)
        jstack=jstack-2
else
        k=(l+ir)/2
        itemp=indx(k)
        indx(k)=indx(l+1)
        indx(l+1)=itemp
        if(arr(indx(l+1)).gt.arr(indx(ir))) then
            itemp=indx(l+1)
            indx(l+1)=indx(ir)
            indx(ir)=itemp
        endif
        if(arr(indx(l)).gt.arr(indx(ir))) then
            itemp=indx(l)
            indx(l)=indx(ir)
            indx(ir)=itemp
        endif
        if(arr(indx(l+1)).gt.arr(indx(l))) then
            itemp=indx(l+1)
            indx(l+1)=indx(l)
            indx(l)=itemp
        endif
        i=l+1
        j=ir
```

```
        indx(1)=indx(j)
        indx(j)=indx(1)
        indx(1)=indx(j)
        indx(j)=indx(1)
        jstack=jstack+2
        if(jstack.gt.NSTACK)pause 'NSTACK too small in indexx'
        if(ir-i+1.ge.j-1)then
            istack(jstack)=ir
            istack(jstack-1)=i
            ir=j-1
        else
            istack(jstack)=j-1
            istack(jstack-1)=1
            l=i
        endif
    endif
    goto 1
END
```

Γ.4 Συναρτησοειδές \mathcal{F}

```

function Functional(nlower)
implicit real*8(a-h,o-z)
parameter (nelecp=7,np=2*nelecp,norbp=10,norbpE=10,aa0=0.d0,
1 nlowerp=10,nconfp=10,kp=10) !physical dimension. (You can increase
dimension FiFn(nlowerp),FiHFn(nlowerp)
common/czazbE/zaE(norbp),zbE(norbp) !excited
common/czoldE/zaoldE(norbp),zboldE(norbp) !excited
common/cnorbE/norbE
common/cESlower/HFi(0:nlowerp),SFi(0:nlowerp),
1 SqFi(0:nlowerp),EFi(0:nlowerp)
F0F0=SFi(0)
EFn=HFi(0)/F0F0
do ilower=1,nlower
denom=sqrt(abs(F0F0))*SqFi(ilower)
FiFn(ilower)=SFi(ilower)/denom
FiHFn(ilower)=HFi(ilower)/denom
end do
sum=aa0
do ilower=1,nlower
sum=sum+FiFn(ilower)**2
end do
Paronomastis=1-sum
sum=aa0
do ilower=1,nlower
anum=(EFn*FiFn(ilower)-FiHFn(ilower))**2
denom=EFn-EFi(ilower)
sum=sum+anum/denom
end do
Arithmitis=2*sum
Functional=EFn+Arithmitis/Paronomastis

```

```
print*, 'F=', Functional
do jvirt=1, norbE
  zaoldE(jvirt)=zaE(jvirt)
  zboldE(jvirt)=zbE(jvirt)
end do
return
end
```

Γ.5 Δημιουργία Κατώτερων Καταστάσεων

```
subroutine readwrite(iuin, iuout)
  character*1000 grammi
  open (iuin)
  open (iuout)
  do i=1, 1000
    read (iuin, '(a)', err=1) grammi
    write(iuout, '(a)') trim(grammi)
  end do
  1 close (iuin)
  close(iuout)
  end
```

Παράρτημα Δ΄

Matrix Elements

Δ΄.1 Επικαλύψεως

```
function ovlp(iul,iur)
implicit real*8 (a-h,o-z)
parameter (norbp=10,nlowerp=10,aa0=0.d0)
real*8 fl(0:50),fm(0:50)
real*8 gl(0:50),gm(0:50)
common/czszdfgnfngG/zs(norbp),zd(norbp),Rjump(norbp),p(norbp),
1 f(0:50,norbp),g(0:50,norbp),nf(norbp),ng(norbp),
2 N(norbp),L(norbp),M(norbp)
common/czazbG/zaa(norbp,0:nlowerp),zbb(norbp,0:nlowerp)
common/czazb/za(norbp),zb(norbp)
common/czszdfgnfngE/zsE(norbp),zdE(norbp),RjumpE(norbp),pE(norbp),
1 fE(0:50,norbp),gE(0:50,norbp),nfE(norbp),ngE(norbp),
2 NE(norbp),LE(norbp),ME(norbp)
common/czazbE/zaE(norbp),zbE(norbp)
common/cRzpLm0/R, z1,zm, pl,pm
common/cxi/xii,xif
common/ceta/etai,etaf
common/cilower/ilower
```

```
common/cnorbE/norbE,nlower

do jvirt=1,norbE
za(jvirt)=zaa(jvirt,ilower)
zb(jvirt)=zbb(jvirt,ilower)
end do

c Lamda:azimuthial quantum number,left oedm
c Mi:azimuthial quantum number,right oedm
c l:left oedm's index,ang. mom. proj.
c m:right oedm's index,ang. mom. proj.

c zl:effective left,zm:effective right

c nl:sum's top limit for the left oedm,angular function
c nm:sum's top limit for the right oedm,angular function

c nkl or kl:sum's top limit for the left oedm,radial function
c nkm or km:sum's top limit for the right oedm,radial function

ovlp=hete_ovlp(iul,iur)
return
print*,'hete_ovlp= ',ovlp
end
```



```

function hete_ovlp(lorb,morb)
implicit real*8 (a-h,o-z)
parameter (norbp=10,aa0=0.d0)
common/cRzpLm0/R, z1,zm, p1,pm
common/cszdfgnfngG/zs(norbp),zd(norbp),Rjump(norbp),p(norbp),
1 f(0:50,norbp),g(0:50,norbp),nf(norbp),ng(norbp),
2 N(norbp),L(norbp),M(norbp)
common/cszdfgnfngE/zsE(norbp),zdE(norbp),RjumpE(norbp),pE(norbp),
1 fE(0:50,norbp),gE(0:50,norbp),nfE(norbp),ngE(norbp),
2 NE(norbp),LE(norbp),ME(norbp)
real*8 fl(0:50),fm(0:50)
real*8 gl(0:50),gm(0:50)
common/cfn/fl,fm, nl,nm
common/cgk/gl,gm, nnl,nnm
common/clm/ml,mm
common/cnorb/lorbb,morbb
common/cmsgnss/msgnll(norbp),msgnrr(norbp)
character c6*200
write (c6,*)lorb,morb !,
read (c6,*)lorbb,morbb !,Rjump1,Rjumpm,icen1,icenm

do i=0,50
gl(i)=aa0
end do

do i=0,50
gm(i)=aa0
end do

do i=0,50

```

```
fl(i)=aa0
end do

do i=0,50
fm(i)=aa0
end do

zl=zsE(lorb)
zm=zs(morb)

      pl=pE(lorb)
      pm=p(morb)

nnl=ngE(lorb)
nnm=ng(morb)

nl=nfE(lorb)
nm=nf(morb)

ml=ME(lorb)*msgnll(lorb)
mm=M(morb)*msgnrr(morb)

do i=0,nnl
gl(i)=gE(i,lorb)
end do

do i=0,nnm
gm(i)=g(i,morb)
```

```
end do

do i=0,nl
fl(i)=fE(i,lorb)
end do

do i=0,nm
fm(i)=f(i,morb)
end do

hete_ovlp= hete_ovlp_delta(R)
print*, ' hete_ovlp= ', hete_ovlp
end

function hete_ovlp_delta(R)
implicit real*8(a-h,o-z)
parameter (pi=3.141592653589793238462d0)
parameter (aa0=0.d0,aa4=4.d0)
common/clm/l,m

if (l.ne.m) then
hete_ovlp_delta=aa0
else
call O_All_prod(product_sum)
hete_ovlp_delta=((pi*R**3)/aa4)*product_sum
end if
end

subroutine O_All_prod(product_sum)
implicit real*8 (a-h,o-z)
```

```
parameter (aa0=0.d0,norbp=10)
common/cxi/xii,xif
common/ceta/etai,etaf
common/cRzpLm0/R, z1,zm, p1,pm
common/czazb/za(norbp),zb(norbp)
common/czazbE/zaE(norbp),zbE(norbp)
common/cnorb/norb1,norb2
common/cicenG0/icen_0(norbp),lstart_0(norbp)
common/cicenE/icenE(norbp),lstartE(norbp)
common/clm/ml,mm

Ang_1=aa0
Ang_2=aa0

if (zaE(norb1).eq.zbE(norb1).and.za(norb2).eq.zb(norb2)) then
Klm0=lstartE(norb1)+lstart_0(norb2)+ml+mm
if(mod(Klm0,2).eq.0) then
Ang_1=Ang_Hol(etai,etaf)
Ang_2=Ang_eta_sq_Hol(etai,etaf)
else
Ang_1=aa0
Ang_2=aa0
end if

else
Ang_1=Ang_Hol(etai,etaf)
Ang_2=Ang_eta_sq_Hol(etai,etaf)
end if

Rad_1=aa0
Rad_2=aa0
```

```

if (Ang_1.ne.aa0) Rad_1=O_Rad_xi_sq_Hol(xii,xif)
if (Ang_2.ne.aa0) Rad_2=O_Rad_Hol(xii,xif)

product_sum=Ang_1*Rad_1 - Ang_2*Rad_2

print*, 'O_All_prod= ', product_sum
end

```

Δ.1.1 Γωνιακά Ολοκληρώματα

```

function Ang_Hol(a,b)
implicit real*8 (a-h,o-z)
external O_sum_prod
parameter (aa1=1.d0,aa10=10.d0)
common/ck/k,mo
do k=14,14
ep=aa10**(-k)
if (a.eq.aa1.or.b.eq.aa1) then
iQDAGS=1
else
iQDAGS=0
end if

Ang_Hol=
1 qromb_DQDAGS_2(iQDAGS,1,a,b,EP,mo,2,O_sum_prod)
      print*, 'Ang_Hol,k,mo= ',Ang_Hol,k,mo
end do
end

function Ang_eta_sq_Hol(a,b)

```

```
implicit real*8 (a-h,o-z)
external O_sum_prod_eta_sq
parameter (aa1=1.d0,aa10=10.d0)
common/ck/k,mo
do k=14,14
ep=aa10**(-k)
if (a.eq.aa1.or.b.eq.aa1) then
iQDAGS=1
else
iQDAGS=0
end if

Ang_eta_sq_Hol=
1 qromb_DQDAGS_2(iQDAGS,1,a,b,EP,mo,2,O_sum_prod_eta_sq)
print*, 'Ang_eta_sq_Hol,k,mo= ',Ang_eta_sq_Hol,k,mo
end do
end

function O_sum_prod_eta_sq(eta)
implicit real*8 (a-h,o-z)
parameter (aa0=0.d0)
if (eta.eq.aa0) then
O_sum_prod_eta_sq=aa0
return
else
O_sum_prod_eta_sq=eta**2*O_sum_prod(eta)
end if
return
end

function O_sum_prod(eta)
```

```
implicit real*8 (a-h,o-z)
real*8 f1(0:50),f2(0:50)
parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0)
parameter (norbp=10)
common/cRzpLm0/R, z1,z2, p1,p2
common/czszdfgnfngG/zs(norbp),zd(norbp),Rjump(norbp),p(norbp),
1 f(0:50,norbp),g(0:50,norbp),nf(norbp),ng(norbp),
2 N(norbp),L(norbp),M(norbp)
common/czszdfgnfngE/zsE(norbp),zdE(norbp),RjumpE(norbp),pE(norbp),
1 fE(0:50,norbp),gE(0:50,norbp),nfE(norbp),ngE(norbp),
2 NE(norbp),LE(norbp),ME(norbp)
common/czazb/za(norbp),zb(norbp)
common/czazbE/zaE(norbp),zbE(norbp)
common/cnorb/norb1,norb2
common/cicenG0/icen_0(norbp),lstart_0(norbp)
common/cicenE/icenE(norbp),lstartE(norbp)
common/clm/m1,m2
common/cfn/f1,f2, n1,n2

abet=abs(eta)

icen1=icenE(norb1)
icen2=icen_0(norb2)

lstart1=lstartE(norb1)
lstart2=lstart_0(norb2)

Rjump1=RjumpE(norb1)
Rjump2=Rjump(norb2)

za1=zaE(norb1)
```

```
zb1=zbE (norb1)
za2=za (norb2)
zb2=zb (norb2)

m1=ME (norb1)
m2=M (norb2)

p1=pE (norb1)
p2=p (norb2)

c for icen=1 choose Exp(+p1*eta)*ALegP(m1+i1,m1,+eta)
c for icen=2 choose Exp(-p1*eta)*ALegP(m1+i1,m1,-eta)

inv1=1
if(za1.lt.zb1) inv1=-1
inv2=1
if(za2.lt.zb2) inv2=-1

isgn1=1-2*(icen1-1)
isgn2=1-2*(icen2-1)

if (za1.eq.zb1) then
sum1=aa0
do i1=0,n1
GP=ALegP(2*i1+lstart1,m1,eta)
sum1=sum1+f1(i1)*GP
end do
if (sum1.eq.aa0) then
hete_2e_ang=aa0
```



```

return
end if
else if (za1.ne.zb1) then
if (R.lt.Rjump1) then
sum1=aa0
do i1=0,n1
  GP=Exp(isgn1*p1*eta)*ALegP(2*i1+lstart1,m1,isgn1*eta)
  sum1=sum1+f1(i1)*GP
end do
if (sum1.eq.aa0) then
O_sum_prod=aa0
return
end if
else if (R.gt.Rjump1) then

if (abet.eq.aa1.and.m1.ne.0) then
O_sum_prod=aa0
return
end if
c for icen=1 choose (aa1-eta**2)**(m1/aa2)*Exp(-p1*eta)*sum[(aa1+eta)
c for icen=2 choose (aa1-eta**2)**(m1/aa2)*Exp(+p1*eta)*sum[(aa1-eta)
sum1=aa0
do i1=0,n1
  sum1=sum1+f1(i1)*(aa1+isgn1*eta)**(2*i1+lstart1) (aa1+isgn1*eta)**
end do
sum1=sum1*(aa1-eta**2)**(m1/aa2)*Exp(-isgn1*p1*eta)
end if
end if

c for icen=1 choose Exp(+p2*eta)*ALegP(m2+i2,m2,+eta)
c for icen=2 choose Exp(-p2*eta)*ALegP(m2+i2,m2,-eta)

```

```

if (za2.eq.zb2) then
sum2=aa0
do i2=0,n2
GP=ALegP(2*i2+lstart2,m2,eta)
  sum2=sum2+f2(i2)*GP
end do
if (sum2.eq.aa0) then
hete_2e_ang=aa0
return
end if
else if (za2.ne.zb2) then
if (R.lt.Rjump2) then
sum2=aa0
do i2=0,n2
GP=Exp(isgn2*p2*eta)*ALegP(2*i2+lstart2,m2,isgn2*eta)
  sum2=sum2+f2(i2)*GP
end do
if (sum2.eq.aa0) then
O_sum_prod=aa0
return
end if
else if (R.gt.Rjump2) then

if (abet.eq.aa1.and.m2.ne.0) then
O_sum_prod=aa0
return
end if
c for icen=1 choose (aa1-eta**2)**(m2/aa2)*Exp(-p2*eta)*sum[(aa1+eta)
c for icen=2 choose (aa1-eta**2)**(m2/aa2)*Exp(+p2*eta)*sum[(aa1-eta)
sum2=aa0
do i2=0,n2

```

```

    sum2=sum2+f2(i2)*(aa1+isgn2*eta)**(2*i2+lstart2)
end do
sum2=sum2*(aa1-eta**2)**(m2/aa2)*Exp(-isgn2*p2*eta)
end if
end if

O_sum_prod=sum1*sum2
return
end

```

Δ.1.2 Ακτινικά Ολοκληρώματα

```

function O_Rad_xi_sq_Hol(a,b)
implicit real*8 (a-h,o-z)
external O_Rad_xi_sq
parameter (aa10=10.d0)
common/ck/k,mo
do k=14,14
ep=aa10**(-k)
O_Rad_xi_sq_Hol=
1 qromb_DQDAGS_1(0,0,a,b,EP,mo,2,O_Rad_xi_sq)
print*, 'O_Rad_xi_sq_Hol,k,mo= ',O_Rad_xi_sq_Hol,k,mo
end do
end

function O_Rad_xi_sq(xi)
implicit real*8 (a-h,o-z)
parameter (aa0=0.d0)
if (xi.eq.aa0) then
O_Rad_xi_sq=aa0

```

```
return
else
O_Rad_xi_sq=xi**2*O_Rad(xi)
end if
return
end
```

```
function O_Rad_Hol(a,b)
implicit real*8 (a-h,o-z)
external O_Rad
parameter (aa10=10.d0)
common/ck/k,mo
do k=14,14
ep=aa10**(-k)
O_Rad_Hol=
1 qromb_DQDAGS_1(0,0,a,b,EP,mo,2,O_Rad)
print*, 'O_Rad_Hol,k,mo= ',O_Rad_Hol,k,mo
end do
end
```

```
function O_Rad(xi)
implicit real*8 (a-h,o-z)
real*8 gl(0:50),gm(0:50)
common/cRzpLm0/R, z1,zm, pl,pm
common/cgk/gl,gm, kl,km
common/clm/l,m
parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0)
sumlR=aa0
do il=0,kl
sumlR=sumlR+gl(il)*((xi-aa1)/(xi+aa1))**il
```

```

end do
if (sumlR.eq.aa0) then
O_Rad=aa0
return
end if
summR=aa0
do im=0,km
summR=summR+gm(im)*((xi-aa1)/(xi+aa1))**im
end do
if (summR.eq.aa0) then
O_Rad=aa0
return
end if
O_Rad=Exp(-(pl*xi) - pm*xi)*(-aa1 + xi)**(l/aa2 + m/aa2)*
- sumlR*summR*(aa1 + xi)**
- (-aa2 - l/aa2 - m/aa2 + (R*z1)/(aa2*pl) + (R*zm)/(aa2*pm))
end

```

Δ.2 Μονολεκτρονιακά

```

function e_1(iul,iur)
implicit real*8 (a-h,o-z)
parameter (norbp=10,nlowerp=10,aa0=0.d0)
common/cRzpLm1/R, pl,pm, zzs,zzd, zl,zm, zzl,zzm
common/czszdfgnfngG1/zs_1(norbp),zd_1(norbp),Rjump_1(norbp),
1p_1(norbp),f_1(0:50,norbp),g_1(0:50,norbp),nf_1(norbp),ng_1(norbp),
1 N_1(norbp),L_1(norbp),M_1(norbp)
common/czszdfgnfngE1/zs_1E(norbp),zd_1E(norbp),Rjump_1E(norbp),
1p_1E(norbp),f_1E(0:50,norbp),g_1E(0:50,norbp),nf_1E(norbp),
1 ng_1E(norbp),N_1E(norbp),L_1E(norbp),M_1E(norbp)

```

```
1 , jvelec
common/czazbG/zaa (norbp, 0:nlowerp) , zbb (norbp, 0:nlowerp)
common/czazb/za (norbp) , zb (norbp)
common/czazbE/zaE (norbp) , zbE (norbp)
common/cxi/xii, xif
common/ceta/etai, etaf
common/cilower/ilower
common/cnorbE/norbE, nlower

do jvirt=1, norbE
za (jvirt)=zaa (jvirt, ilower)
zb (jvirt)=zbb (jvirt, ilower)
end do

c Lamda:azimuthial quantum number, left oedm
c Mi:azimuthial quantum number, right oedm

c l:left oedm's index, ang. mom. proj.
c m:right oedm's index, ang. mom. proj.

c zzs=Za+Zb:sum of nuclear charges

c zs=Za*+Zb*:sum of effective nuclear charges (left or right oedm)

c zzd=Za-Zb:dif. of nuclear charges

c zd=Za*-Zb*:dif. of effective nuclear charges (left or right oedm)
```

```

c nl:sum's top limit for the left oedm,angular function
c nm:sum's top limit for the right oedm,angular function

c nkl or kl:sum's top limit for the left oedm,radial function
c nkm or km:sum's top limit for the right oedm,radial function

e_1= hete_1e(iul,iur)
return
print*,' hete_1e= ', e_1
end

function hete_1e(lorb,morb)
implicit real*8 (a-h,o-z)
parameter (norbp=10,aa0=0.d0)
common/cRzpLm1/R, pl,pm, zzs,zzd, zl,zm, zzl,zzm
common/czszdfgnfngG1/zs_1(norbp),zd_1(norbp),Rjump_1(norbp),
1p_1(norbp),f_1(0:50,norbp),g_1(0:50,norbp),nf_1(norbp),ng_1(norbp),
1 N_1(norbp),L_1(norbp),M_1(norbp)
common/czszdfgnfngE1/zs_1E(norbp),zd_1E(norbp),Rjump_1E(norbp),
1p_1E(norbp),f_1E(0:50,norbp),g_1E(0:50,norbp),nf_1E(norbp),
1 ng_1E(norbp),N_1E(norbp),L_1E(norbp),M_1E(norbp)
real*8 fl(0:50),fm(0:50)
real*8 gl(0:50),gm(0:50)
common/cfn1/fl,fm, nl,nm
common/cgk1/gl,gm, nnl,nnm
common/clm1/ml,mm
common/cnorb/lorbb,morbb

```

```
common/cmsgnss/msgnll(norbp),msgnrr(norbp)
character c5*200
write (c5,*)lorb,morb !,
read (c5,*)lorbb,morbb !,Rjump1,Rjumpm,icen1,icenm

do i=0,50
gl(i)=aa0
end do

do i=0,50
gm(i)=aa0
end do

do i=0,50
fl(i)=aa0
end do

do i=0,50
fm(i)=aa0
end do

zl=zs_1E(lorb)
zm=zs_1(morb)

zzl=zd_1E(lorb)
zzm=zd_1(morb)

pl=p_1E(lorb)
```



```
pm=p_1(morb)

nnl=ng_1E(lorb)
nnm=ng_1(morb)

nl=nf_1E(lorb)
nm=nf_1(morb)

ml=M_1E(lorb)*msgnll(lorb) !left m:angular momentum projection eigen
mm=M_1(morb)*msgnrr(morb) !right m: ...

do i=0,nnl
gl(i)=g_1E(i,lorb)
end do

do i=0,nnm
gm(i)=g_1(i,morb)
end do

do i=0,nl
fl(i)=f_1E(i,lorb)
end do

do i=0,nm
fm(i)=f_1(i,morb)
end do

hete_1e= hete_1e_delta(R)
print*, ' hete_1e= ', hete_1e
```

end

```
function hete_1e_delta(R)
implicit real*8(a-h,o-z)
parameter (pi=3.141592653589793238462d0)
parameter (aa0=0.d0,aa2=2.d0)
common/clm1/l,m

if (l.ne.m) then
hete_1e_delta=aa0
else
    call All_prod(product_sum)
hete_1e_delta=- (pi*R)/aa2)*product_sum
end if
end
```

```
subroutine All_prod(product_sum)
implicit real*8 (a-h,o-z)
parameter (aa0=0.d0,norbp=10)
common/cxi/xii,xif
common/ceta/etai,etaf
common/cRzpLm1/R, pl,pm, zzs,zzd, zl,zm, zzl,zzm
common/czazb/za(norbp),zb(norbp)
common/czazbE/zaE(norbp),zbE(norbp)
common/cnorb/norb1,norb2
common/cicenG1/icen_1(norbp),lstart_1(norbp)
common/cicenE/icenE(norbp),lstartE(norbp)
common/clm1/ml,mm
```

```
Ang_1=aa0
Ang_2=aa0
Ang_3=aa0
Ang_4=aa0

if (zaE(norb1).eq.zbE(norb1).and.za(norb2).eq.zb(norb2)) then
Klm1=lstartE(norb1)+lstart_1(norb2)+ml+mm
if(mod(Klm1,2).eq.0) then
Ang_3=Ang(etai,etaf)
Ang_1=pm**2*Ang_3
Ang_2=Ang_eta_sq(etai,etaf)
else
Ang_1=aa0
Ang_2=aa0
Ang_3=aa0
end if

else
Ang_3=Ang(etai,etaf)
Ang_1=pm**2*Ang_3
Ang_2=Ang_eta_sq(etai,etaf)
end if

if (zaE(norb1).eq.zbE(norb1).and.za(norb2).eq.zb(norb2)) then
Klm1eta=lstartE(norb1)+lstart_1(norb2)+ml+mm+1
if(mod(Klm1eta,2).eq.0) then
Ang_4=R*(zzd-zzm)*Ang_eta(etai,etaf)
else
Ang_4=aa0
end if
else
```

```

Ang_4=R*(zzd-zzm)*Ang_eta(etai,etaf)
end if

Rad_1=aa0
Rad_2=aa0
Rad_3=aa0
Rad_4=aa0

if(Ang_1.ne.aa0) Rad_1=Rad_xi_sq_Hol(xii,xif)
if(Ang_4.ne.aa0) Rad_4=Rad_Hol(xii,xif)
if(Ang_2.ne.aa0.and.Ang_4.ne.aa0)then
Rad_2=pm**2*Rad_4
else
if(Ang_2.ne.aa0) Rad_2=pm**2*Rad_Hol(xii,xif)
end if
if(Ang_3.ne.aa0) Rad_3=R*(zzs-zm)*Rad_xi_Hol(xii,xif)

product_sum=Ang_1*Rad_1 - Ang_2*Rad_2 + Ang_3*Rad_3 - Ang_4*Rad_4

print*, 'Ang_1=', Ang_1, 'Ang_2=', Ang_2, 'Ang_3=', Ang_3, 'Ang_4=', Ang_4,
1 'Rad_1=', Rad_1, 'Rad_2=', Rad_2, 'Rad_3=', Rad_3, 'Rad_4=', Rad_4
print*, 'All_prod= ', product_sum
end

```

Δ.2.1 Γωνιακά Ολοκληρώματα

```

function Ang_eta_sq(a,b)
implicit real*8 (a-h,o-z)
external sum_prod_eta_sq
parameter (aa1=1.d0,aa10=10.d0)

```

```
common/ck/k,mo
do k=14,14
ep=aa10**(-k)
if (a.eq.aa1.or.b.eq.aa1) then
iQDAGS=1
else
iQDAGS=0
end if
Ang_eta_sq=
1 qromb_DQDAGS_2(iQDAGS,1,a,b,EP,mo,2,sum_prod_eta_sq)
print*, 'Ang_eta_sq= ',Ang_eta_sq,k,mo
end do
end
```

```
function sum_prod_eta_sq(eta)
implicit real*8 (a-h,o-z)
parameter (aa0=0.d0)
if (eta.eq.aa0) then
sum_prod_eta_sq=aa0
return
else
sum_prod_eta_sq=eta**2*sum_prod(eta)
end if
return
end
```

```
function Ang_eta(a,b)
implicit real*8 (a-h,o-z)
external sum_prod_eta
parameter (aa1=1.d0,aa10=10.d0)
common/ck/k,mo
```

```
do k=14,14
ep=aa10**(-k)
if (a.eq.aa1.or.b.eq.aa1) then
iQDAGS=1
else
iQDAGS=0
end if
Ang_eta=
1 qromb_DQDAGS_2(iQDAGS,1,a,b,EP,mo,2,sum_prod_eta)
print*, 'Ang_eta= ',Ang_eta,k,mo
end do
end
```

```
function sum_prod_eta(eta)
implicit real*8 (a-h,o-z)
parameter (aa0=0.d0)
if (eta.eq.aa0) then
sum_prod_eta=aa0
return
else
sum_prod_eta=eta*sum_prod(eta)
end if
return
end
```

```
function Ang(a,b)
implicit real*8 (a-h,o-z)
external sum_prod
parameter (aa1=1.d0,aa10=10.d0)
common/ck/k,mo
```

```

do k=14,14
ep=aa10**(-k)
if (a.eq.aa1.or.b.eq.aa1) then
iQDAGS=1
else
iQDAGS=0
end if

```

```

Ang=
1 qromb_DQDAGS_2(iQDAGS,1,a,b,EP,mo,2,sum_prod)
print*, 'Ang,k,mo= ',Ang,k,mo
end do
end

```

```

function sum_prod(eta)
implicit real*8 (a-h,o-z)
real*8 f1(0:50),f2(0:50)
parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0)
parameter (norbp=10)
common/cRzpLm1/R, p1,p2, zzs,zzd, z1,z2, zz1,zz2
common/czszdfgnfngG1/zs_1(norbp),zd_1(norbp),Rjump_1(norbp),
1p_1(norbp),f_1(0:50,norbp),g_1(0:50,norbp),nf_1(norbp),ng_1(norbp),
1 N_1(norbp),L_1(norbp),M_1(norbp)
common/czszdfgnfngE1/zs_1E(norbp),zd_1E(norbp),Rjump_1E(norbp),
1p_1E(norbp),f_1E(0:50,norbp),g_1E(0:50,norbp),nf_1E(norbp),
1 ng_1E(norbp),N_1E(norbp),L_1E(norbp),M_1E(norbp)
common/czazb/za(norbp),zb(norbp)
common/czazbE/zaE(norbp),zbE(norbp)
common/cnorb/norb1,norb2
common/cicenG1/icen_1(norbp),lstart_1(norbp)
common/cicenE/icenE(norbp),lstartE(norbp)

```

```
common/clm1/m1,m2
common/cfn1/f1,f2, n1,n2

abet=abs(eta)

icen1=icenE(norb1)
icen2=icen_1(norb2)

lstart1=lstartE(norb1)
lstart2=lstart_1(norb2)

Rjump1=Rjump_1E(norb1)
Rjump2=Rjump_1(norb2)

za1=zaE(norb1)
zb1=zbE(norb1)
za2=za(norb2)
zb2=zb(norb2)

m1=M_1E(norb1)
m2=M_1(norb2)

p1=p_1E(norb1)
p2=p_1(norb2)

c for icen=1 choose Exp(+p1*eta)*ALegP(m1+i1,m1,+eta)
c for icen=2 choose Exp(-p1*eta)*ALegP(m1+i1,m1,-eta)
```



```

inv1=1
if(za1.lt.zb1) inv1=-1
inv2=1
if(za2.lt.zb2) inv2=-1

isgn1=(1-2*(icen1-1))*inv1
isgn2=(1-2*(icen2-1))*inv2

if (za1.eq.zb1) then
sum1=aa0
do i1=0,n1
GP=ALegP(2*i1+lstart1,m1,eta)
  sum1=sum1+f1(i1)*GP
end do
if (sum1.eq.aa0) then
hete_2e_ang=aa0
return
end if
else if (za1.ne.zb1) then
if (R.lt.Rjump1) then
sum1=aa0
do i1=0,n1
c for the coefficients fj start from j=0 in the order given by the oe
c and for P(l,m) take correctly l=j+m
  GP=Exp(isgn1*p1*eta)*ALegP(2*i1+lstart1,m1,isgn1*eta)
  sum1=sum1+f1(i1)*GP
end do
if (sum1.eq.aa0) then
sum_prod=aa0
return
end if

```

```

else if (R.gt.Rjump1) then

if (abet.eq.aa1.and.m1.ne.0) then
sum_prod=aa0
return
end if

c for icen=1 choose (aa1-eta**2)**(m1/aa2)*Exp(-p1*eta)*sum[(aa1+eta)
c for icen=2 choose (aa1-eta**2)**(m1/aa2)*Exp(+p1*eta)*sum[(aa1-eta)
sum1=aa0
do i1=0,n1
    sum1=sum1+f1(i1)*(aa1+isgn1*eta)**(2*i1+lstart1) (aa1+isgn1*eta)**
end do
sum1=sum1*(aa1-eta**2)**(m1/aa2)*Exp(-isgn1*p1*eta)
end if
end if

c for icen=1 choose Exp(+p2*eta)*ALegP(m2+i2,m2,+eta)
c for icen=2 choose Exp(-p2*eta)*ALegP(m2+i2,m2,-eta)
if (za2.eq.zb2) then
sum2=aa0
do i2=0,n2
GP=ALegP(2*i2+lstart2,m2,eta)
    sum2=sum2+f2(i2)*GP
end do
if (sum2.eq.aa0) then
hete_2e_ang=aa0
return
end if
else if (za2.ne.zb2) then
if (R.lt.Rjump2) then
sum2=aa0
do i2=0,n2

```

```

GP=Exp(isgn2*p2*eta)*ALegP(2*i2+lstart2,m2,isgn2*eta)
  sum2=sum2+f2(i2)*GP
end do
if (sum2.eq.aa0) then
sum_prod=aa0
return
end if
else if (R.gt.Rjump2) then

if (abet.eq.aa1.and.m2.ne.0) then
sum_prod=aa0
return
end if
c for icen=1 choose (aa1-eta**2)**(m2/aa2)*Exp(-p2*eta)*sum[(aa1+eta)
c for icen=2 choose (aa1-eta**2)**(m2/aa2)*Exp(+p2*eta)*sum[(aa1-eta)
sum2=aa0
do i2=0,n2
  sum2=sum2+f2(i2)*(aa1+isgn2*eta)**(2*i2+lstart2) (aa1+isgn2*eta)**
end do
sum2=sum2*(aa1-eta**2)**(m2/aa2)*Exp(-isgn2*p2*eta)
end if
end if

sum_prod=sum1*sum2
return
end

```

Δ.2.2 Ακτινικά Ολοκληρώματα

```
function Rad_xi_sq_Hol(a,b)
```

```
implicit real*8 (a-h,o-z)
external Rad_xi_sq
parameter (aa10=10.d0)
common/ck/k,mo
do k=14,14
ep=aa10**(-k)
Rad_xi_sq_Hol=
1 qromb_DQDAGS_1(1,0,a,b,EP,mo,2,Rad_xi_sq)
print*, 'Rad_xi_sq_Hol,k,mo= ',Rad_xi_sq_Hol,k,mo
end do
end
```

```
function Rad_xi_sq(xi)
implicit real*8 (a-h,o-z)
parameter (aa0=0.d0,aa1=1.d0)
if (xi.eq.aa0.or.xi.eq.aa1) then
Rad_xi_sq=aa0
return
else
Rad_xi_sq=xi**2*Rad(xi)
end if
return
end
```

```
function Rad_xi_Hol(a,b)
implicit real*8 (a-h,o-z)
external Rad_xi
parameter (aa10=10.d0)
common/ck/k,mo
do k=14,14
ep=aa10**(-k)
```

```
Rad_xi_Hol=  
1 qromb_DQDAGS_1(1,0,a,b,EP,mo,2,Rad_xi)  
print*, 'Rad_xi_Hol,k,mo= ',Rad_xi_Hol,k,mo  
end do  
end
```

```
function Rad_xi(xi)  
implicit real*8 (a-h,o-z)  
parameter (aa0=0.d0,aa1=1.d0)  
if (xi.eq.aa0.or.xi.eq.aa1) then  
Rad_xi=aa0  
return  
else  
Rad_xi=xi*Rad(xi)  
end if  
return  
end
```

```
function Rad_Hol(a,b)  
implicit real*8 (a-h,o-z)  
external Rad  
parameter (aa10=10.d0)  
common/ck/k,mo  
do k=14,14  
ep=aa10**(-k)  
Rad_Hol=  
1 qromb_DQDAGS_1(1,0,a,b,EP,mo,2,Rad)  
print*, 'Rad_Hol,k,mo= ',Rad_Hol,k,mo  
end do  
end
```

```

function Rad(xi)
implicit real*8 (a-h,o-z)
real*8 gl(0:50),gm(0:50)
common/cRzpLm1/R, pl,pm, zzs,zzd, zl,zm, zzl,zzm
common/cgk1/gl,gm, kl,km
common/clm1/l,m
parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0)
sumlR=aa0
do il=0,kl
sumlR=sumlR+gl(il)*((xi-aa1)/(xi+aa1))**il
end do
if (sumlR.eq.aa0) then
Rad=aa0
return
end if
summR=aa0
do im=0,km
summR=summR+gm(im)*((xi-aa1)/(xi+aa1))**im
end do
if (summR.eq.aa0) then
Rad=aa0
return
end if
Rad= Exp(-(pl*xi) - pm*xi)*(-aa1 + xi)**(l/aa2 + m/aa2)*
- (aa1 + xi)**
- (-aa2 - l/aa2 - m/aa2 + (R*zl)/(aa2*pl) + (R*zm)/(aa2*pm))*
- sumlR*summR
end

```

Δ.3 Διηλεκτρονικά

```

function e_2(iulI,iulJ,iurK,iurL)
implicit real*8 (a-h,o-z)
parameter (norbp=10,nlowerp=10,aa0=0.d0)
character fG05Lm1 *6400,fG05L05m1 *6400,fGL15 *3200
common/czszdfgnfngG2/zs_2(norbp),zd_2(norbp),Rjump_2(norbp),
1p_2(norbp),f_2(0:50,norbp),g_2(0:50,norbp),nf_2(norbp),ng_2(norbp),
1 N_2(norbp),L_2(norbp),M_2(norbp)
common/czszdfgnfngE2/zs_2E(norbp),zd_2E(norbp),Rjump_2E(norbp),
1p_2E(norbp),f_2E(0:50,norbp),g_2E(0:50,norbp),nf_2E(norbp),
1 ng_2E(norbp),N_2E(norbp),L_2E(norbp),M_2E(norbp)

1 ,jvelec
common/czazbG/zaa(norbp,0:nlowerp),zbb(norbp,0:nlowerp)
common/czazb/za(norbp),zb(norbp)
common/czazbE/zaE(norbp),zbE(norbp)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, Lamda,mu

common/cGamma/Gamma05Lm1(-15:15,0:15),Gamma05L05m1(-15:15,0:15),
1 GammaL15(0:15)
character*3 ffile
common/cilower/ilower
common/cnorbE/norbE,nlower

do jvirt=1,norbE
za(jvirt)=zaa(jvirt,ilower)
zb(jvirt)=zbb(jvirt,ilower)
end do

```

```

open (75, file=
1 'G:\Hector\Functon Library\Gamma_Hpg.txt')
read (75, *) ((iL, imu, Gamma05Lm1(imu, iL), Gamma05L05m1(imu, iL),
1 GammaL15(iL), iimu=-iL, iL), iL=0, 15)

e_2=hete_2e(iulI, iulJ, iurK, iurL)

close (75)
return
print*, 'hete_2e= ', e_2
end

function hete_2e(iorb, jorb, korb, lorb)
implicit real*8 (a-h, o-z)
parameter (norbp=10)
common/czszdfgnfngG2/zs_2(norbp), zd_2(norbp), Rjump_2(norbp),
1p_2(norbp), f_2(0:50, norbp), g_2(0:50, norbp), nf_2(norbp), ng_2(norbp),
1 N_2(norbp), L_2(norbp), M_2(norbp)
common/czszdfgnfngE2/zs_2E(norbp), zd_2E(norbp), Rjump_2E(norbp),
1p_2E(norbp), f_2E(0:50, norbp), g_2E(0:50, norbp), nf_2E(norbp),
1 ng_2E(norbp), N_2E(norbp), L_2E(norbp), M_2E(norbp)
real*8 fi(0:50), fk(0:50), fj(0:50), fl(0:50)
real*8 gi(0:50), gk(0:50), gj(0:50), gl(0:50)
common/cRzpLm/R, zi, zk, zj, zl, pi, pk, pj, pl, Lamda, mu
common/cFhete_2e_Rad_in/mj, ml, nj, nl, gj, gl
common/cFhete_2e_Rad_out/mi, mk, ni, nk, gi, gk
common/chete_2e_Ang_ik/nni, nnk, fi, fk, iorbb, korbb
common/chete_2e_Ang_jl/nnj, nnl, fj, fl, jorbb, lorbb
common /cxii/xii

```



```
common /cxif/xif
common/cetai/etai
common/cetaf/etaf
common/cLamda/maLamda
common/cmsgnss/msgnll(norbp),msgnrr(norbp)

parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0,aa3=3.d0,bb14=1.4d0)
parameter(n0=0,nFac=29)
common /cfac/fac(n0:nfac)
character ffac*504,c3*200
write (c3,*)iorb,korb,jorb,lorb
read (c3,*)iorbb,korbb,jorbb,lorbb

do i=0,50
gi(i)=aa0
end do

do i=0,50
gk(i)=aa0
end do

do i=0,50
gj(i)=aa0
end do

do i=0,50
gl(i)=aa0
end do

do i=0,50
fi(i)=aa0
```

```
end do

do i=0,50
fk(i)=aa0
end do

do i=0,50
fj(i)=aa0
end do

do i=0,50
fl(i)=aa0
end do

xii=1.d0
xif=100.d0

etai=-1.d0
etaf=1.d0

zi=zs_2E(iorb)
zk=zs_2(korb)
zj=zs_2E(jorb)
zl=zs_2(lorb)

pi=p_2E(iorb)
pk=p_2(korb)
pj=p_2E(jorb)
pl=p_2(lorb)
```

```
ni=ng_2E(iorb)
nk=ng_2(korb)
nj=ng_2E(jorb)
nl=ng_2(lorb)

nni=nf_2E(iorb)
nnk=nf_2(korb)
nnj=nf_2E(jorb)
nnl=nf_2(lorb)

mi=M_2E(iorb)*msgnll(iorb)
mk=M_2(korb)*msgnrr(korb)
mj=M_2E(jorb)*msgnll(jorb)
ml=M_2(lorb)*msgnrr(lorb)

do i=0,ni
gi(i)=g_2E(i,iorb)
end do

do i=0,nk
gk(i)=g_2(i,korb)
end do

do i=0,nj
gj(i)=g_2E(i,jorb)
end do

do i=0,nl
gl(i)=g_2(i,lorb)
end do
```

```
do i=0,nni
fi(i)=f_2E(i,iorb)
end do
```

```
do i=0,nnk
fk(i)=f_2(i,korb)
end do
```

```
do i=0,nnj
fj(i)=f_2E(i,jorb)
end do
```

```
do i=0,nnl
fl(i)=f_2(i,lorb)
end do
```

```
hete_2e= hete_2e_DS(R,maLamda)
print*, ' hete_2e= ', hete_2e
end
```

```
function hete_2e_DS(R,maL)
implicit real*8 (a-h,o-z)
parameter (aa0=0.d0,aa2=2.d0,aa8=8.d0,pi=3.141592653589793238462d0)
parameter (n0=0,nFac=29)
common /cfac/fac(n0:nfac)
common/cFhete_2e_Rad_in/mj,ml, nj,nl, gj,gl
common/cFhete_2e_Rad_out/mi,mk, ni,nk, gi,gk
sumL=aa0
do iL=0,maL
```

```
    summu=aa0
    do imu=0,iL
print*, 'iL,imu=' ,iL,imu
id=Kdelta(mi,mk+imu)*Kdelta(ml,mj+imu)
1 +Kdelta(mi,mk-imu)*Kdelta(ml,mj-imu)
if(id.eq.0)then

else

fun=All_Hol(iL,imu)
summu=summu+(-1)**imu*ie(imu)*fun*id*
1 (fac(iL-imu)/fac(iL+imu))**2
end if
end do
    sumL=sumL+((2*iL+1)/aa2)*summu
print*, 'iL,sumL=', iL, sumL

    continue
end do
hete_2e_DS =((pi**2*R**5)/aa8)*sumL
end

function ie(imu)
ie=2-Kdelta(imu,0)
end

function Kdelta(i,j)
Kdelta=0
if(i.eq.j) Kdelta=1
end
```

```

function All_Hol(L,mu)
implicit real*8 (a-h,o-z)
parameter (aa0=0.d0,norbp=10)
real*8 fi(0:50),fk(0:50),fj(0:50),fl(0:50)
real*8 gi(0:50),gk(0:50),gj(0:50),gl(0:50)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, LL,mmu
common/cFhete_2e_Rad_in/mj,ml, nj,nl, gj,gl
common/cFhete_2e_Rad_out/mi,mk, ni,nk, gi,gk
common/chete_2e_Ang_ik/nni,nnk, fi,fk, iorb,korb
common/chete_2e_Ang_jl/nnj,nnl, fj,fl, jorb,lorb
common/cxi/xii,xif
common/ceta/etai,etaf
c common/cxifinalin/xifinalin
c common/cxifinal/xifinal
common/cicenG2/icen_2(norbp),lstart_2(norbp)
common/cicenE/icenE(norbp),lstartE(norbp)
common/czazb/za(norbp),zb(norbp)
common/czazbE/zaE(norbp),zbE(norbp)

LL=L
mmu=mu

Aik=aa0
Ajl=aa0
Ansqik=aa0
Ansqjl=aa0

if(zaE(iorb).eq.zbE(iorb).and.za(korb).eq.zb(korb)) then
Kik=lstartE(iorb)+lstart_2(korb)+mi+mk+L+mu
if(mod(Kik,2).eq.0) Aik=hete_2e_ang_gen(etai,etaf, R,
1 zi,zk, pi,pk, mi,mk, L,mu,nni,nnk, fi,fk, iorb,korb)

```

```

if(mod(Kik,2).eq.0) Ansqik=hete_2e_ang_eta_sq_gen(etai,etaf, R,
1  zi,zk, pi,pk, mi,mk, L,mu,nni,nnk, fi,fk, iorb,korb)
else
Aik=hete_2e_ang_gen(etai,etaf, R,
1  zi,zk, pi,pk, mi,mk, L,mu,nni,nnk, fi,fk, iorb,korb)
Ansqik=hete_2e_ang_eta_sq_gen(etai,etaf, R,
1  zi,zk, pi,pk, mi,mk, L,mu,nni,nnk, fi,fk, iorb,korb)
end if

if(zaE(jorb).eq.zbE(jorb).and.za(lorb).eq.zb(lorb)) then
Kjl=lstartE(jorb)+lstart_2(lorb)+mj+ml+L+mu
if(mod(Kjl,2).eq.0) Ajl=hete_2e_ang_gen(etai,etaf, R,
1  zj,zl, pj,pl, mj,ml, L,mu,nnj,nnl, fj,fl, jorb,lorb)
if(mod(Kjl,2).eq.0) Ansqjl=hete_2e_ang_eta_sq_gen(etai,etaf, R,
1  zj,zl, pj,pl, mj,ml, L,mu,nnj,nnl, fj,fl, jorb,lorb)
else
Ajl=hete_2e_ang_gen(etai,etaf, R,
1  zj,zl, pj,pl, mj,ml, L,mu,nnj,nnl, fj,fl, jorb,lorb)
Ansqjl=hete_2e_ang_eta_sq_gen(etai,etaf, R,
1  zj,zl, pj,pl, mj,ml, L,mu,nnj,nnl, fj,fl, jorb,lorb)
end if

R1= aa0
R2= aa0
R3= aa0
R4= aa0
R5= aa0
R6= aa0
R7= aa0
R8= aa0

```

```

if (Aik*Ajl.ne.aa0) R1=hete_2e_Rad_1(xii,xif)
if (Aik*Ansqjl.ne.aa0) R2=hete_2e_Rad_2(xii,xif)
if (Aik*Ajl.ne.aa0) R3=hete_2e_Rad_3(xii,xif)
if (Aik*Ansqjl.ne.aa0) R4=hete_2e_Rad_4(xii,xif)
if (Ajl*Ansqik.ne.aa0) R5=hete_2e_Rad_5(xii,xif)
if (Ansqik*Ansqjl.ne.aa0) R6=hete_2e_Rad_6(xii,xif)
if (Ajl*Ansqik.ne.aa0) R7=hete_2e_Rad_7(xii,xif)
if (Ansqik*Ansqjl.ne.aa0) R8=hete_2e_Rad_8(xii,xif)

All_Hol=Aik*((R1+R3)*Ajl-(R2+R4)*Ansqjl)+
1 Ansqik*((R6+R8)*Ansqjl-(R5+R7)*Ajl)

print*, 'All_Hol = ', All_Hol
end

```

Δ.3.1 Ακτινικά Ολοκληρώματα

```

function hete_2e_ang_eta_sq_gen(etai,etaf, R,
1 z1,z2, p1,p2, m1,m2, L,mu, n1,n2, f1,f2, norb1,norb2)
implicit real*8 (a-h,o-z)
common /cRzpLmu/RR, zz1,zz2, pp1,pp2, LL,mmu
common /cFhete_2e_ang_eta_sq_gen/mm1, mm2, nn1, nn2, ff1, ff2,
1 nnorb1, nnorb2
real*8 f1(0:50),f2(0:50),ff1(0:50),ff2(0:50)
parameter (aa1=1.d0,am1=-1.d0)
character c1*225, c2*2800

write (c1,*) R,z1,z2,p1,p2,L,mu
read (c1,*) RR,zz1,zz2,pp1,pp2,LL,mmu
write (c2,*) m1, m2, n1, n2, f1, f2, norb1, norb2
read (c2,*) mm1, mm2, nn1, nn2, ff1, ff2, nnorb1, nnorb2

```



```

hete_2e_ang_eta_sq_gen=hete_2e_ang_eta_sq_gen_Hol(etai,etaf)
print *,'hete_2e_ang_eta_sq_gen= ',hete_2e_ang_eta_sq_gen
end

function het_2e_ang_eta_sq_gen_Hol(a,b)
implicit real*8 (a-h,o-z)
external Fhete_2e_ang_eta_sq_gen
parameter (aa10=10.d0)
common/ck/k,mo
do k=15,15
ep=aa10**(-k)
hete_2e_ang_eta_sq_gen_Hol=
1      qromb_DQDAGS_2(0,1,a,b,EP,mo,2,Fhete_2e_ang_eta_sq_gen)
print *, het_2e_ang_eta_sq_gen_Hol,k,mo
end do
end

function Fhete_2e_ang_eta_sq_gen(eta)
implicit real*8 (a-h,o-z)
real*8 f1(0:50),f2(0:50)
common /cRzpLmu/R, z1,z2, p1,p2, L,mu
common /cFhete_2e_ang_eta_sq_gen/m1, m2, n1, n2, f1, f2,
1 norb1, norb2
Fhete_2e_ang_eta_sq_gen=
1 het_2e_ang_eta_sq(R, z1,z2, p1,p2, eta,
1      m1,m2, L,mu, n1,n2, f1,f2, norb1,norb2)
return
end

function het_2e_ang_eta_sq(R, z1,z2, p1,p2, eta,

```

```

1  m1,m2, L,mu,n1,n2, f1,f2, norb1,norb2)
implicit real*8 (a-h,o-z)
real*8 f1(0:50),f2(0:50)
parameter (aa0=0.d0)
if (eta.eq.aa0) then
hete_2e_ang_eta_sq=aa0
return
end if
hete_2e_ang_eta_sq=eta**2*
1 het_e_ang(R, z1,z2, p1,p2, eta, m1,m2, L,mu,
1  n1,n2, f1,f2,norb1,norb2)
end

function het_e_ang_gen(etai,etaf, R,
1  z1,z2, p1,p2, m1,m2,L,mu, n1,n2, f1,f2, norb1,norb2)
implicit real*8 (a-h,o-z)
common /cRzpLmu/RR, zz1,zz2, pp1,pp2, LL,mmu
common /cFhete_2e_ang_gen/mm1,mm2, nn1,nn2, ff1,ff2, nnorb1,nnorb2
real*8 f1(0:50),f2(0:50),ff1(0:50),ff2(0:50)
parameter (aa1=1.d0,am1=-1.d0)
character c1*225, c2*2800

write (c1,*) R,z1,z2,p1,p2,L,mu
read (c1,*) RR,zz1,zz2,pp1,pp2,LL,mmu
write (c2,*) m1, m2, n1, n2, f1, f2, norb1, norb2
read (c2,*) mm1, mm2, nn1, nn2, ff1, ff2, nnorb1,nnorb2

hete_2e_ang_gen=hete_2e_ang_gen_Hol(etai,etaf)
print *, 'hete_2e_ang_gen= ',hete_2e_ang_gen
end

```

```
function hete_2e_ang_gen_Hol(a,b)
implicit real*8 (a-h,o-z)
external Fhete_2e_ang_gen
parameter (aa10=10.d0)
common/ck/k,mo
do k=15,15
ep=aa10**(-k)
hete_2e_ang_gen_Hol=
1 qromb_DQDAGS_2(0,1,a,b,EP,mo,2,Fhete_2e_ang_gen)
print *, hete_2e_ang_gen_Hol,k,mo
end do
end
```

```
function Fhete_2e_ang_gen(eta)
implicit real*8 (a-h,o-z)
real*8 f1(0:50),f2(0:50)
common /cRzpLmu/R, z1,z2, p1,p2, L,mu
common /cFhete_2e_ang_gen/m1, m2, n1, n2, f1, f2, norb1, norb2
Fhete_2e_ang_gen=
1 hete_2e_ang(R, z1,z2, p1,p2, eta, m1,m2, L,mu,
1 n1,n2, f1,f2, norb1,norb2)
return
end
```

```
function hete_2e_ang(R, z1,z2, p1,p2, eta, m1,m2,
1 LL,mu,n1,n2, f1,f2, norb1,norb2)
implicit real*8 (a-h,o-z)
real*8 f1(0:50),f2(0:50)
parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0)
parameter (norbp=10)
```

```
common/czszdfgnfngG2/zs_2 (norbp) , zd_2 (norbp) , Rjump_2 (norbp) ,  
1p_2 (norbp) , f_2 (0:50, norbp) , g_2 (0:50, norbp) , nf_2 (norbp) , ng_2 (norbp) ,  
1 N_2 (norbp) , L_2 (norbp) , M_2 (norbp)  
common/czszdfgnfngE2/zs_2E (norbp) , zd_2E (norbp) , Rjump_2E (norbp) ,  
1p_2E (norbp) , f_2E (0:50, norbp) , g_2E (0:50, norbp) , nf_2E (norbp) ,  
1 ng_2E (norbp) , N_2E (norbp) , L_2E (norbp) , M_2E (norbp)  
common/czazb/za (norbp) , zb (norbp)  
common/czazbE/zaE (norbp) , zbE (norbp)  
common/cicenG2/icen_2 (norbp) , lstart_2 (norbp)  
common/cicenE/icenE (norbp) , lstartE (norbp)  
ma=abs (mu)  
  
abet=abs (eta)  
  
icen1=icenE (norb1)  
icen2=icen_2 (norb2)  
  
lstart1=lstartE (norb1)  
lstart2=lstart_2 (norb2)  
  
za1=zaE (norb1)  
zb1=zbE (norb1)  
za2=za (norb2)  
zb2=zb (norb2)  
  
Rjump1=Rjump_2E (norb1)  
Rjump2=Rjump_2 (norb2)  
  
inv1=1  
if (za1.lt.zb1) inv1=-1
```

```
inv2=1
if (za2.lt.zb2) inv2=-1

isgn1=1-2*(icen1-1)
isgn2=1-2*(icen2-1)

GP0=ALegP(LL,ma,eta)
if (GP0.eq.aa0) then
hete_2e_ang=aa0
return
end if

c for icen=1 choose Exp(+p1*eta)*ALegP(m1+i1,m1,+eta)
c for icen=2 choose Exp(-p1*eta)*ALegP(m1+i1,m1,-eta)

if (za1.eq.zb1) then
sum1=aa0
do i1=0,n1
GP=ALegP(2*i1+lstart1,m1,eta)
sum1=sum1+f1(i1)*GP
end do
if (sum1.eq.aa0) then
hete_2e_ang=aa0
return
end if
else if (za1.ne.zb1) then
if (R.lt.Rjump1) then
sum1=aa0
do i1=0,n1
GP=Exp(isgn1*p1*eta)*ALegP(2*i1+lstart1,m1,isgn1*eta)
```

```

    sum1=sum1+f1(i1)*GP
end do
if (sum1.eq.aa0) then
hete_2e_ang=aa0
return
end if
else if (R.gt.Rjump1) then

if (abet.eq.aa1.and.m1.ne.0) then
hete_2e_ang=aa0
return
end if
c for icen=1 choose (aa1-eta**2)**(m1/aa2)*Exp(-p1*eta)*sum[(aa1+eta)
c for icen=2 choose (aa1-eta**2)**(m1/aa2)*Exp(+p1*eta)*sum[(aa1-eta)
sum1=aa0
do i1=0,n1
    sum1=sum1+f1(i1)*(aa1+isgn1*eta)**(2*i1+lstart1)
end do
sum1=sum1*(aa1-eta**2)**(m1/aa2)*Exp(-isgn1*p1*eta)
end if ! if (R.lt.Rjump1) then
end if ! if (za1.eq.zb1) then

c for icen=1 choose Exp(+p2*eta)*ALegP(m2+i2,m2,+eta)
c for icen=2 choose Exp(-p2*eta)*ALegP(m2+i2,m2,-eta)
if (za2.eq.zb2) then
sum2=aa0
do i2=0,n2
GP=ALegP(2*i2+lstart2,m2,eta)
    sum2=sum2+f2(i2)*GP
end do

```

```

if (sum2.eq.aa0) then
hete_2e_ang=aa0
return
end if
else if (za2.ne.zb2) then
if (R.lt.Rjump2) then
sum2=aa0
do i2=0,n2
GP=Exp(isgn2*p2*eta)*ALegP(2*i2+lstart2,m2,isgn2*eta)
sum2=sum2+f2(i2)*GP
end do
if (sum2.eq.aa0) then
hete_2e_ang=aa0
return
end if
else if (R.gt.Rjump2) then

if (abet.eq.aa1.and.m2.ne.0) then
hete_2e_ang=aa0
return
end if
c for icen=1 choose (aa1-eta**2)**(m2/aa2)*Exp(-p2*eta)*sum[(aa1+eta)
c for icen=2 choose (aa1-eta**2)**(m2/aa2)*Exp(+p2*eta)*sum[(aa1-eta)
sum2=aa0
do i2=0,n2
sum2=sum2+f2(i2)*(aa1+isgn2*eta)**(2*i2+lstart2) (aa1+isgn2*eta)**
end do
sum2=sum2*(aa1-eta**2)**(m2/aa2)*Exp(-isgn2*p2*eta)
end if ! if (R.lt.Rjump2) then
end if ! if (za2.eq.zb2) then

```

```

hete_2e_ang=sum1*sum2*GP0
return
end

      FUNCTION ALegP(l,mm,x)
      INTEGER l,m
      REAL*8 ALegP,x
      INTEGER i,ll
      REAL*8 fact,p11,pmm,pmp1,somx2
parameter (aa1=1.d0,aa2=2.d0)
parameter (n0=0,nFac=29)
common /cfac/fac(n0:nfac)
m=abs(mm)
      if(m.lt.0.or.m.gt.1.or.abs(x).gt.aa1)then
print*,'bad arguments in ALegP'
pause
end if
pmm=aa1
if(m.gt.0) then
      somx2=sqrt((aa1-x)*(aa1+x))
      fact=aa1
      do 11 i=1,m
          pmm=-pmm*fact*somx2
          fact=fact+aa2
11      continue
endif
if(l.eq.m) then
      ALegP=pmm
else
      pmp1=x*(aa2*m+aa1)*pmm
      if(l.eq.m+1) then

```



```

        ALegP=pmp1
    else
        do 12 ll=m+2,1
            pll=(x*(aa2*ll-aa1)*pmp1-(ll+m-aa1)*pmm)/(ll-m)
            pmm=pmp1
            pmp1=pll
12        continue
        ALegP=pll
    endif
endif
if(mm.lt.0)
1 ALegP=(-1)**m*(fac(l-m)/fac(l+m))*ALegP
    return
END

```

Δ.3.2 Ακτινικά Ολοκληρώματα

```

function hete_2e_Rad_1(xii,xif)
implicit real*8 (a-h,o-z)
parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0,aa4=4.d0,aa7=7.d0,bb7=1.d-7,
1 bb3=1.d-3,tol=1.d-2)
        external Fhete_2e_Rad_1_prod_in_x_out,
1 Fhete_2e_Rad_1_prod_in_x_out_small

common/ciQDAGS/EPS1,iQDAGS

EPS1=1.d-8
hete_2e_Rad_1=hete_2e_Rad_1_Hol_out(xii,xif)
print *,'hete_2e_Rad_1= ',hete_2e_Rad_1
end

```

```

function hete_2e_Rad_1_Hol_out(a,b)
    implicit real*8(a-h,o-z)
    external Fhete_2e_Rad_1_prod_in_x_out
parameter (aa10=10.d0)
common/ckk1/k1,mo1
    do k=15,15
    do k1=10,10
ep=aa10**(-k)
hete_2e_Rad_1_Hol_out=
1 qromb_DQDAGS(0,0,a,b,EP,mo,2,Fhete_2e_Rad_1_prod_in_x_out)
print *, hete_2e_Rad_1_Hol_out,k,k1,mo,mo1
    end do !k1=1,16
    end do !k=1,15
end

    function Fhete_2e_Rad_1_prod_in_x_out(xi1)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
parameter (aa0=0.d0)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common /cxii/a
common/cFhete_2e_Rad_out/mi, mk, ni, nk, gi, gk
if (a.eq.xi1) then
Fhete_2e_Rad_1_prod_in_x_out=aa0
else
out=hete_2e_Rad_1_out(R, zi,zk, pi,pk, xi1, mi,mk, L,mu,
1 ni,nk, gi,gk)
if(out.eq.aa0)then
Hol_in=aa0
Fhete_2e_rad_1_prod_in_x_out=aa0

```

```

else
Hol_in=hete_2e_rad_1_Hol_in(a,xi1)
Fhete_2e_rad_1_prod_in_x_out= Hol_in*out
end if
end if
        end

function hete_2e_Rad_1_out (R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
        hete_2e_Rad_1_out=FLegQxisq(R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)
end

function hete_2e_Rad_1_Hol_in(a,xi1)
        implicit real*8(a-h,o-z)
        external Fhete_2e_Rad_1_in
parameter (aa0=0.d0,aa10=10.d0)
real*8 gj(0:50),gl(0:50)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_Rad_in/mj, ml, nj, nl, gj, gl
common/ckk1/k1,mo1

ep=aa10**(-k1)
if (a.eq.xi1) then
hete_2e_Rad_1_Hol_in=aa0
else
hete_2e_Rad_1_Hol_in=
1 qromb_DQDAGS_1(0,0,a,xi1,EP,Mo1,2,Fhete_2e_Rad_1_in)
end if

```

```

end

function Fhete_2e_Rad_1_in(xi2)
implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_Rad_in/mj, ml, nj, nl, gj, gl
Fhete_2e_Rad_1_in=
1 hete_2e_Rad_1_in(R, zj,zl, pj,pl, xi2, mj,ml, L,mu,
1 nj,nl, gj,gl)
return
end

function hete_2e_Rad_1_in(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)
hete_2e_Rad_1_in=FLegPxisq(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
end

function hete_2e_Rad_2(xii,xif)
implicit real*8 (a-h,o-z)
parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0,aa4=4.d0,aa7=7.d0,bb7=1.d-7,
1 bb3=1.d-3,tol=1.d-2)
external Fhete_2e_Rad_2_prod_in_x_out,
1 Fhete_2e_Rad_2_prod_in_x_out_small
common/ciQDAGS/EPS1,iQDAGS
EPS1=1.d-8
hete_2e_Rad_2=hete_2e_Rad_2_Hol_out(xii,xif)
print *,'hete_2e_Rad_2= ',hete_2e_Rad_2

```

```

end

function hete_2e_Rad_2_Hol_out(a,b)
    implicit real*8(a-h,o-z)
    external Fhete_2e_Rad_2_prod_in_x_out
parameter (aa10=10.d0)
common/ckk1/k1,mo1
    do k=15,15
        do k1=10,10
ep=aa10**(-k)
hete_2e_Rad_2_Hol_out=
1 qromb_DQDAGS(0,0,a,b,EP,mo,2,Fhete_2e_Rad_2_prod_in_x_out)
print *, hete_2e_Rad_2_Hol_out,k,k1,mo,mo1
        end do !k1=1,16

    end do !k=1,15
end

function Fhete_2e_Rad_2_prod_in_x_out(xi1)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
parameter (aa0=0.d0)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common /cxii/a
common/cFhete_2e_Rad_out/mi, mk, ni, nk, gi, gk
if (a.eq.xi1) then
Fhete_2e_Rad_2_prod_in_x_out=aa0
else
out=hete_2e_Rad_2_out(R, zi,zk, pi,pk, xi1, mi,mk, L,mu,
1 ni,nk, gi,gk)
if(out.eq.aa0)then

```

```

Hol_in=aa0
Fhete_2e_rad_2_prod_in_x_out=aa0
else
Hol_in=hete_2e_rad_2_Hol_in(a,xi1)
Fhete_2e_rad_2_prod_in_x_out= Hol_in*out
end if
end if
        return
        end

function hete_2e_Rad_2_out(R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
        hete_2e_Rad_2_out=FLegQxisq(R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)
end

function hete_2e_Rad_2_Hol_in(a,xi1)
        implicit real*8(a-h,o-z)
        external Fhete_2e_Rad_2_in
parameter (aa0=0.d0,aa10=10.d0)
real*8 gj(0:50),gl(0:50)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_Rad_in/mj, ml, nj, nl, gj, gl
common/ckk1/k1,mol
ep=aa10**(-k1)
if (a.eq.xi1) then
hete_2e_Rad_2_Hol_in=aa0
else
hete_2e_Rad_2_Hol_in=

```

```

1 qromb_DQDAGS_1(0,0,a,xi1,EP,Mo1,2,Fhete_2e_Rad_2_in)
end if
    end

        function Fhete_2e_Rad_2_in(xi2)
implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_Rad_in/mj, ml, nj, nl, gj, gl
    Fhete_2e_Rad_2_in=
1 hete_2e_Rad_2_in(R, zj,zl, pj,pl, xi2, mj,ml, L,mu,
    1 nj,nl, gj,gl)
    return
    end

        function hete_2e_Rad_2_in(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)
hete_2e_Rad_2_in=FLegP(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
end

function hete_2e_Rad_3(xii,xif)
implicit real*8 (a-h,o-z)
parameter (aa1=1.d0,aa2=2.d0,aa4=4.d0,aa7=7.d0,bb7=1.d-7,
1 tol=1.d-2)
    external Fhete_2e_Rad_3_prod_in_x_out,
1 Fhete_2e_Rad_3_prod_in_x_out_small
common /cxif/b

```

```
common/ciQDAGS/EPS1,iQDAGS
iQDAGS=1
EPS1=1.d-8
hete_2e_Rad_3=hete_2e_Rad_3_Hol_out(xii,xif)
print *,'hete_2e_Rad_3= ',hete_2e_Rad_3
end

function Fhete_2e_Rad_3_prod_in_x_out_small(xi)
implicit real*8 (a-h,o-z)
common/cfsmall/fsmall
Fhete_2e_Rad_3_prod_in_x_out_small=
1 Fhete_2e_Rad_3_prod_in_x_out(xi)-fsmall
end

function hete_2e_rad_3_Hol_out(a,bb)
implicit real*8(a-h,o-z)
external Fhete_2e_rad_3_prod_in_x_out
parameter (aa10=10.d0)
common/ckk1/k1,mo1
common /cxif/b
b=bb
do k=15,15
do k1=10,10
ep=aa10**(-k)
hete_2e_rad_3_Hol_out=
1 qromb_DQDAGS(0,0,a,b,EP,mo,2,Fhete_2e_rad_3_prod_in_x_out)
print *, hete_2e_rad_3_Hol_out,k,k1,mo,mo1
end do !k1=1,16
end do !k=1,15
end
```



```

function Fhete_2e_Rad_3_prod_in_x_out(xi1)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
parameter (aa0=0.d0)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common /cxii/a
common /cxif/b
common /cFhete_2e_Rad_out/mi, mk, ni, nk, gi, gk
if (b.eq.xi1) then
Fhete_2e_Rad_3_prod_in_x_out=aa0
else
out=hete_2e_Rad_3_out(R, zi,zk, pi,pk, xi1, mi,mk, L,mu,
1 ni,nk, gi,gk)
if(out.eq.aa0)then
Hol_in=aa0
Fhete_2e_rad_3_prod_in_x_out=aa0
else
Hol_in=hete_2e_rad_3_Hol_in(xi1,b)
Fhete_2e_rad_3_prod_in_x_out= Hol_in*out
end if
end if
end

function hete_2e_rad_3_out(R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
hete_2e_rad_3_out=FLegPxisq(R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)

```

```

end

function hete_2e_rad_3_Hol_in(xi1,b)
    implicit real*8(a-h,o-z)
    external Fhete_2e_rad_3_in
parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0,aa4=4.d0,aa7=7.d0,
1 aa10=10.d0,bb8=1.d-8,bb6=1.d-6,bb3=1.d-3,bb5=1.d-5,tol=1.d-2)
real*8 gj(0:50),gl(0:50)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_rad_in/mj, ml, nj, nl, gj, gl
common/ckk1/k1,mo1
common/ciQDAGS/EPS1,iQDAGS

if (b.eq.xi1) then
hete_2e_rad_3_Hol_in=aa0
else
EPS1=1.d-8
if (xi1.eq.aa1) then
iQDAGS=1
else
iQDAGS=0
end if
hete_2e_rad_3_Hol_in=
1 qromb_DQDAGS_1(iQDAGS,0,xi1,b,
1 EP,Mo1,2,Fhete_2e_rad_3_in)
end if
    end

    function Fhete_2e_rad_3_in(xi2)
implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)

```

```

common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_rad_in/mj, ml, nj, nl, gj, gl
      Fhete_2e_rad_3_in=
1 hete_2e_rad_3_in(R, zj,zl, pj,pl, xi2, mj,ml, L,mu,
      1 nj,nl, gj,gl)
      return
      end

      function hete_2e_rad_3_in(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)
hete_2e_rad_3_in=FLegQxisq(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
end

function hete_2e_Rad_4(xii,xif)
implicit real*8 (a-h,o-z)
parameter (aa1=1.d0,aa2=2.d0,aa4=4.d0,aa7=7.d0,bb7=1.d-7,
1 tol=1.d-2)
      external Fhete_2e_Rad_4_prod_in_x_out,
1 Fhete_2e_Rad_4_prod_in_x_out_small
common /cxif/b
common/ciQDAGS/EPS1,iQDAGS
iQDAGS=1
EPS1=1.d-8
hete_2e_Rad_4=hete_2e_Rad_4_Hol_out(xii,xif)
print *, 'hete_2e_Rad_4= ',hete_2e_Rad_4
end

```

```

function Fhete_2e_Rad_4_prod_in_x_out_small(xi)
implicit real*8 (a-h,o-z)
common/cfsmall/fsmall
Fhete_2e_Rad_4_prod_in_x_out_small=
1 Fhete_2e_Rad_4_prod_in_x_out(xi)-fsmall
end

```

```

function hete_2e_Rad_4_Hol_out(a,bb)
    implicit real*8(a-h,o-z)
    external Fhete_2e_Rad_4_prod_in_x_out
parameter (aa10=10.d0)
common/ckk1/k1,mo1
common /cxif/b
b=bb
    do k=15,15
    do k1=10,10
ep=aa10**(-k)
hete_2e_Rad_4_Hol_out=
1 qromb_DQDAGS(0,0,a,b,EP,mo,2,Fhete_2e_Rad_4_prod_in_x_out)
print *, hete_2e_Rad_4_Hol_out,k,k1,mo,mo1
    end do !k1=1,16
    end do !k=1,15
end

```

```

function Fhete_2e_Rad_4_prod_in_x_out(xi1)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
parameter (aa0=0.d0)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu

```

```

common /cxii/a
common /cxif/b
common /cFhete_2e_Rad_out/mi, mk, ni, nk, gi, gk
if (b.eq.xi1) then
Fhete_2e_Rad_4_prod_in_x_out=aa0
else
out=hete_2e_Rad_4_out(R, zi,zk, pi,pk, xi1, mi,mk, L,mu,
1 ni,nk, gi,gk)
if(out.eq.aa0)then
Hol_in=aa0
Fhete_2e_rad_4_prod_in_x_out=aa0
else
      Hol_in=hete_2e_rad_4_Hol_in(xi1,b)
Fhete_2e_rad_4_prod_in_x_out= Hol_in*out
end if
end if
      end

      function hete_2e_rad_4_out(R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
      hete_2e_rad_4_out=FLegPxisq(R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)
end

function hete_2e_rad_4_Hol_in(xi1,b)
      implicit real*8(a-h,o-z)
      external Fhete_2e_rad_4_in
parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0,aa4=4.d0,aa7=7.d0,
1 aa10=10.d0,bb3=1.d-3,bb5=1.d-5,tol=1.d-2)

```

```

real*8 gj(0:3),gl(0:3)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_rad_in/mj, ml, nj, nl, gj, gl
common/ckk1/k1,mo1
common/ciQDAGS/EPS1,iQDAGS
if (b.eq.xi1) then
hete_2e_rad_4_Hol_in=aa0
else
EPS1=1.d-8
if (xi1.eq.aa1) then
iQDAGS=1
else
iQDAGS=0
end if
hete_2e_rad_4_Hol_in=
1 qromb_DQDAGS_1(iQDAGS,0,xi1,b,
1 EP,Mo1,2,Fhete_2e_rad_4_in)
end if
      end

      function Fhete_2e_rad_4_in(xi2)
implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_rad_in/mj, ml, nj, nl, gj, gl
Fhete_2e_rad_4_in=
1 hete_2e_rad_4_in(R, zj,zl, pj,pl, xi2, mj,ml, L,mu,
1 nj,nl, gj,gl)
      return
      end

```

```

function hete_2e_rad_4_in(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)
hete_2e_rad_4_in=FLegQ(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
end

```

```

function hete_2e_Rad_5(xii,xif)
implicit real*8 (a-h,o-z)
parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0,aa4=4.d0,aa7=7.d0,bb7=1.d-7,
1 bb3=1.d-3,tol=1.d-2)
      external Fhete_2e_Rad_5_prod_in_x_out,
1 Fhete_2e_Rad_5_prod_in_x_out_small
common/ciQDAGS/EPS1,iQDAGS
EPS1=1.d-8
hete_2e_Rad_5=hete_2e_Rad_5_Hol_out(xii,xif)
print *,'hete_2e_Rad_5= ',hete_2e_Rad_5
end

```

```

function Fhete_2e_Rad_5_prod_in_x_out_small(xi)
implicit real*8 (a-h,o-z)
common/cfsmall/fsmall
Fhete_2e_Rad_5_prod_in_x_out_small=
1 abs(Fhete_2e_Rad_5_prod_in_x_out(xi))-fsmall
end

```

```

function hete_2e_rad_5_Hol_out(a,b)
      implicit real*8(a-h,o-z)

```

```

        external Fhete_2e_rad_5_prod_in_x_out
parameter (aa10=10.d0)

common/ckk1/k1,mo1
    do k=15,15
    do k1=10,10
ep=aa10**(-k)
hete_2e_rad_5_Hol_out=
1 qromb_DQDAGS(0,0,a,b,EP,mo,2,Fhete_2e_rad_5_prod_in_x_out)
print *, hete_2e_rad_5_Hol_out,k,k1,mo,mo1
    end do !k1=1,16
    end do !k=1,15
end

        function Fhete_2e_rad_5_prod_in_x_out(xi1)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
parameter (aa0=0.d0)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common /cxii/a
common/cFhete_2e_rad_out/mi, mk, ni, nk, gi, gk
if (a.eq.xi1) then
Fhete_2e_rad_5_prod_in_x_out=aa0
else
out=hete_2e_rad_5_out(R, zi,zk, pi,pk, xi1, mi,mk, L,mu,
1 ni,nk, gi,gk)
if(out.eq.aa0)then
Hol_in=aa0
Fhete_2e_rad_5_prod_in_x_out=aa0
else
Hol_in=hete_2e_rad_5_Hol_in(a,xi1)

```



```

Fhete_2e_rad_5_prod_in_x_out= Hol_in*out
end if
end if
      end

function hete_2e_rad_5_out(R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
      hete_2e_rad_5_out=FLegQ(R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)
end

function hete_2e_rad_5_Hol_in(a,xi1)
      implicit real*8(a-h,o-z)
      external Fhete_2e_rad_5_in
parameter (aa0=0.d0,aa10=10.d0)
real*8 gj(0:50),gl(0:50)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_rad_in/mj, ml, nj, nl, gj, gl
common/ckk1/k1,mo1
ep=aa10**(-k1)
if (a.eq.xi1) then
hete_2e_rad_5_Hol_in=aa0
else
hete_2e_rad_5_Hol_in=
1 qromb_DQDAGS_1(0,0,a,xi1,EP,Mo1,2,Fhete_2e_rad_5_in)
end if
      end

      function Fhete_2e_rad_5_in(xi2)

```

```

implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_rad_in/mj, ml, nj, nl, gj, gl
      Fhete_2e_rad_5_in=
1 hete_2e_rad_5_in(R, zj,zl, pj,pl, xi2, mj,ml, L,mu,
      1 nj,nl, gj,gl)
      return
      end

      function hete_2e_rad_5_in(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)
hete_2e_rad_5_in=FLegPxisq(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
end

function hete_2e_Rad_6(xii,xif)
implicit real*8 (a-h,o-z)
parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0,aa4=4.d0,aa7=7.d0,bb7=1.d-7,
1 bb3=1.d-3,tol=1.d-2)
      external Fhete_2e_Rad_6_prod_in_x_out,
1 Fhete_2e_Rad_6_prod_in_x_out_small
common/ciQDAGS/EPS1,iQDAGS
EPS1=1.d-8
hete_2e_Rad_6=hete_2e_Rad_6_Hol_out(xii,xif)
print *,'hete_2e_Rad_6= ',hete_2e_Rad_6
end

```

```

function hete_2e_rad_6_Hol_out(a,b)
    implicit real*8(a-h,o-z)
    external Fhete_2e_rad_6_prod_in_x_out
parameter (aa10=10.d0)
common/ckk1/k1,mo1
    do k=15,15
    do k1=10,10
ep=aa10**(-k)
hete_2e_rad_6_Hol_out=
1 qromb_DQDAGS(0,0,a,b,EP,mo,2,Fhete_2e_rad_6_prod_in_x_out)
print *, hete_2e_rad_6_Hol_out,k,k1,mo,mo1
    enddo !k1=1,16
    end do !k=1,15
end

```

```

function Fhete_2e_rad_6_prod_in_x_out(xi1)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
parameter (aa0=0.d0)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common /cxii/a
common/cFhete_2e_rad_out/mi, mk, ni, nk, gi, gk
if (a.eq.xi1) then
Fhete_2e_rad_6_prod_in_x_out=aa0
else
out=hete_2e_rad_6_out(R, zi,zk, pi,pk, xi1, mi,mk, L,mu,
1 ni,nk, gi,gk)
if(out.eq.aa0)then
Hol_in=aa0
Fhete_2e_rad_6_prod_in_x_out=aa0
else

```

```

Hol_in=hete_2e_rad_6_Hol_in(a,xi1)
Fhete_2e_rad_6_prod_in_x_out= Hol_in*out
end if
end if
      end

function hete_2e_rad_6_out(R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
      hete_2e_rad_6_out=FLegQ(R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)
end

function hete_2e_rad_6_Hol_in(a,xi1)
      implicit real*8(a-h,o-z)
      external Fhete_2e_rad_6_in
parameter (aa0=0.d0,aa10=10.d0)
real*8 gj(0:50),gl(0:50)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_rad_in/mj, ml, nj, nl, gj, gl
common/ckk1/k1,mo1
ep=aa10**(-k1)
if (a.eq.xi1) then
hete_2e_rad_6_Hol_in=aa0
else
hete_2e_rad_6_Hol_in=
1 qromb_DQDAGS_1(0,0,a,xi1,EP,Mo1,2,Fhete_2e_rad_6_in)
end if
      end

```

```

function Fhete_2e_rad_6_in(xi2)
implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_rad_in/mj, ml, nj, nl, gj, gl
Fhete_2e_rad_6_in=
1 hete_2e_rad_6_in(R, zj,zl, pj,pl, xi2, mj,ml, L,mu,
1 nj,nl, gj,gl)
return
end

```

```

function hete_2e_rad_6_in(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)
hete_2e_rad_6_in=FLegP(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
end

```

```

function hete_2e_Rad_7(xii,xif)
implicit real*8 (a-h,o-z)
parameter (aa1=1.d0,aa2=2.d0,aa4=4.d0,aa7=7.d0,bb7=1.d-7,
1 tol=1.d-2)
external Fhete_2e_Rad_7_prod_in_x_out,
1 Fhete_2e_Rad_7_prod_in_x_out_small
common /cxif/b
common/ciQDAGS/EPS1,iQDAGS
iQDAGS=1
EPS1=1.d-8

```

```

hete_2e_Rad_7=hete_2e_Rad_7_Hol_out(xii,xif)
print *,'hete_2e_Rad_7= ',hete_2e_Rad_7
end

```

```

function hete_2e_rad_7_Hol_out(a,bb)
    implicit real*8(a-h,o-z)
    external Fhete_2e_rad_7_prod_in_x_out
parameter (aa10=10.d0)
common /cxif/b
common/ckk1/k1,mo1
b=bb
    do k=15,15

        do k1=10,10
ep=aa10**(-k)
hete_2e_rad_7_Hol_out=
1 qromb_DQDAGS(0,0,a,b,EP,mo,2,Fhete_2e_rad_7_prod_in_x_out)
print *, hete_2e_rad_7_Hol_out,k,k1,mo,mo1
        end do !k1=1,16
        end do !k=1,15
    end

```

```

function Fhete_2e_rad_7_prod_in_x_out(xi1)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
parameter (aa0=0.d0)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common /cxii/a
common /cxif/b
common/cFhete_2e_rad_out/mi, mk, ni, nk, gi, gk

```

```

if (b.eq.xi1) then
Fhete_2e_rad_7_prod_in_x_out=aa0
else
out=hete_2e_rad_7_out(R, zi,zk, pi,pk, xi1, mi,mk, L,mu,
1 ni,nk, gi,gk)
if(out.eq.aa0)then
Hol_in=aa0
Fhete_2e_rad_7_prod_in_x_out=aa0
else
      Hol_in=hete_2e_rad_7_Hol_in(xi1,b)
Fhete_2e_rad_7_prod_in_x_out= Hol_in*out
end if
end if
      end

function hete_2e_rad_7_out(R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
      hete_2e_rad_7_out= FLegP(R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)
end

function hete_2e_rad_7_Hol_in(xi1,b)
      implicit real*8(a-h,o-z)
      external Fhete_2e_rad_7_in
parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0,aa4=4.d0,aa7=7.d0,
1 aa10=10.d0,bb3=1.d-3,bb5=1.d-5,tol=1.d-2)
real*8 gj(0:50),gl(0:50)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_rad_in/mj, ml, nj, nl, gj, gl

```

```

common/ckk1/k1,mo1
common/ciQDAGS/EPS1,iQDAGS
if (b.eq.xi1) then
hete_2e_rad_7_Hol_in=aa0
else
EPS1=1.d-8
if (xi1.eq.aa1) then
iQDAGS=1
else
iQDAGS=0
end if
hete_2e_rad_7_Hol_in=
1 qromb_DQDAGS_1(iQDAGS,0,xi1,b,
1 EP,Mo1,2,Fhete_2e_rad_7_in)
end if
      end

```

```

function Fhete_2e_Rad_7_in_smallin(xi)
implicit real*8 (a-h,o-z)
common/cfsmallin/fsmallin
Fhete_2e_Rad_7_in_smallin=
1 abs(Fhete_2e_rad_7_in(xi))-fsmallin
end

```

```

      function Fhete_2e_rad_7_in(xi2)
implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_rad_in/mj, ml, nj, nl, gj, gl
      Fhete_2e_rad_7_in=

```



```

1 hete_2e_rad_7_in(R, zj,zl, pj,pl, xi2, mj,ml, L,mu,
  1 nj,nl, gj,gl)
  return
end

```

```

function hete_2e_rad_7_in(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)
hete_2e_rad_7_in=FLegQxisq(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
end

```

```

function hete_2e_Rad_8(xii,xif)
implicit real*8 (a-h,o-z)
parameter (aa1=1.d0,aa2=2.d0,aa4=4.d0,aa7=7.d0,bb7=1.d-7,
1 tol=1.d-2)
  external Fhete_2e_Rad_8_prod_in_x_out,
1 Fhete_2e_Rad_8_prod_in_x_out_small
common /cxif/b
common/ciQDAGS/EPS1,iQDAGS
iQDAGS=1
EPS1=1.d-8
hete_2e_Rad_8=hete_2e_Rad_8_Hol_out(xii,xif)
print *,'hete_2e_Rad_8= ',hete_2e_Rad_8
end

```

```

function Fhete_2e_Rad_8_prod_in_x_out_small(xi)
implicit real*8 (a-h,o-z)

```

```

common/cfsmall/fsmall
Fhete_2e_Rad_8_prod_in_x_out_small=
1 Fhete_2e_Rad_8_prod_in_x_out(xi)-fsmall
end

function hete_2e_rad_8_Hol_out(a,bb)
    implicit real*8(a-h,o-z)
    external Fhete_2e_rad_8_prod_in_x_out
parameter (aa10=10.d0)
common/ckk1/k1,mol
common /cxif/b
b=bb
    do k=15,15
    do k1=10,10
ep=aa10**(-k)
hete_2e_rad_8_Hol_out=
1 qromb_DQDAGS(0,0,a,b,EP,mo,2,Fhete_2e_rad_8_prod_in_x_out)
print *, hete_2e_rad_8_Hol_out,k,k1,mo,mol
    end do !k1=1,16
    end do !k=1,15
end

function Fhete_2e_rad_8_prod_in_x_out(xi1)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
parameter (aa0=0.d0)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common /cxii/a
common /cxif/b
common/cFhete_2e_rad_out/mi, mk, ni, nk, gi, gk
if (b.eq.xi1) then

```

```

Fhete_2e_rad_8_prod_in_x_out=aa0
else
out=hete_2e_rad_8_out(R, zi,zk, pi,pk, xi1, mi,mk, L,mu,
1 ni,nk, gi,gk)
if(out.eq.aa0)then
Hol_in=aa0
Fhete_2e_rad_8_prod_in_x_out=aa0
else
Hol_in=hete_2e_rad_8_Hol_in(xi1,b)
Fhete_2e_rad_8_prod_in_x_out= Hol_in*out
end if
end if
end

```

```

function hete_2e_rad_8_out(R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)
implicit real*8 (a-h,o-z)
real*8 gi(0:50),gk(0:50)
hete_2e_rad_8_out=FLegP(R, zi,zk, pi,pk, xi1,
1 mi,mk, L,mu, ni,nk, gi,gk)
end

```

```

function hete_2e_rad_8_Hol_in(xi1,b)
implicit real*8(a-h,o-z)
external Fhete_2e_rad_8_in
parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0,aa4=4.d0,aa7=7.d0,
1 aa10=10.d0,bb3=1.d-3,bb5=1.d-5,tol=1.d-2)
real*8 gj(0:50),gl(0:50)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_rad_in/mj, ml, nj, nl, gj, gl
common/ckk1/k1,mol

```

```

common/ciQDAGS/EPS1,iQDAGS
if (b.eq.xi1) then
hete_2e_rad_8_Hol_in=aa0
else
EPS1=1.d-8
if (xi1.eq.aa1) then
iQDAGS=1
else
iQDAGS=0
end if
hete_2e_rad_8_Hol_in=
1 qromb_DQDAGS_1(iQDAGS,0,xi1,b,
1 EP,Mo1,2,Fhete_2e_rad_8_in)
end if
      end

```

```

function Fhete_2e_Rad_8_in_smallin(xi)
implicit real*8 (a-h,o-z)
common/cfsmallin/fsmallin
Fhete_2e_Rad_8_in_smallin=
1 abs(Fhete_2e_rad_8_in(xi))-fsmallin
end

```

```

      function Fhete_2e_rad_8_in(xi2)
implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)
common/cRzpLm/R, zi,zk,zj,zl ,pi,pk,pj,pl, L,mu
common/cFhete_2e_rad_in/mj, ml, nj, nl, gj, gl
      Fhete_2e_rad_8_in=
1 hete_2e_rad_8_in(R, zj,zl, pj,pl, xi2, mj,ml, L,mu,
1 nj,nl, gj,gl)

```

```
        return
    end

function hete_2e_rad_8_in(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
implicit real*8 (a-h,o-z)
real*8 gj(0:50),gl(0:50)
hete_2e_rad_8_in=FLegQ(R, zj,zl, pj,pl, xi2, mj,ml,
1 L,mu, nj,nl, gj,gl)
end

function FLegPxisq(R, z1,z2, p1,p2, xi, m1,m2, L,mu,
1 n1,n2, g1,g2)
implicit real*8 (a-h,o-z)
real*8 g1(0:50),g2(0:50)
FLegPxisq=xi**2*FLegP(R, z1,z2, p1,p2, xi, m1,m2, L,mu, n1,n2,
1 g1,g2)
end

function FLegP(R, z1,z2, p1,p2, xi, m1,m2, L,mu, n1,n2, g1,g2)
implicit real*8 (a-h,o-z)
    dimension PM(0:100,0:100),PD(0:100,0:100)
    real*8 g1(0:50),g2(0:50)
parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0,MM=100)

sum1=0
```

```

do i1=0,n1
  sum1=sum1+g1(i1)*((xi - aa1)/(xi + aa1))**i1
end do
sum2=0
do i2=0,n2
  sum2=sum2+g2(i2)*((xi - aa1)/(xi + aa1))**i2
  end do
Plm=aa0
call LegP(MM,mu,L,xi,PM,PD)
Plm=PM(mu,L)

FLegP=(Plm*sum1*sum2*
- (-aa1 + xi)**(m1/aa2 + m2/aa2)*
- (aa1 + xi)**
- (-aa2*aa1 - m1/aa2 - m2/aa2 + (R*z1)/(aa2*p1) +
- (R*z2)/(aa2*p2))
- /Exp(p1*xi + p2*xi)
end

```

```

subroutine LegP(MM,M,N,X,PM,PD)
C
C =====
C Purpose: Compute the associated Legendre functions
C          Pmn(x) and their derivatives Pmn'(x)
C Input :  x --- Argument of Pmn(x)
C          m --- Order of Pmn(x), m = 0,1,2,...,n
C          n --- Degree of Pmn(x), n = 0,1,2,...,N
C          mm --- Physical dimension of PM and PD
C Output:  PM(m,n) --- Pmn(x)
C          PD(m,n) --- Pmn'(x)
C =====

```

```

C
      IMPLICIT DOUBLE PRECISION (LegP,P,X)
      DIMENSION PM(0:MM,0:N+1),PD(0:MM,0:N+1)
      DO 10 I=0,N
      DO 10 J=0,M
          PM(J,I)=0.0D0
10      PD(J,I)=0.0D0
      PM(0,0)=1.0D0
      IF (DABS(X).EQ.1.0D0) THEN
          DO 15 I=1,N
              PM(0,I)=X**I
15          PD(0,I)=0.5D0*I*(I+1.0D0)*X**(I+1)
          DO 20 J=1,N
          DO 20 I=1,M
              IF (I.EQ.1) THEN
                  PD(I,J)=1.0D+300
              ELSE IF (I.EQ.2) THEN
                  PD(I,J)=-0.25D0*(J+2)*(J+1)*J*(J-1)*X**(J+1)
              ENDIF
20          CONTINUE
          RETURN
      ENDIF
      LS=1
      IF (DABS(X).GT.1.0D0) LS=-1
      XQ=DSQRT(LS*(1.0D0-X*X))
      XS=LS*(1.0D0-X*X)
      DO 30 I=1,M
30          PM(I,I)=-LS*(2.0D0*I-1.0D0)*XQ*PM(I-1,I-1)
      DO 35 I=0,M
35          PM(I,I+1)=(2.0D0*I+1.0D0)*X*PM(I,I)
      DO 40 I=0,M

```

```

          DO 40 J=I+2,N
              PM(I,J)=((2.0D0*J-1.0D0)*X*PM(I,J-1)-
&              (I+J-1.0D0)*PM(I,J-2))/(J-I)
40      CONTINUE
c      PD(0,0)=0.0D0
c      DO 45 J=1,N
c45      PD(0,J)=LS*J*(PM(0,J-1)-X*PM(0,J))/XS
c      DO 50 I=1,M
c      DO 50 J=I,N
c      PD(I,J)=LS*I*X*PM(I,J)/XS+(J+I)
c      &      *(J-I+1.0D0)/XQ*PM(I-1,J)
c50      CONTINUE
c LegP=PM(M,N)
c print*, 'Plm=', LegP
      RETURN
      END

```

```

function FLegQxisq(R, z1,z2, p1,p2, xi, m1,m2, L,mu, n1,n2, g1,g2)
implicit real*8 (a-h,o-z)
real*8 g1(0:50),g2(0:50)
      FLegQxisq=xi**2*
1 FLegQ(R, z1,z2, p1,p2, xi, m1,m2, L,mu, n1,n2, g1,g2)
end

```

```

function FLegQ(R, z1,z2, p1,p2, xi, m1,m2, L,mu, n1,n2, g1,g2)
implicit real*8 (a-h,o-z)
      dimension QM(0:100,0:100),QD(0:100,0:100)
real*8 g1(0:50),g2(0:50)
parameter (SqrtPi=1.7724538509055159d0)
parameter (aa0=0.d0,aa1=1.d0,aa2=2.d0,MM=100)

```



```

sum1=0
do i1=0,n1
  sum1=sum1+g1(i1)*((xi - aa1)/(xi + aa1))**i1
end do
sum2=0
do i2=0,n2
  sum2=sum2+g2(i2)*((xi - aa1)/(xi + aa1))**i2
end do
Qlm=aa0
call LegQ(MM,mu,L,xi,QM,QD)
Qlm=QM(mu,L)
  FLegQ=(Qlm*sum1*sum2*
-      (-aa1 + xi)**(m1/aa2 + m2/aa2)*
-      (aa1 + xi)**
-      (-aa2 - m1/aa2 - m2/aa2 + (R*z1)/(aa2*p1) +
-      (R*z2)/(aa2*p2)))/
-      (Exp(p1*xi + p2*xi))
end

```

```

subroutine LegQ(MM,M,N,X,QM,QD)
C
C
C =====
C Purpose: Compute the associated Legendre functions of the
C           second kind, Qmn(x) and Qmn'(x)
C Input  :  x  --- Argument of Qmn(x)
C           m  --- Order of Qmn(x)   ( m = 0,1,2,úúú )
C           n  --- Degree of Qmn(x)  ( n = 0,1,2,úúú )
C           mm --- Physical dimension of QM and QD
C Output:  QM(m,n) --- Qmn(x)
C           QD(m,n) --- Qmn'(x)

```

```

C      =====
C
      IMPLICIT DOUBLE PRECISION (LegQ,Q,X)
      DIMENSION QM(0:MM,0:N),QD(0:MM,0:N)
      IF (DABS(X).EQ.1.0D0) THEN
          DO 10 I=0,M
              DO 10 J=0,N
                  QM(I,J)=1.0D+300
                  QD(I,J)=1.0D+300
10          CONTINUE
          RETURN
      ENDIF
      LS=1
      IF (DABS(X).GT.1.0D0) LS=-1
      XS=LS*(1.0D0-X*X)
      XQ=DSQRT(XS)
      Q0=0.5D0*DLOG(DABS((X+1.0D0)/(X-1.0D0)))
      IF (DABS(X).LT.1.0001D0) THEN
if(M.eq.0.and.N.eq.0)then
QM(0,0)=Q0
else if (M.eq.0.and.N.eq.1)then
QM(M,N)=X*Q0-1.0D0
else if (M.eq.1.and.N.eq.0)then
    QM(M,N)=-1.0D0/XQ
    else if (M.eq.1.and.N.eq.1)then
        QM(M,N)=-XQ*(Q0+X/(1.0D0-X*X))
else if (M.eq.0.or.M.eq.1.and.N.gt.1)then
c          DO 15 I=0,1
              DO 15 J=2,N
                  QM(M,J) = ((2.0D0*J-1.0D0)*X*QM(I,J-1)
&                      -(J+I-1.0D0)*QM(I,J-2))/(J-I)

```

```

15          CONTINUE
      else if (M.gt.1.and.N.eq.N) then
c          DO 20 J=0,N
            DO 20 I=2,M
                QM(I,N)=-2.0D0*(I-1.0D0)*X/XQ*QM(I-1,J)-LS*
&                (J+I-1.0D0)*(J-I+2.0D0)*QM(I-2,J)
20          CONTINUE
end if

      ELSE
          IF (DABS(X).GT.1.1D0) THEN
              KM=40+M+N
          ELSE
              KM=(40+M+N)*INT(-1.0-1.8*LOG(X-1.0))
          ENDIF
          QF2=0.0D0
          QF1=1.0D0
          DO 25 K=KM,0,-1
              QF0=((2*K+3.0D0)*X*QF1-(K+2.0D0)*QF2)/(K+1.0D0)
              IF (K.LE.N) QM(0,K)=QF0
              QF2=QF1
25          QF1=QF0
          DO 30 K=0,N
30          QM(0,K)=Q0*QM(0,K)/QF0
          QF2=0.0D0
          QF1=1.0D0
          DO 35 K=KM,0,-1
              QF0=((2*K+3.0D0)*X*QF1-(K+1.0D0)*QF2)/(K+2.0D0)
              IF (K.LE.N) QM(1,K)=QF0
              QF2=QF1
35          QF1=QF0
          Q10=-1.0D0/XQ

```

```

DO 40 K=0,N
40      QM(1,K)=Q10*QM(1,K)/QF0
DO 45 J=0,N
      Q0=QM(0,J)
      Q1=QM(1,J)
DO 45 I=0,M-2
      QF=-2.0D0*(I+1)*X/XQ*Q1+(J-I)*(J+I+1.0D0)*Q0
      QM(I+2,J)=QF
      Q0=Q1
      Q1=QF
45      CONTINUE
      ENDIF
c      QD(0,0)=LS/XS
c      DO 50 J=1,N
c50      QD(0,J)=LS*J*(QM(0,J-1)-X*QM(0,J))/XS
c      DO 55 J=0,N
c      DO 55 I=1,M
c      QD(I,J)=LS*I*X/XS*QM(I,J)+(I+J)*(J-I+1.0D0)/XQ*QM(I-1,J)
c55      CONTINUE
c LegQ= QM(M,N)
c print*, 'Qlm=', LegQ
      RETURN
      END

```

Δ.4 Προγράμματα Ολοκλήρωσης

```

C      implicit real*8(a-h,o-z)
C      external fun
C      a=-1.2345
C      b= .54321

```

```

C      do k=1,16
C      ep=10.**(-k)
C      s=ANC4(a,b,EP,M,2,FUN)
C      d=2.-fun(dabs(a))-fun(dabs(b))-s
C      print *,k,s,d,m
Cc     if (dabs(d).lt.1e-18) stop
C      end do
C      end

c      function fun(x)
c      implicit real*8(a-h,o-z)
c      fun=dexp(-dabs(x))
c      return
c      end
c
      FUNCTION ANC4(A1,B1,EP,M,N,FUN)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      ANC4: ADAPTIVE NEWTON-COATES INTEGRATOR
C      A1= LOWER LIMIT OF INTEGRATION
C      B1= UPPER LIMIT
C      EP= TOLERANCE
C      M= NUMBER OF FUNCTION CALLS REQUIRED
C      N= TYPE OF ERROR TOLERANCE
C      = 1, ABSOLUTE (COMPARED TO UNITY)
C      = 2, RELATIVE (COMPARED TO NET INTEGRAL )
C      = 3, RELATIVE (COMPARED TO MAXIMUM INTEGRAND VALUE)
C      FUN= SINGLE ARGUMENT FUNCTION TO BE EVALUATED
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT REAL*8(A-H,O-Z)

```

```
DIMENSION F2(30),F4(30),FTP(30),FBP(30),FTST(5),EST2(30)
DIMENSION NRTR(30),AEST2(30),XB(30)
parameter (aa63=63.0d0,aa0=0.d0,aa3=3.0d0,aa4=4.0d0,aa05=0.5d0,
1 aa32=32.0d0,aa12=12.0d0,aa90=90.0d0,aa2=2.0d0,aa8=8.0d0,aa5=5.0d0
1 ,aa1=1.0d0,aa7=7.0d0)
```

```
C THE PARAMETER SETUP FOR THE INITIAL CALL
```

```
IF(N.LE.0) THEN
    PRINT *, ' ERROR RETURN-N.LE.0'
    RETURN
END IF
IF(N.GT.3) THEN
    PRINT *, ' ERROR RETURN-N.GT.3'
    RETURN
END IF
if (a1.eq.b1) then
    ANC4=aa0
    return
end if
if (a1.ge.b1) then
    print *, ' error return-b1.le.a1'
    return
end if
```

```
A=A1
B=B1
EPS=EP*aa63
ESUM=aa0
TSUM=aa0
```

```

LVL=1
DA=B-A
FA=FUN (A)
FS=FUN ( (aa3*A+B) /aa4)
FM=FUN ( (A+B) *aa05)
FT=FUN ( (A+aa3*B) /aa4)
FB=FUN (B)
M=5
FMAX=DABS (FA)
FTST (1) =FMAX
FTST (2) =DABS (FS)
FTST (3) =DABS (FM)
FTST (4) =DABS (FT)
FTST (5) =DABS (FB)
DO 100 I=2, 5
    fmax=max (FMAX, FTST (I) )
100    CONTINUE
EST=(aa7* (FA+FB) +aa32* (FS+FT) +aa12*FM ) *DA/aa90
ABSAR=(aa7* (FTST (1) +FTST (5) ) +aa32* (FTST (2) +FTST (4) ) +aa12*
1    FTST (3) ) *DA/aa90
AEST=ABSAR
C    1=RECUR
1    SX=(DA/ (aa2**LVL) ) /aa90
    F1=FUN ( (aa7*A+B) /aa8)
    F3=FUN ( (aa5*A+aa3*B) /aa8)
    F2 (LVL) =FUN ( (aa3*A+aa5*B) /aa8)
    F4 (LVL) =FUN ( (A+aa7*B) /aa8)
    EST1=SX* (aa7* (FA+FM) +aa32* (F1+F3) +aa12*FS)
    FBP (LVL) =FB
    FTP (LVL) =FT
    XB (LVL) =B

```

```

EST2 (LVL) =SX* (aa7* (FM+FB) +aa32* (F2 (LVL) +F4 (LVL) ) +aa12*FT)
SUM=EST1+EST2 (LVL)
FTST (1) =DABS (F1)
FTST (2) =DABS (F2 (LVL) )
FTST (3) =DABS (F3)
FTST (4) =DABS (F4 (LVL) )
FTST (5) =DABS (FM)
AEST1=SX* (aa7* (DABS (FA) +FTST (5) ) +aa32* (FTST (1) +FTST (3) ) +aa12*
1      DABS (FS) )
AEST2 (LVL) =SX* (aa7* (FTST (5) +DABS (FB) ) +aa32* (FTST (2) +FTST (4) ) +
1      aa12*DABS (FT) )
ABSAR=ABSAR-AEST+AEST1+AEST2 (LVL)
M=M+4
if (n.eq.1) DELTA=aa1
if (n.eq.2) DELTA=ABSAR
if (n.eq.3) then
    DO 203 I=1,4
        fmax=max (FMAX, FTST (I) )
203    CONTINUE
    DELTA=FMAX
end if
DAFT=EST-SUM
DIFF=DABS (DAFT)
DAFT=DAFT/aa63
ded=DIFF-EPS*DELTA
IF ((ded.le.aa0.and.lvl.eq.1)
1    .or. (ded.gt.aa0.and.lvl.lt.30) ) then
    NRTR (LVL) =1
    EST=EST1
    AEST=AEST1
    FB=FM

```



```

      FT=F3
      FM=FS
      FS=F1
      B=(A+B)/aa2
      EPS=EPS/aa2
else                                     !   UP
      A=B
      ESUM=ESUM+DAFT
      TSUM=TSUM+SUM
9     LVL=LVL-1
      L=NRTR(LVL)
      if (l.eq.1) then                   !   R1
          NRTR(LVL)=2
          FA=FB
          FS=F2(LVL)
          FM=FTP(LVL)
          FT=F4(LVL)
          FB=FBP(LVL)
          B=XB(LVL)
          EST=EST2(LVL)
          AEST=AEST2(LVL)
      else                                 !   R2
          IF(LVL.le.1) then
              ANC4=TSUM-ESUM
              RETURN
          end if
          EPS=aa2*EPS
          goto 9
      end if
end if
LVL=LVL+1

```

```

GO TO 1
END

FUNCTION ANC41 (A1, B1, EP, M, N, FUN)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C          ANC4: ADAPTIVE NEWTON-COATES INTEGRATOR
C          A1= LOWER LIMIT OF INTEGRATION
C          B1= UPPER LIMIT
C          EP= TOLERANCE
C          M= NUMBER OF FUNCTION CALLS REQUIRED
C          N= TYPE OF ERROR TOLERANCE
C             = 1, ABSOLUTE (COMPARED TO UNITY)
C             = 2, RELATIVE (COMPARED TO NET INTEGRAL )
C             = 3, RELATIVE (COMPARED TO MAXIMUM INTEGRAND VALUE)
C          FUN= SINGLE ARGUMENT FUNCTION TO BE EVALUATED
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION F2 (30), F4 (30), FTP (30), FBP (30), FTST (5), EST2 (30)
      DIMENSION NRTR (30), AEST2 (30), XB (30)
parameter (aa63=63.0d0, aa0=0.0d0, aa3=3.0d0, aa4=4.0d0, aa05=0.5d0,
1 aa32=32.0d0, aa12=12.0d0, aa90=90.0d0, aa2=2.0d0, aa8=8.0d0, aa5=5.0d0
1 , aa1=1.0d0, aa7=7.0d0)

C          THE PARAMETER SETUP FOR THE INITIAL CALL
      IF (N.LE.0) THEN
          PRINT *, ' ERROR RETURN-N.LE.0 '
          RETURN
      END IF
      IF (N.GT.3) THEN

```

```
        PRINT *, '  ERROR RETURN-N.GT.3'
        RETURN
    END IF
    if (a1.eq.b1) then
        ANC41=aa0
        return
    end if
    if (a1.ge.b1) then
        print *, '  error return-b1.le.a1'
        return
    end if

A=A1
B=B1
EPS=EP*aa63
ESUM=aa0
TSUM=aa0
LVL=1
DA=B-A
FA=FUN(A)
FS=FUN((aa3*A+B)/aa4)
FM=FUN((A+B)*aa05)
FT=FUN((A+aa3*B)/aa4)
FB=FUN(B)
M=5
FMAX=DABS(FA)
FTST(1)=FMAX
FTST(2)=DABS(FS)
FTST(3)=DABS(FM)
```

```

      FTST (4) =DABS (FT)
      FTST (5) =DABS (FB)
      DO 100 I=2, 5
          fmax=max (FMAX, FTST (I) )
100      CONTINUE
      EST=(aa7* (FA+FB) +aa32* (FS+FT) +aa12*FM ) *DA/aa90
      ABSAR=(aa7* (FTST (1) +FTST (5) ) +aa32* (FTST (2) +FTST (4) ) +aa12*
1      FTST (3) ) *DA/aa90
      AEST=ABSAR
C      1=RECUR
1      SX=(DA/ (aa2**LVL) ) /aa90
      F1=FUN ( (aa7*A+B) /aa8)
      F3=FUN ( (aa5*A+aa3*B) /aa8)
      F2 (LVL) =FUN ( (aa3*A+aa5*B) /aa8)
      F4 (LVL) =FUN ( (A+aa7*B) /aa8)
      EST1=SX* (aa7* (FA+FM) +aa32* (F1+F3) +aa12*FS)
      FBP (LVL) =FB
      FTP (LVL) =FT
      XB (LVL) =B
      EST2 (LVL) =SX* (aa7* (FM+FB) +aa32* (F2 (LVL) +F4 (LVL) ) +aa12*FT)
      SUM=EST1+EST2 (LVL)
      FTST (1) =DABS (F1)
      FTST (2) =DABS (F2 (LVL) )
      FTST (3) =DABS (F3)
      FTST (4) =DABS (F4 (LVL) )
      FTST (5) =DABS (FM)
      AEST1=SX* (aa7* (DABS (FA) +FTST (5) ) +aa32* (FTST (1) +FTST (3) ) +aa12*
1          DABS (FS) )
      AEST2 (LVL) =SX* (aa7* (FTST (5) +DABS (FB) ) +aa32* (FTST (2) +FTST (4) ) +
1          aa12*DABS (FT) )
      ABSAR=ABSAR-AEST+AEST1+AEST2 (LVL)

```

```

M=M+4
if (n.eq.1) DELTA=aa1
if (n.eq.2) DELTA=ABSAR
if (n.eq.3) then
    DO 203 I=1,4
        fmax=max(FMAX,FTST(I))
203    CONTINUE
        DELTA=FMAX
end if
DAFT=EST-SUM
DIFF=DABS(DAFT)
DAFT=DAFT/aa63
ded=DIFF-EPS*DELTA
IF ((ded.le.aa0.and.lvl.eq.1)
1    .or. (ded.gt.aa0.and.lvl.lt.30) ) then
    NRTR(LVL)=1
    EST=EST1
    AEST=AEST1
    FB=FM
    FT=F3
    FM=FS
    FS=F1
    B=(A+B)/aa2
    EPS=EPS/aa2
else
    A=B
    ESUM=ESUM+DAFT
    TSUM=TSUM+SUM
9    LVL=LVL-1
    L=NRTR(LVL)
    if (l.eq.1) then

```

```
NRTR (LVL) =2
FA=FB
FS=F2 (LVL)
FM=FTP (LVL)
FT=F4 (LVL)
FB=FBP (LVL)
B=XB (LVL)
EST=EST2 (LVL)
AEST=AEST2 (LVL)
else                                     !   R2
  IF (LVL.le.1) then
    ANC41=TSUM-ESUM
    RETURN
  end if
  EPS=aa2*EPS
  goto 9
end if
end if
LVL=LVL+1
GO TO 1
END
```

```

      FUNCTION qromb_DQDAGS (iQDAGS, iAng, A1, B1, EP, M, N, FUN)
use imsl
      IMPLICIT REAL*8 (A-H, O-Z)
parameter (aa24=24.d0, aa0=0.d0, bb8=1.d-8, bb12=1.d-12)
c real*8 POINTS(1)
common/ciR8/iR8
common/cRzpLm/R, zi, zk, zj, zl , pi, pk, pj, pl, L, mu
external fun

dx=(B1-A1)/aa24
S=aa0
do ix=1,24
x1=A1+((B1-dx-A1)/23)*(ix-1)
x2=x1+dx
ss=aa0
if (iQDAGS.eq.0)
1 call qromb(fun,x1,x2,ss)
c print*, 'ss qromb',ss
!                               Get output unit number
CALL UMACH (2, NOUT)
ERRABS = 0.0d0
errrel = 0.00001d0
c NPTS = 1
c POINTS(1) = 1.0
if (iQDAGS.eq.1)
c 1 CALL DQDAGP (fun, A1, B1, NPTS, POINTS, ERRABS, ERRREL, ss,
c 2 ERREST)
1 CALL DQDAGS (fun, x1,x2, ERRABS, ERRREL, ss, ERREST)

c write(24,*) x1," ",ss

```

```

S=S+ss
if(x1.gt.0.7.and.abs(ss/S).lt.bb8)go to 1
end do
  1 qromb_DQDAGS=S
c print*, 'x1,x2,ss =',x1,x2,ss,ix

END

SUBROUTINE qromb(func,a,b,ss)
  INTEGER JMAX,JMAXP,K,KM
  DOUBLE PRECISION a,b,func,ss,EPS,EPS1
  EXTERNAL func
  PARAMETER (EPS=1.d-8, JMAX=30, JMAXP=JMAX+1, K=5, KM=K-1,
1 bb16=1.d-16)
CU  USES polint,trapzd
  INTEGER j
  DOUBLE PRECISION dss,h(JMAXP),s(JMAXP)
common/ciQDAGS/EPS1,iQDAGS
c common/ciR3/iR3
  h(1)=1.d0
  do 11 j=1,JMAX
if(j.ge.20) EPS1=1.d-7
if(j.ge.21) EPS1=1.d-7
if(j.ge.22) EPS1=1.d-6
if(j.ge.25) EPS1=1.d-5
if(j.ge.28) EPS1=1.d-4
  call trapzd(func,a,b,s(j),j)
  if (j.ge.K) then
if (abs(s(1))+abs(s(2))+abs(s(3))+abs(s(4))
1 +abs(s(5)).le.bb16)
1 return

```



```
call polint(h(j-KM),s(j-KM),K,0.d0,ss,dss)
if (abs(dss).le.EPS1*abs(ss)) return
c if (iR3.eq.1) print*, 'j, EPS=', j, EPS1
endif
s(j+1)=s(j)
h(j+1)=0.25d0*h(j)
11 continue
pause 'too many steps in qromb'
END
```

```
SUBROUTINE trapzd(func,a,b,s,n)
INTEGER n
DOUBLE PRECISION a,b,s,func
EXTERNAL func
INTEGER it,j
DOUBLE PRECISION del,sum,tnm,x
if (n.eq.1) then
s=0.5d0*(b-a)*(func(a)+func(b))
else
it=2**(n-2)
tnm=it
del=(b-a)/tnm
x=a+0.5d0*del
sum=0.d0
do 11 j=1,it
sum=sum+func(x)
x=x+del
11 continue
s=0.5d0*(s+(b-a)*sum/tnm)
endif
return
```

```
END

SUBROUTINE polint(xa,ya,n,x,y,dy)
INTEGER n,NMAX
DOUBLE PRECISION dy,x,y,xa(n),ya(n)
PARAMETER (NMAX=10)
INTEGER i,m,ns
DOUBLE PRECISION den,dif,dift,ho,hp,w,c(NMAX),d(NMAX)
ns=1
dif=abs(x-xa(1))
do 11 i=1,n
    dift=abs(x-xa(i))
    if (dift.lt.dif) then
        ns=i
        dif=dift
    endif
    c(i)=ya(i)
    d(i)=ya(i)
11 continue
y=ya(ns)
ns=ns-1
do 13 m=1,n-1
    do 12 i=1,n-m
        ho=xa(i)-x
        hp=xa(i+m)-x
        w=c(i+1)-d(i)
        den=ho-hp
        if(den.eq.0.d0)pause 'failure in polint'
        den=w/den
        d(i)=hp*den
        c(i)=ho*den
    enddo
enddo
```

```

12      continue
        if (2*ns.lt.n-m) then
            dy=c(ns+1)
        else
            dy=d(ns)
            ns=ns-1
        endif
        y=y+dy
13      continue
        return
        END

```

```

        FUNCTION qromb_DQDAGS_1(iQDAGS,iAng,A1,B1,EP,M,N,FUN)
use imsl
        IMPLICIT REAL*8(A-H,O-Z)
parameter (aa24=24.d0,aa0=0.d0,bb8=1.d-8,bb12=1.d-12)
c real*8 POINTS(1)
external fun
dx=(B1-A1)/aa24
S=aa0
do ix=1,24
x1=A1+((B1-dx-A1)/23)*(ix-1)
x2=x1+dx
ss=aa0
if (iQDAGS.eq.0)
1 call qromb1(fun,x1,x2,ss)
c print*, 'ss qromb',ss
!
        CALL UMACH (2, NOUT)

```

Get output unit number

```

        ERRABS = 0.d0
errrel = 1.d-5
c if (B1.eq.25.d0)pause
if (iQDAGS.eq.1) then
CALL DQDAGS (fun, x1,x2, ERRABS, ERRREL, ss, ERREST)
c print '(a,2f7.2,1pg25.15)', 'eswterikh oloklhrwsh metexu'
c 1 ,A1,B1,ss
end if
c 1 CALL DQ2AGS (fun, A1, B1, ERRABS, ERRREL, ss, ERREST,
c 1 MAXSUB, NEVAL, NSUBIN, ALIST, BLIST,
c 1 RLIST, ELIST, IORD)
c print*, 'qdags ss',ss
S=S+ss
if(x1.gt.0.7.and.abs(ss/S).lt.bb8)go to 1
end do
  1 qromb_DQDAGS_1=S
c print*, 'x1,x2,ss =',x1,x2,ss,ix
END

SUBROUTINE qromb1(func,a,b,ss)
        INTEGER JMAX,JMAXP,K,KM
        DOUBLE PRECISION a,b,func,ss,EPS,EPS1
        EXTERNAL func
        PARAMETER (EPS=1.d-8, JMAX=30, JMAXP=JMAX+1, K=5, KM=K-1
1 ,bb16=1.d-16)
common/ciQDAGS/EPS1,iQDAGS
CU      USES polint,trapzd
        INTEGER j
        DOUBLE PRECISION dss,h(JMAXP),s(JMAXP)
c common/ciR3/iR3

```

```

ss=0
      h(1)=1.d0
      do 11 j=1,JMAX
if(j.ge.20) EPS1=1.d-7
if(j.ge.21) EPS1=1.d-7
if(j.ge.22) EPS1=1.d-6
if(j.ge.25) EPS1=1.d-5
if(j.ge.28) EPS1=1.d-4
      call trapzd1(func,a,b,s(j),j)
      if (j.ge.K) then
if (abs(s(1))+abs(s(2))+abs(s(3))+abs(s(4))
1 +abs(s(5)).le.bb16)
1 return
      call polint1(h(j-KM),s(j-KM),K,0.d0,ss,dss)
if (abs(dss).le.EPS1*abs(ss)) return
c if (iR3.eq.1) print*, 'j1, EPS1=', j, EPS1
      endif
      s(j+1)=s(j)
      h(j+1)=0.25d0*h(j)
11 continue
      pause 'too many steps in qromb 1'
      END

```

```

SUBROUTINE trapzd1(func,a,b,s,n)
      INTEGER n
      DOUBLE PRECISION a,b,s,func
      EXTERNAL func
      INTEGER it,j
      DOUBLE PRECISION del,sum,tnm,x
if (n.eq.1) then
      s=0.5d0*(b-a)*(func(a)+func(b))

```

```

else
  it=2**(n-2)
  tnm=it
  del=(b-a)/tnm
  x=a+0.5d0*del
  sum=0.d0
  do 11 j=1,it
    sum=sum+func(x)
    x=x+del
11  continue
    s=0.5d0*(s+(b-a)*sum/tnm)
endif
return
END

SUBROUTINE polint1(xa,ya,n,x,y,dy)
  INTEGER n,NMAX
  DOUBLE PRECISION dy,x,y,xa(n),ya(n)
  PARAMETER (NMAX=10)
  INTEGER i,m,ns
  DOUBLE PRECISION den,dif,dift,ho,hp,w,c(NMAX),d(NMAX)
  ns=1
  dif=abs(x-xa(1))
  do 11 i=1,n
    dift=abs(x-xa(i))
    if (dift.lt.dif) then
      ns=i
      dif=dift
    endif
    c(i)=ya(i)
    d(i)=ya(i)

```

```

11  continue
    y=ya (ns)
    ns=ns-1
    do 13 m=1,n-1
        do 12 i=1,n-m
            ho=xa (i)-x
            hp=xa (i+m)-x
            w=c (i+1)-d (i)
            den=ho-hp
            if (den.eq.0.d0)pause 'failure in polint'
            den=w/den
            d (i)=hp*den
            c (i)=ho*den
12  continue
    if (2*ns.lt.n-m) then
        dy=c (ns+1)
    else
        dy=d (ns)
        ns=ns-1
    endif
    y=y+dy
13  continue
    return
    END

    FUNCTION qromb_DQDAGS_2 (iQDAGS, iAng, A1, B1, EP, M, N, FUN)
use imsl
    IMPLICIT REAL*8 (A-H, O-Z)
real*8 POINTS (1)
parameter (aa24=24.d0, aa0=0.d0, bb8=1.d-8,
1  xx1=0.6938166528d0, xx2=0.2109359921d0)

```

```
c real*8 POINTS(1)
external fun
ss=aa0
if (fun(xx1).eq.-fun(-xx1).and.fun(xx2).eq.-fun(-xx2)) then
c print*, 'skip ss1,ss2, ODD FUNCTION'
go to 1
end if

if (iQDAGS.eq.0)
1 call qromb2(fun,A1,B1,ss)
c print*, 'ss qromb',ss
!                               Get output unit number
      CALL UMACH (2, NOUT)
      ERRABS = 0.d0
errrel = 0.00001d0
NPTS = 1

if (iQDAGS.eq.1) then
POINTS(1) = -1.d0
      CALL DQDAGP (fun, A1,aa0, NPTS, POINTS, ERRABS, ERRREL, ss1,
2 ERREST)
print*, 'ss1=',ss1
POINTS(1) = 1.d0
      CALL DQDAGP (fun, aa0,B1, NPTS, POINTS, ERRABS, ERRREL, ss2,
2 ERREST)
print*, 'ss2=',ss2
ss=ss1+ss2
end if
c 1 CALL DQDAGS (fun, A1, B1, ERRABS, ERRREL, ss, ERREST)
c print*, 'qdags ss',ss
  1 qromb_DQDAGS_2=ss
```


END

```
c FUNCTION ANC42 (A2, B2, EP, M, N, FUN)
c      IMPLICIT REAL*8 (A-H, O-Z)
c used only in computing hypergeometric 2F1 by direct integration
c external fun
c call qromb2 (fun, A2, B2, ss)
c ANC42=ss
c      END
```

```
SUBROUTINE qromb2 (func, a, b, ss)
      INTEGER JMAX, JMAXP, K, KM
      DOUBLE PRECISION a, b, func, ss, EPS
      EXTERNAL func
      PARAMETER (EPS=1.d-12, JMAX=30, JMAXP=JMAX+1, K=5, KM=K-1, aa0=0.
CU      USES polint, trapzd
      INTEGER j
      DOUBLE PRECISION dss, h (JMAXP), s (JMAXP)
      h (1)=1.d0
      do 11 j=1, JMAX
EPS2=aa0
      if (j.le.20) EPS2=1.d-12
      if (j.gt.20) EPS2=1.d-11
      if (j.ge.21) EPS2=1.d-10
      if (j.ge.22) EPS2=1.d-9
      if (j.ge.24) EPS2=1.d-8
      if (j.ge.27) EPS2=1.d-7
      if (j.ge.28) EPS2=1.d-6
      call trapzd2 (func, a, b, s (j), j)
```

```

        if (j.ge.K) then
            call polint2(h(j-KM),s(j-KM),K,0.d0,ss,dss)
c xx=abs(dss)-EPS2*abs(ss)
            if (abs(dss).le.EPS2*abs(ss)) return
        endif
        s(j+1)=s(j)
        h(j+1)=0.25d0*h(j)
11 continue
    pause 'too many steps in qromb2'
    END

```

```

SUBROUTINE trapzd2(func,a,b,s,n)
    INTEGER n
    DOUBLE PRECISION a,b,s,func
    EXTERNAL func
    INTEGER it,j
    DOUBLE PRECISION del,sum,tnm,x
    if (n.eq.1) then
        s=0.5d0*(b-a)*(func(a)+func(b))
    else
        it=2**(n-2)
        tnm=it
        del=(b-a)/tnm
        x=a+0.5d0*del
        sum=0.d0
        do 11 j=1,it
            sum=sum+func(x)
            x=x+del
11 continue
        s=0.5d0*(s+(b-a)*sum/tnm)
    endif

```

```
    return
END

SUBROUTINE polint2(xa, ya, n, x, y, dy)
    INTEGER n, NMAX
    DOUBLE PRECISION dy, x, y, xa(n), ya(n)
    PARAMETER (NMAX=10)
    INTEGER i, m, ns
    DOUBLE PRECISION den, dif, dift, ho, hp, w, c(NMAX), d(NMAX)
    ns=1
    dif=abs(x-xa(1))
    do 11 i=1, n
        dift=abs(x-xa(i))
        if (dift.lt.dif) then
            ns=i
            dif=dift
        endif
        c(i)=ya(i)
        d(i)=ya(i)
11    continue
    y=ya(ns)
    ns=ns-1
    do 13 m=1, n-1
        do 12 i=1, n-m
            ho=xa(i)-x
            hp=xa(i+m)-x
            w=c(i+1)-d(i)
            den=ho-hp
            if(den.eq.0.d0)pause 'failure in polint'
            den=w/den
            d(i)=hp*den
        enddo
    enddo
```

```
        c(i)=ho*den
12      continue
        if (2*ns.lt.n-m) then
            dy=c(ns+1)
        else
            dy=d(ns)
            ns=ns-1
        endif
        y=y+dy
13      continue
        return
        END
```

Κατάλογος σχημάτων

2.1	Διαγραμματική απεικόνιση του δυναμικού επιφανείας.	13
2.2	Σχηματική αναπαράσταση της επιφάνειας δυναμικής ενέργειας, με σταδιακές βελτιστοποιήσεις προς την θέση ισορροπίας.	13
3.1	Ηλεκτρόνιο δέσμιο από το πεδίο των δύο πυρήνων H.	18
3.2	Γραφική παράσταση της ενέργειας συναρτήσει της διαπυρηνικής αποστάσεως.	21
3.3	Ενέργειες των διηγεμένων καταστάσεων συναρτήσει της διαπυρηνικής αποστάσεως R	22
3.4	Γεωμετρία του H_2^+	23
3.5	Διαγραμματική απεικόνιση του H_2^+	24
4.1	Διαγραμματική απεικόνιση ενός πολυηλεκτρονιακού διατομικού μορίου.	34
4.2	Διαγραμματική απεικόνιση του H_2	35
4.3	Γραφική παράσταση της ενέργειας συναρτήσει της διαπυρηνικής αποστάσεως, των καταστάσεων $1^1\Sigma_g^+$ και $3^1\Sigma_u^+$, για το H_2	40
4.4	Γραφική παράσταση της ενέργειας συναρτήσει της διαπυρηνικής αποστάσεως, των καταστάσεων $1^1\Sigma_g^+$, $1^1\Sigma_u^+$, $2^1\Sigma_g^+$ και $3^1\Sigma_u^+$, για το H_2	42
A.1	Έλλειψη και υπερβολή στο επίπεδο xy στο a) το $r_A + r_B$ είναι σταθερό, ενώ στο b) το $r_A - r_B$ είναι σταθερό.	84
A.2	a) Επίμηκες και b) πεπλατυσμένο σφαιροειδές ως επιφάνειες εκ περιστροφής.	85
A.3	a) Υπερβολοειδές δύο και b) ενός φύλλου ως επιφάνειες εκ περιστροφής.	85

-
- A'.4 Επιφάνειες για σταθερές a) επιμήκεις και b) πεπλατυσμένες συντεταγμένες. Τα σφαιροειδή είναι επιφάνειες σταθερού ξ , τα υπερβολοειδή σταθερού η και τα ημι-επίπεδα σταθερού φ 86
- A'.5 Διαγραμματική απεικόνιση του συσχετισμού των σφαιρικών συντεταγμένων r, θ και των σφαιροειδών ακτίνων r_A, r_B 94

Βιβλιογραφία

- [1] N. C. Bacalis, Z. Xiong, D. Karaoulanis: *Inherent restrictions of the Hylleraas-Undheim-MacDonald higher roots, and minimization functionals at the excited states*, J. of Comp. Meth. in Sc. and Eng. 8 (2008) 277–285
- [2] N. C. Bacalis: *Neutral “parent” states of the ionic He_2^- $^4\Phi_g$* , Phys. Rev. A 55, 1996.
- [3] N. C. Bacalis, Y. Komninos and C. A. Nicolaides: *State-specific theory and method for the computation of diatomic molecules: Application to He_2^{2+} $^1\Sigma_g^+$* , Phys. Rev. A 55, 1991.
- [4] A. Szabo and N.S.Ostlund: *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, (Dover, 1996).
- [5] P. Atkins and R. Friedman: *Molecular Quantum Mechanics*, (Oxford University Press, 4th ed, 2004) καθώς και η αντίστοιχη μετάφραση στα ελληνικά από τον Α. Μαυρίδη
- [6] Α. Παπακονδύλης και Κ. Βύρας: *Ειδικά Κεφάλαια Φυσικοχημείας*, ΕΚΠΑ (2006).
- [7] J.D. Power, Philos. Trans. R. Soc. London A 274, 663 (1973), *QCPE program OEDM* (No. 233), Quantum Chemistry Program Exchange, Chemistry Department, Indiana University, Bloomington, Indiana, 47401.

- [8] L. D. Landau and L. M. Lifshitz: *Quantum Mechanics: Non-Relativistic Theory*, Volume 3, (Butterworth-Heinemann, 3rd edition, 1981).
- [9] B. Arfken , H. J. Weber and F. E. Harris: *Mathematical Methods for Physicists: A Comprehensive Guide*, (Elsevier, 7th edition, 2013).
- [10] H. Margenau and G.M. Murphy: *The Mathematics of Physics and Chemistry*, (East-West, 2nd edition, 1966).
- [11] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions*, (Dover, New York, 1972).
- [12] B. Simons: *Advanced Quantum Physics*.
- [13] Σ.Λ. Τραχανάς: *Κβαντομηχανική I*, (ΠΕΚ, 2009).
- [14] Σ.Λ. Τραχανάς: *Συνήθειες Διαφορικές Εξισώσεις*, (ΠΕΚ, 5η έκδοση, 1999).
- [15] Δ. Φούρλας: *Συνήθειες Διαφορικές Εξισώσεις*, (Πανεπιστήμιο Πατρών, 1η έκδοση, 2008).
- [16] J. D. Power, *One-electron diatomic molecules*, QCPE program No. 233.
- [17] E. A. McCullough, *The partialwave selfconsistentfield method for diatomic molecules: Computational formalism and results for small molecules*, Comput. Phys. Rep. 4, 265 (1986).
- [18] J. Meixner, R.W. Schäfke, G. Wolf: *Mathieu Functions and Spheroidal Functions and Their Mathematical Foundations* (further studies), Lecture Notes in Mathematics 837 (Springer-Verlag, Berlin, 1980).
- [19] I.A. C. Wahl, P. E. Cade, and C. C. J. Roothaan, J. Chem. Phys. 41, 2578 (1964).

- [20] Δ. Μπονάτσος: *Κβαντομηχανική II*, (Ε.Κ.Ε.Φ.Ε. Δημόκριτος, 2000).
- [21] E.W. Leaver: *Solutions to a generalized spheroidal wave equation*, J. Math. Phys. 27, 1238 (1986).
- [22] J.D. Jackson: *Classical Electrodynamics* (3rd ed.) (Wiley, New York, 1999).
- [23] E.A. Hylleraas: *Über die Elektronenterme des Wasserstoffmoleküls*, Z. Phys. 71, 739 (1931).
- [24] G. Jaffé: *Zur Theorie des Wasserstoffmoleküls*, Z. Phys. 87, 535 (1934).
- [25] W.G. Baber and H.R. Hassé, Proc. Cambridge Phil. Soc. 25, 564 (1935).
- [26] P.M. Morse and H. Feshbach: *Methods of Theoretical Physics*. (McGraw-Hill, New York, 1953).
- [27] Howard S. COHL: *Fourier, Gegenbauer and Jacobi Expansions for a Power-Law Fundamental Solution of the Polyharmonic Equation and Polyspherical Addition Theorems*, Math-ph (2013).
- [28] K. Ruedenberg, J. Chem. Phys. 19, 1459 (1951).
- [29] E. Jahnke and F. Emde, *Tables of Functions* (Dover, New York, 1945).
- [30] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes* (Cambridge University Press, New York, 1986).
- [31] H. Goldstein, C. Poole and J. Safko: *Classical Mechanics*, (Pearson, 3rd ed., 2002)
- [32] Π. Ιωάννου, Θ. Αποστολάτος : *Θεωρητική Μηχανική II*, (Leader Books, Αθήνα, 2004)

- [33] W. Schweizer: Progress in Theoretical Chemistry and Physics Vol 9 *Numerical Quantum Dynamics*, (Kluwer Academic Publishers, 2002)
- [34] L. Wei Li, X.K. Kang and M.S. Leong: *Spheroidal Wave Functions in Electromagnetic Theory*, (J.Wiley & Sons, 2002).
- [35] E. Hylleraas and B. Undheim, Z Phys 65 (1930), 759; J.K.L. McDonald, Phys Rev 43 (1933), 830.
- [36] C. Eckart, Phys Rev 36 (1930), 878.
- [37] S.S.R.R. Perumal, *Spin-Orbit and Spin-Spin coupling in the triplet state*, PhD disertation, Department of Theoretical Chemistry and Biology, KTH Royal Institute of Technology, Stockholm Sweden (2012).
- [38] O. Krieglsteiner, *Hyperfine Structure of Cs₂ Molecules in Electronically Excited States*, PhD disertation, Fakultät für Mathematik, Informatik und Physik der Universität Innsbruck (2011).