ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ & ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΔΠΜΣ “ΕΦΑΡΜΟΣΜΕΝΗ ΜΗΧΑΝΙΚΗ”

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Υπολογιστική Προσομοίωση της Μεταβατικής Συμπεριφοράς Σφαιρικών Φλογών Καύσης

Νικόλαος Μπόσινας
Μηχανικός Αεροσκαφών

Επιβλέπων: Κωνσταντίνος Ι. Σιέττος
Αναπλ. Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2014

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ & ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΠΜΣ “ΕΦΑΡΜΟΣΜΕΝΗ ΜΗΧΑΝΙΚΗ”

Υπολογιστική Προσομοίωση της Μεταβατικής Συμπεριφοράς Σφαιρικών Φλογών Καύσης

Νικόλαος Μπόσινας

Μηχανικός Αεροσκαφών

Μεταπτυχιακή Εργασία

Επιβλέπων:

Κωνσταντίνος Ι. Σιέττος
Αναπλ.Καθηγητής Ε.Μ.Π.

Τριμελής Εξεταστική Επιτροπή:

Κωνσταντίνος Ι. Σιέττος	Δημήτρης Α. Γκούσης	Ευστάθιος Ε. Θεοτόκογλου
Αναπλ.Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Στα παιδιά μου, Παναγιώτη και Κωνσταντίνα

ΠΕΡΙΛΗΨΗ

Στην εργασία αυτή γίνεται η κατάστρωση του πολύπλοκου μεγάλης κλίμακας μοντέλου μερικών διαφορικών εξισώσεων της καύσης σφαιρικών φλογών διάχυσης και προαναμεμιγμένων φλογών χρησιμοποιώντας το περιβάλλον προγραμματισμού Matlab, ενσωματώνοντας τους λεπτομερείς μηχανισμούς καύσης (κινητικές αντιδράσεων, θερμοδυναμικές καταστατικές εξισώσεις) και στη συνέχεια η μελέτη της δυναμικής συμπεριφοράς επιλύοντας τις υποκείμενες εξισώσεις στο χρόνο και χώρο.

λέξεις κλειδιά: σφαιρικές φλόγες διάχυσης, προαναμεμιγμένες φλόγες, μηχανισμός καύσης, δυναμική συμπεριφορά, ταλαντώσεις, matlab

ABSTRACT

In this thesis we set-up the complex large-scale partial differential equations model of the combustion of spherical diffusion and premixed flames by using Matlab programming environment. Inside the model we incorporate the detailed combustion mechanism (reaction transport equations, thermodynamic state equations) and then we study the transient behavior by solving the underlying discretized equations in time and space.

keywords: spherical diffusion flames, premixed flames, combustion mechanism, transient behavior, matlab

ΠΡΟΛΟΓΟΣ

Η παρούσα εργασία αποτελεί την μεταπτυχιακή μου εργασία στο Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών με τίτλο “Εφαρμοσμένη Μηχανική”.

Αφορμή της εργασίας ήταν το μάθημα “Μαθηματικές Μέθοδοι στη Μηχανική Ι”, όπου σε αυτό διδάσκεται η θεωρία διακλαδώσεων. Η θεωρία διακλαδώσεων αποτελεί κατά την άποψή μου ένα πολύ σημαντικό εργαλείο στη δυναμική ανάλυση μη γραμμικών συστημάτων, τα οποία περιγράφουν το μεγαλύτερο μέρος των δυναμικών διεργασιών που συναντώνται στη φύση, αλλά και των συστημάτων που έχει κατασκευάσει ο άνθρωπος.

Η θεωρία αυτή μας δίνει τη δυνατότητα να ανιχνεύσουμε με συστηματικό τρόπο τους διάφορους τρόπους με τους οποίους ένα δυναμικό σύστημα μπορεί να εξελιχθεί στο χρόνο και υπό ποιες προϋποθέσεις. Ένα σύστημα, που εξελίσσεται στο χρόνο, μπορεί ανάλογα με τις συνθήκες να καταλήξει σε κατάσταση ισορροπίας, να εξαφανιστεί, να μπει σε ταλαντώσεις ή να αποκτήσει χαοτική συμπεριφορά. Για παράδειγμα, η πτέρυγα ενός αεροσκάφους μπορεί υπό συγκεκριμένες συνθήκες να αρχίσει να ταλαντώνεται (fluttering), το οποίο μπορεί να οδηγήσει σε καταστροφή της πτέρυγας, ή ο ανεξέλεγκτος πολλαπλασιασμός ενός οργανισμού, που βρίσκεται σε συγκεκριμένο χώρο μπορεί υπό προϋποθέσεις να οδηγήσει τελικά στην εξάλειψή του. Είναι λοιπόν προφανές ότι η ανάλυση και καλύτερη γνώση της συμπεριφοράς αυτής είναι ζωτικής σημασίας για την ασφαλή και ελεγχόμενη εξέλιξη των συστημάτων αυτών.

Ως δυναμικό σύστημα προς μελέτη επιλέχθηκε η καύση φλογών, διότι παρά το γεγονός ότι έχει μελετηθεί εκτενώς και έχει εντοπιστεί ταλαντωτική συμπεριφορά υπό συγκεκριμένες συνθήκες (αριθμός Lewis αρκετά μικρότερος ή μεγαλύτερος της μονάδας και φλόγα κοντά στο όριο απόσβεσής της), εντούτοις δεν έχουν παρατηρηθεί οριακοί κύκλοι κατά την επίλυση του περίπλοκου μοντέλου, που προσομοιάζει τις πραγματικές συνθήκες.

Η παρούσα εργασία δεν ασχολήθηκε εκτενώς με την ανάλυση διακλαδώσεων της καύσης φλογών αλλά κυρίως με την μοντελοποίηση του περίπλοκου συστήματος της, συμπεριλαμβανομένου και του πλήρους μηχανισμού αντιδράσεων, κατά τέτοιο τρόπο ώστε το μοντέλο να μπορεί εκ των υστέρων να χρησιμοποιηθεί για συστηματική μελέτη της δυναμικής του συμπεριφοράς με θεωρία διακλαδώσεων. Ένα πολύ διαδεδομένο εργαλείο συστηματικής ανάλυσης διακλαδώσεων είναι το Matcont, το οποίο είναι πακέτο του Matlab. Συνεπώς, η παρούσα εργασία επιδίωξε να μοντελοποιήσει σε μορφή Matlab τον πλήρη μηχανισμό καύσεων φλογών, προκειμένου στο μέλλον να μπορεί να χρησιμοποιηθεί από το Matcont.

Η μοντελοποίηση του πλήρους μηχανισμού της καύσης φλογών σε Matlab αποτέλεσε μεγάλη πρόκληση, καθόσον περιέχει πάρα πολλές εξισώσεις και μαθηματικούς τύπους

υπολογισμού κινητικών και θερμοδυναμικών ιδιοτήτων. Ως σημείο αναφοράς χρησιμοποιήθηκε το λογισμικό CHEMKIN, το οποίο αναπτύχθηκε από τη NASA σε FORTRAN και αποτελεί ένα από τα πιο διαδεδομένα εργαλεία μοντελοποίησης της καύσης φλογών. Το CHEMKIN βοήθησε ώστε να γίνει αντιπαραβολή αποτελεσμάτων με τον κώδικα του Matlab και να διορθωθούν ατέλειες και σφάλματα.

Το τελικό μοντέλο που αναπτύχθηκε σε περιβάλλον Matlab είναι σε θέση να μοντελοποιήσει φλόγες διάχυσης και προαναμεμιγμένες φλόγες, υπό ποικίλες συνθήκες και να εξετάσει την κατάσταση ισορροπίας, αλλά και τη δυναμική συμπεριφορά του, χρησιμοποιώντας λεπτομερή μηχανισμό αντιδράσεων. Επίσης, είναι φτιαγμένο κατά τέτοιο τρόπο, ώστε να μπορεί εύκολα να εισαχθεί στο Matcont και να γίνει συστηματική ανάλυση της δυναμικής του συμπεριφοράς με θεωρία διακλαδώσεων.

Στο μέλλον θα επιδιωχθεί να γίνει περαιτέρω βελτιστοποίηση του κώδικα, ιδίως σε ότι έχει να κάνει με την ταχύτητα υπολογισμών, αλλά και η προσθήκη περισσότερων επιλογών και δυνατοτήτων, όπως για παράδειγμα η δυνατότητα αριθμητικής επίλυσης με μέθοδο Arc-Length, εκτός από Newton, η επίλυση σε 2 ή τρεις χωρικές διαστάσεις αλλά και η ενσωμάτωση περισσότερων επιλογών όσον αφορά το ίδιο το σύστημα καύσης (συνοριακές συνθήκες, χρήση πολυσυστατικής-multicomponent μεθοδολογίας υπολογισμού των κινητικών και θερμοδυναμικών ιδιοτήτων εκτός από mixture-average).

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον καθηγητή μου, κ. Κωνσταντίνο Σιέττο, για την αμέριστη βοήθεια που μου παρείχε κατά τη διάρκεια εκπόνησης της παρούσας εργασίας, αλλά και την υπομονή που επέδειξε λαμβάνοντας υπόψη τις συνθήκες υπό τις οποίες έγινε. Επίσης θα ήθελα να ευχαριστήσω την ομάδα του κ. Σιέττου και του κ. Γκούση, η οποία με βοήθησε και μου έδωσε πολύτιμες πληροφορίες όταν έπρεπε. Θα ήταν χαρά και τιμή για μένα να συνεργαστώ μαζί τους και στη συνέχεια.

Επίσης, θα ήθελα να ευχαριστήσω τους καθηγητές κ. Δημήτρη Γκούση και κ. Ευστάθιο Θεοτόκογλου για την τιμή που μου έκαναν να συμμετάσχουν στην τριμελή επιτροπή.

Θα ήθελα επίσης να ευχαριστήσω την πρώην σύζυγό μου για την υπομονή και υποστήριξή της το διάστημα των τελευταίων τριών μηνών που ασχολούμουν εντατικά με την παρούσα εργασία.

Επίσης θα ήθελα να ευχαριστήσω τη μητέρα μου, την αδερφή μου και τον αδερφό μου για την υποστήριξή και την αγάπη τους, η οποία ήταν πάντα εκεί όταν τη ζήτησα.

Τέλος, θα ήθελα να ευχαριστήσω από τα βάθη της καρδιάς μου τα παιδιά μου, Παναγιώτη και Κωνσταντίνα, για την έμπνευση και τη δύναμη που μου δίνει το χαμόγελό τους καθημερινά. Η παρούσα εργασία δε θα μπορούσε παρά να αφιερώνεται σε αυτούς.

ΠΕΡΙΕΧΟΜΕΝΑ

1 Εισαγωγή 1

1.1 Φλόγες Καύσης 2

1.2 Δυναμική Ανάλυση 3

2 Μεθοδολογία 5

3 Θεωρητική Περιγραφή του Προβλήματος 8

3.1 Βασικές Εξισώσεις 8

3.2 Μηχανισμοί Αντίδρασης 10

3.2.1 Κινητικές Ιδιότητες 10

3.2.2 Θερμοδυναμικές Ιδιότητες 17

4 Αριθμητική Περιγραφή του Προβλήματος 21

4.1 Συνοριακές Συνθήκες 21

4.2 Χωρική Διακριτοποίηση 22

4.3 Κατάσταση Ισορροπίας 26

4.4 Τροποποιημένη Μέθοδος Newton Με Απόσβεση 28

4.5 Μεταβατική Κατάσταση 31

4.5.1 Επίλυση Μεταβατικής Κατάστασης με PDE 31

4.5.2 Επίλυση Μεταβατικής Κατάστασης με ODEs 34

5 Ανάπτυξη Αλγόριθμων στο Matlab 35

5.1 Ρουτίνες Ανάγνωσης Χημικών Δεδομένων 36

5.2 Ρουτίνες Πολυωνυμικής Προσέγγισης 36

5.3 Ρουτίνες Υπολογισμού Θερμοδυναμικών και Κινητικών Ιδιοτήτων 38

5.4 Ρουτίνες Επίλυσης 39

5.5 Διαθέσιμες Επιλογές του Αλγορίθμου 41

5.6 Ροϊκά Διαγράμματα Ρουτινών 41
5.7 Επιλογές Συστήματος Εξισώσεων 42
5.8 Πιστοποίηση Αλγόριθμων 43
5.8.1 Πιστοποίηση Πρώτου Επιπέδου 44
5.8.2 Πιστοποίηση Δεύτερου Επιπέδου 45
6 Επίλυση Προβλημάτων στο Matlab 49
6.1 Πρόβλημα 1 – Προαναμεμιγμένες Φλόγες Μεθανίου-Αέρα 49
6.2 Πρόβλημα 2 – Φλόγες Διάχυσης Υδρογόνου 52
7 Συμπεράσματα 58
7.1 Μελλοντική Έρευνα 58
ΠΑΡΑΡΤΗΜΑ Α 60
ΠΑΡΑΡΤΗΜΑ Β 63
ΠΑΡΑΡΤΗΜΑ Γ 65
ΠΑΡΑΡΤΗΜΑ Δ 68
ΠΑΡΑΡΤΗΜΑ Ε 70
ΠΑΡΑΡΤΗΜΑ ΣΤ 75
ΒΙΒΛΙΟΓΡΑΦΙΑ 88

ΠΙΝΑΚΕΣ

Πίνακας 1 - Ρουτίνες Ανάγνωσης Χημικών Δεδομένων 36
Πίνακας 2 - Ρουτίνες Πολυωνυμικής Προσέγγισης 37
Πίνακας 3 - Ρουτίνες Υπολογισμού Θερμοδυναμικών και Κινητικών Ιδιοτήτων 38
Πίνακας 4 - Ρουτίνες Επίλυσης 39
Πίνακας 5 – Επιλογές Συστήματος Μοντελοποιημένων Εξισώσεων 42
Πίνακας 6 – Μηχανισμός Καύσης Προαναμεμιγμένης Φλόγας H_2-O_2 44
Πίνακας 7 – Αρχική Λύση PREMIX Προβλήματος Καύσης H_2-O_2 46

Πίνακας 8 – Τελική Λύση PREMIX Προβλήματος Καύσης H2-O2 46
Πίνακας 9 – Αρχείο Προβλήματος Καύσης Προαναμεμιγμένης Φλόγας H2-O2 47
Πίνακας 10 - Αρχική Λύση Matlab Προβλήματος Καύσης H2-O2 47
Πίνακας 11 - Τελική Λύση Matlab Προβλήματος Καύσης H2-O2 48
Πίνακας 12 – Περιγραφή Προβλήματος Φλογών Διάχυσης 52
Πίνακας 13 – Αρχείο Παραμέτρων Προβλήματος Φλογών Διάχυσης Υδρογόνου (α) 53
Πίνακας 14 – Αρχείο Παραμέτρων Προβλήματος Φλογών Διάχυσης Υδρογόνου (β) 55
Πίνακας 15 – Διαθέσιμες Επιλογές Αλγορίθμου Matlab 60
Πίνακας 16 – Πολυωνυμικοί Συντελεστές Κινητικών Ιδιοτήτων Καύσης Φλογών H_2-O_2 65
Πίνακας 17 – Κώδικας TestChemkinInput2.m 68
Πίνακας 18 – Μηχανισμός Αντίδρασης Φλόγας CH_4-Αέρα 70
Πίνακας 19 – Αρχεία Εισόδου Προβλήματος Καύσης Φλόγας CH4-Αέρα 72
Πίνακας 20 – Τελική Λύση σε Κατάσταση Ισορροπίας Προβλήματος Καύσης Φλόγας CH4-Αέρα 73
Πίνακας 21 – Κώδικας CalculateBinaryDiffusionCoefficientFits.m 75
Πίνακας 22 – Κώδικας CalculateThermalDiffusionRatioFits.m 77
Πίνακας 23 – Κώδικας CalculateThermalConductivityandViscosityFits.m 78
Πίνακας 24 – Κώδικας CalculateBinaryDiffusionCoefficientD12.m 80
Πίνακας 25 – Κώδικας CalculateOmega12.m 81
Πίνακας 26 – Κώδικας CalculateOmegaXX.m 84
Πίνακας 27 – Κώδικας CalculateRotationalCollisionNumbers.m 85
Πίνακας 28 – Κώδικας GetKfi.m 85
Πίνακας 29 – Κώδικας GetKri.m 86
Πίνακας 30 – Κώδικας Getqri.m 87

ΔΙΑΓΡΑΜΜΑΤΑ

Διάγραμμα 1 – Ροϊκό διάγραμμα αλγορίθμου επίλυσης κατάστασης ισορροπίας 42
Διάγραμμα 2 – Ροϊκό Διάγραμμα Αλγορίθμου Newton 63
Διάγραμμα 3 – Ροϊκό Διάγραμμα Αλγόριθμου Χρονικού Βηματισμού 64

ΕΙΚΟΝΕΣ

Εικόνα 1 – Πρόβλημα 1: Μεταβολή Ρυθμού Ροής Μάζας στο Χρόνο 50
Εικόνα 2 – Πρόβλημα 1: Μεταβολή Μέγιστης Θερμοκρασίας στο Χρόνο 50
Εικόνα 3 – Πρόβλημα 1: Μεταβολή Ενθαλπίας Μίγματος στο Χώρο 51
Εικόνα 4 – Πρόβλημα 1: Μεταβολή Μέγιστης Ενθαλπίας Μίγματος στο Χρόνο 51
Εικόνα 5 – Πρόβλημα 2(α): Μεταβολή Μέγιστης Θερμοκρασίας στο Χρόνο 54
Εικόνα 6 – Πρόβλημα 2(α): Μεταβολή Θερμοκρασίας στο Χώρο 55
Εικόνα 7 – Πρόβλημα 2(β): Μεταβολή Μέγιστης Θερμοκρασίας στο Χρόνο 56
Εικόνα 8 – Πρόβλημα 2(β): Μεταβολή Ενθαλπίας στο Χώρο 57
Εικόνα 9 – Πρόβλημα 2(β): Μεταβολή Θερμοκρασίας στο Χώρο..... 57

1 Εισαγωγή

Δυναμικό είναι κάθε σύστημα, φυσικό, χημικό, βιολογικό, οικολογικό, οικονομικό ή ακόμα και κοινωνικό, που εξελίσσεται στο χρόνο και περιλαμβάνει ένα σύνολο πιθανών καταστάσεων καθώς και ένα νόμο εξέλιξης στο χρόνο που μας βοηθά να προβλέπουμε παρελθούσες ή μελλοντικές καταστάσεις (Kuznetzov, 1998).

Η μελέτη δυναμικών συστημάτων είναι σημαντική για την κατανόηση της συμπεριφοράς τους και για την πρόβλεψη της εξέλιξής τους. Η απόκριση ενός δυναμικού συστήματος μπορεί να επηρεαστεί από πολλές παραμέτρους, που οδηγούν το σύστημα είτε σε ευστάθεια, είτε σε περιοδικές ταλαντώσεις, είτε σε χαοτικές συμπεριφορές είτε ακόμα και στην καταστροφή. Οι ποιοτικές αλλαγές στη δυναμική ενός συστήματος ονομάζονται **διακλαδώσεις** και οι τιμές των παραμέτρων για τις οποίες συμβαίνουν ονομάζονται **σημεία διακλάδωσης** (Strogatz, 1994).

Αν και η φυσική παρατήρηση συστημάτων ή φυσικών/μηχανολογικών διεργασιών αποτελεί τον πιο αξιόπιστο και ρεαλιστικό τρόπο μελέτης της απόκρισής τους, δεν είναι πάντα δυνατή ή αποτελεσματική, διότι μερικές φορές απαιτεί εξοπλισμό και μετρητικές συσκευές υψηλού κόστους, ή επιβάλλει κινδύνους ασφάλειας λόγω παραγόντων αβεβαιότητας ενώ παρέχει μικρή ευελιξία στην μεταβολή των παραμέτρων.

Ένα δυναμικό σύστημα που εξελίσσεται σε συνεχή χρόνο περιγράφεται συνήθως από ένα σύστημα διαφορικών εξισώσεων. Ανάλογα με τη φύση και πολυπλοκότητα του, το σύστημα διαφορικών εξισώσεων μπορεί να είναι συνήθεις ή μερικές, γραμμικές ή μη γραμμικές. Οι φυσικές / μηχανολογικές διεργασίες περιγράφονται συνήθως από ένα σύστημα μη γραμμικών μερικών διαφορικών εξισώσεων – non-linear partial differential equations (PDE).

Η μαθηματική περιγραφή ενός συστήματος επιτρέπει τη μελέτη της συμπεριφοράς του αναλυτικά. Όμως κάτι τέτοιο είναι εφικτό μόνο σε απλές περιπτώσεις, στις οποίες το σύστημα εξισώσεων μπορεί να λυθεί. Οι φυσικές / μηχανολογικές διεργασίες είναι συνήθως πολύ περίπλοκες για να επιλυθούν αναλυτικά και γι αυτό το λόγο μελετώνται απλουστευμένα μοντέλα, τα οποία δίνουν μία γενική ιδέα της συμπεριφοράς τους.

Η εξέλιξη των ηλεκτρονικών υπολογιστών και της υπολογιστικής μηχανικής έχει καταστήσει εφικτή τη μελέτη περίπλοκων διεργασιών όχι μέσω φυσικής αναπαράστασής τους ή εύρεσης αναλυτικών λύσεων των διαφορικών τους εξισώσεων, αλλά μέσω προσομοίωσης. Είναι δυνατή η μοντελοποίηση και η προσομοίωση της συμπεριφοράς ενός συστήματος με τη χρήση κατάλληλου λογισμικού και ηλεκτρονικού υπολογιστή. Το μεγάλο πλεονέκτημα της προσομοίωσης είναι ότι μπορεί να μελετήσει περίπλοκα συστήματα, αποτελούμενα από εκατοντάδες διαφορικές εξισώσεις, με περίπλοκες γεωμετρίες και συνοριακές συνθήκες. Επιπλέον, από τη στιγμή που ένα σύστημα μοντελοποιηθεί, είναι εύκολη η μεταβολή των παραμέτρων του και η εξομοίωσή του υπό πολλές διαφορετικές σύνθήκες.

1.1 Φλόγες Καύσης

Η παρούσα εργασία μελετά τη δυναμική εξέλιξη και συμπεριφορά σφαιρικών φλογών διάχυσης αλλά και προαναμεμιγμένων φλογών.

Ως **φλόγα διάχυσης** μπορεί να θεωρηθεί οποιαδήποτε φλόγα κατά την οποία το καύσιμο αρχικά βρίσκεται ξεχωριστά από το οξειδωτικό (Williams, 1985). Κατά μία πιο αυστηρή έννοια, η φλόγα διάχυσης είναι ημισταθερή (quasisteady), σχεδόν ισοβαρής και το μεγαλύτερο μέρος της αντίδρασης συμβαίνει σε μία στενή περιοχή, που μπορεί να προσεγγιστεί ως μία επιφάνεια. Παραδείγματα φλογών διάχυσης είναι ένας δίσκος με λάδι που καίγεται στον αέρα, ένα φύλλο αλουμινίου που καίγεται σε μία υπερηχητική ροή αέρα, ένα κερί, μία πυρκαγιά δάσους ή μία σταγόνα καυσίμου που καίγεται σε οξυγόνο σε έναν πυραυλοκινητήρα.

Ως **προαναμεμιγμένη φλόγα** θεωρείται οποιαδήποτε φλόγα κατά την οποία το καύσιμο έχει προαναμιχθεί με το οξειδωτικό πριν εισέλθει στο θάλαμο καύσης (El-Mahallawy & E-Din Habik, 2002).

Οι σφαιρικές φλόγες καύσης είναι μονοδιάστατες λόγω της σφαιρικής τους συμμετρίας (Williams, 1985, p. 56). Συνεπώς, αποτελούν ένα χρήσιμο εργαλείο μελέτης προβλημάτων καύσης. Τα προβλήματα αυτά έχουν μελετηθεί αριθμητικά και πειραματικά σε διαφορετικές συνθήκες και περιβάλλοντα χρησιμοποιώντας ποικίλες μεθόδους και προσεγγίσεις (Chao, et al., 1993; Mills & Matalon, 1998; Chao & Law, 1993).

Τα προβλήματα σφαιρικών φλογών καύσης μπορούν να περιγραφούν μαθηματικά χρησιμοποιώντας ένα σύστημα μερικών διαφορικών εξισώσεων (PDE), το οποίο προκύπτει από τους νόμους διατήρησης της μάζας, των συστατικών, της ενέργειας και ορμής, όπου οι μεταβλητές έχουν μερικά διαφορικά ως προς το μονοδιάστατο χώρο (ακτίνα σφαίρας) και το χρόνο.

1.2 Δυναμική Ανάλυση

Πολλές μελέτες φλογών καύσης εξετάζουν το πρόβλημα της κατάστασης ισορροπίας, όπου οι μεταβλητές του συστήματος – θερμοκρασίες, κλάσματα μάζας των συστατικών, ρυθμοί ροής μάζας – έχουν σταθεροποιηθεί έπειτα από κάποιο χρόνο και δεν μεταβάλλονται σημαντικά με το χρόνο από εκεί και πέρα. Όμως, πριν επέλθει κατάσταση ισορροπίας προηγείται μία περίοδος μεταβατικής συμπεριφοράς. Η μεταβατική κατάσταση υπάρχει από την ανάφλεξη μέχρι το σύστημα να σταθεροποιηθεί. Επίσης, το σύστημα μπορεί να εισέλθει σε μεταβατική κατάσταση σκόπιμα επιβάλλοντας μία μικρή μεταβολή σε μία από τις παραμέτρους του (π.χ. ρυθμό ροής μάζας) ενώ είναι σε ισορροπία. Η μεταβολή αυτή μπορεί να οδηγήσει σε διαφορετικές τελικές καταστάσεις του συστήματος. Το σύστημα μπορεί να επανέλθει στην κατάσταση ισορροπίας γρήγορα ή αργά η μπορεί να παρουσιάσει δυναμικά φαινόμενα (ταλαντώσεις φλόγας) ή μπορεί να καταλήξει σε εξαφάνιση της φλόγας.

Η συμπεριφορά του συστήματος κατόπιν μεταβολής των παραμέτρων του από την κατάσταση ισορροπίας εξαρτάται από πολλούς παράγοντες, όπως:

- Από την κατάσταση των χημικών παραμέτρων του συστήματος, για παράδειγμα μία μεταβολή στο μίγμα καυσίμου και ενώ ο ρυθμός ροής μάζας είναι μικρός είναι πιο πιθανό να οδηγήσει σε εξαφάνιση της φλόγας από ότι αν ο ρυθμός ροής μάζας ήταν υψηλότερος.
- Από την παράμετρο που μεταβάλλεται, για παράδειγμα η μεταβολή του μίγματος καυσίμου (π.χ. χρησιμοποιώντας He ως αδρανές αέριο αντί για N_2) έχει διαφορετικό αποτέλεσμα από ότι μεταβολή του οξειδωτικού ή του ρυθμού ροής μάζας.

- Από το μέγεθος της μεταβολής, για παράδειγμα είναι πιο πιθανό μικρές μεταβολές να οδηγήσουν σε επιστροφή του συστήματος σε κατάσταση ισορροπίας από ότι μεγάλες.

Η τελική κατάσταση του συστήματος σε σχέση με τους παραπάνω παράγοντες μπορεί να αναπαρασταθεί γραφικά χρησιμοποιώντας **διαγράμματα διακλάδωσης**.

Παλαιότερες εργασίες με χρήση αριθμητικών μεθόδων για τη μελέτη της μεταβατικής συμπεριφοράς σφαιρικών φλογών διάχυσης αποκάλυψαν ότι υπό συγκεκριμένες συνθήκες η φλόγα ταλαντώνεται για ένα χρονικό διάστημα πριν επιστρέψει σε κατάσταση ισορροπίας ή εξαφανιστεί (Dietrich, et al., 1997; Nayagam & Williams; 1998, Furi, et al., 2000). Αν και αυτό υπονοεί ότι μπορεί να υπάρχουν συνθήκες υπό τις οποίες η φλόγα εισέρχεται σε μία κατάσταση συνεχούς ταλάντωσης, που αναπαριστάται με **οριακούς κύκλους** στο διάγραμμα διακλάδωσης, τέτοιου είδους συμπεριφορά δεν έχει παρατηρηθεί είτε αριθμητικά είτε πειραματικά (Christiansen, et al., 2003).

Η παρούσα εργασία αφορά αριθμητική μελέτη της δυναμικής συμπεριφοράς σφαιρικών φλογών διάχυσης και προαναμεμιγμένων φλογών χρησιμοποιώντας το περιβάλλον προγραμματισμού Matlab για την αριθμητική επίλυση των υποκείμενων εξισώσεων στο χρόνο, με σκοπό μελλοντικά να συνδυαστεί με τις δυνατότητες του Matcont για την ανάλυση διακλάδωσης. Το Matcont είναι μία βιβλιοθήκη του Matlab (Dhooge, et al., 2003) που μπορεί να εντοπίσει με συστηματικό τρόπο διακλαδώσεις ενός δυναμικού συστήματος, που περιγράφεται από ένα σετ συνήθων διαφορικών εξισώσεων – ordinary differential equations (ODE). Το πλεονέκτημα του Matcont είναι ότι μπορεί να εντοπίσει αυτόματα σημεία διακλάδωσης ενός συστήματος ξεκινώντας από μία αρχική κατάσταση, μεταβάλλοντας τις παραμέτρους του. Περιέχει μαθηματικούς κανόνες για τον εντοπισμό σχεδόν όλων των τύπων σημείων διακλάδωσης, καθώς και εργαλεία για την αναπαράστασή τους γραφικά. Μπορεί να επιτύχει συνέχιση των αλγορίθμων χρονικού βήματος (time stepper algorithms) σε όλο το εύρος των παραμέτρων, ακόμα και σε περιοχές με σημεία ιδιομορφίας.

Επιδίωξη της παρούσας εργασίας είναι καταρχήν η κατάστρωση του πολύπλοκου μεγάλης κλίμακας μοντέλου μερικών διαφορικών εξισώσεων ενσωματώνοντας τους λεπτομερείς μηχανισμούς καύσης (κινητικές αντιδράσεων, θερμοδυναμικές καταστατικές εξισώσεις) και

εν συνεχεία η αριθμητική επίλυση του, τόσο στο χρόνο όσο και στην κατάστρωση του πλαισίου για την ανίχνευση κλάδων λύσεων με την χρήση του MATCONT.

2 Μεθοδολογία

Στα πλαίσια της παρούσας εργασίας πραγματοποιήθηκαν οι παρακάτω ενέργειες:

1. Καθορισμός του μαθηματικού μοντέλου του συστήματος (σύστημα μερικών διαφορικών εξισώσεων)
2. Καθορισμός του θερμοδυναμικού μοντέλου των συστατικών
3. Καθορισμός του κινητικού μοντέλου των συστατικών
4. Εισαγωγή του αλγορίθμου επίλυσης και των μοντέλων χημείας στο Matlab
   4.1. Ανάπτυξη ρουτινών του Matlab για τον υπολογισμό των θερμοδυναμικών ιδιοτήτων των συστατικών
   4.2. Ανάπτυξη ρουτινών του Matlab για τον υπολογισμό των κινητικών ιδιοτήτων των συστατικών
   4.3. Ανάπτυξη ρουτινών του Matlab για την ανάγνωση τυπικών αρχείων που περιέχουν θερμοδυναμικά δεδομένα, κινητικά δεδομένα και αντιδράσεις
   4.4. Επιβεβαίωση των ρουτινών μοντέλων χημείας του Matlab
   4.5. Ανάπτυξη αλγορίθμου του Matlab για την επίλυση προβλημάτων σε κατάσταση ισορροπίας και μεταβατική κατάσταση
5. Υπολογισμός της λύσης ισορροπίας και βελτιστοποίηση του χωρικού πλέγματος
   5.1. Εξαγωγή των χημικών πληροφοριών του συστήματος από αρχεία εισόδου (θερμοδυναμικές και κινητικές ιδιότητες των συστατικών, δεδομένα αντιδράσεων)
   5.2. Καθορισμός του αρχικού πλέγματος
   5.3. Σχηματισμός του PDE ως ένα σύστημα εξισώσεων διακριτοποιημένων στο χώρο
   5.4. Ενσωμάτωση συνοριακών συνθηκών
   5.5. Δημιουργία μίας αρχικής λύσης εκκίνησης για το σύστημα
   5.6. Χρήση επαναληπτικού αλγορίθμου (π.χ. Newton-Raphson) για την επίλυση του προβλήματος ισορροπίας

5.7. Δυναμική αναθεώρηση του χωρικού πλέγματος σε κάθε βήμα για βέλτιστη προσαρμογή
5.8. Υπολογισμός της λύσης ισορροπίας
5.9. Επιβεβαίωση του αλγόριθμου επίλυσης του Matlab

6. Μετατροπή (προκαταρκτική) του PDE σε ένα σύστημα συνήθων διαφορικών εξισώσεων (ODE), που θα απαιτηθούν μελλοντικά από το Matcont
7. Είσοδος του προβλήματος στο Matlab
8. Ανάλυση σταθερής και μεταβατικής κατάστασης με χρήση του Matlab
9. Παραγωγή αποτελεσμάτων

Τα πρώτα τρία βήματα αφορούν στην αναλυτική περιγραφή του προβλήματος και δεν χρειάζονται περαιτέρω επεξήγηση.

Το τέταρτο βήμα απαιτεί τη δημιουργία ενός αλγορίθμου επίλυσης, καθώς επίσης και τον υπολογισμό των χημικών ιδιοτήτων στο Matlab. Στην πραγματικότητα υπάρχουν αρκετοί αλγόριθμοι επίλυσης που εξειδικεύονται στην επίλυση προβλημάτων φλογών καύσης, όπως το PREMIX (Kee, et al., 1998) και το RUN-1DL (Rogg, 1998). Αυτά είναι προγράμματα λογισμικού γραμμένα σε FORTRAN. Το πλεονέκτημά τους είναι ότι έχουν δοκιμαστεί διεξοδικά για πολλά χρόνια και είναι συμβατά με τους πίνακες θερμοδυναμικών ιδιοτήτων της NASA (McBride, et al., 1993) και επίσης με τα πακέτα λογισμικού CHEMKIN και TRANSPORT. Το CHEMKIN είναι ένα λογισμικόσε FORTRAN που αναπτύχθηκε από τη NASA (Kee, et al., 2000)και υπολογίζει της θερμοδυναμικές ιδιότητες των συστατικών με βάση το μηχανισμό αντίδρασης και κάποιες βασικές παραμέτρους του προβλήματος καύσης. Είναι το πιο διαδεδομένο εργαλείο για τη συγκεκριμένη εργασία. Το TRANSPORT είναι μία επέκταση του CHEMKIN που υπολογίζει της κινητικές ιδιότητες των συστατικών, όταν αυτές επιλεχθεί να ληφθούν υπόψη για την επίλυση ενός προβλήματος.

Στην παρούσα εργασία η απαίτηση για χρήση του Matlab για την επίλυση του συστήματος είναι μεγάλη, κάτι το οποίο επιβάλλεται από το γεγονός ότι το Matcont είναι γραμμένο σε Matlab. Κατά τη διάρκεια επίλυσης ενός προβλήματος στοMatlab, οι θερμοδυναμικές και κινητικές ιδιότητες των συστατικών πρέπει να επανυπολογίζονται. Αν χρησιμοποιηθεί το CHEMKIN, τότε θα πρέπει να αναπτυχθεί συνδετικός κώδικας λογισμικού μεταξύ του Matlab και του CHEMKIN ώστε να καταστεί δυνατή η αλληλεπίδραση τους. Στην ουσία αυτό

σημαίνει ότι το CHEMKIN θα πρέπει να εκτελείται κάθε φορά που απαιτείται επανυπολογισμός των θερμοδυναμικών και/ή κινητικών ιδιοτήτων των συστατικών. Αν και από τεχνικής άποψης κάτι τέτοιο είναι εφικτό, εντούτοις ο χρόνος που θα απαιτούνταν για τη μετατροπή των δεδομένων από το Matlab σε μορφή συμβατή με το CHEMKIN, η κλήση εξωτερικής εφαρμογής (δηλαδή του CHEMKIN) από το Matlab, η λήψη των αποτελεσμάτων και η μετατροπή τους πάλι πίσω σε μορφή συμβατή με το Matlabθα ήταν πολύ μεγάλος και θα προκαλούσε καθυστερήσεις στην επίλυση. Επιπλέον, μία τέτοια λύση προσφέρει λιγότερο έλεγχο και ευελιξία από ότι να χρησιμοποιηθεί κώδικας Matlab για την ίδια εργασία.

Συνεπώς, μία από τις ενέργειες της παρούσας εργασίας είναι η ανάπτυξη ρουτινών του Matlab για τον υπολογισμό των θερμοδυναμικών και κινητικών ιδιοτήτων των συστατικών, με τον ίδιο τρόπο που το κάνει και το CHEMKIN. Μαζί με αυτά, θα πρέπει επίσης να αναπτυχθούν ρουτίνες για την ανάγνωση δεδομένων από τυποποιημένους πίνακες θερμοδυναμικών και κινητικών ιδιοτήτων, καθώς επίσης και ρουτίνες ανάγνωσης του μηχανισμού αντίδρασης από αρχεία που ακολουθούν τις προδιαγραφές και τις μορφές εισόδου του CHEMKIN, το οποίο αποτελεί το πρότυπο της βιομηχανίας.

Λόγω της πολυπλοκότητας τους, οι ρουτίνες του Matlab, που θα αναπτυχθούν, θα επιβεβαιωθούν συγκρίνοντας τα αποτελέσματά τους με αυτά του CHEMKIN.

Το πέμπτο βήμα απαιτεί την επίλυση του προβλήματος ισορροπίας, πριν τη χρήση του Matcont. Αν και κάτι τέτοιο δεν απαιτείται για να λειτουργήσει το Matcont (μπορεί να υπολογίσει και το ίδιο την κατάσταση ισορροπίας ενός συστήματος), <u>είναι πολύ βασικό</u> διότι μέσω της διαδικασίας αυτής θα καθοριστεί το βέλτιστο χωρικό πλέγμα. Ιδιαίτερα στις φλόγες διάχυσης για τη μοντελοποίηση όλης της αντίδρασης απαιτείται μεγάλη χωρική περιοχή, π.χ. 500cm, όμως η περιοχή της καύσης, όπου εκεί υπάρχουν μεγάλες κλίσεις των χημικών ιδιοτήτων των συστατικών, είναι μικρότερη από 1cm σε πάχος και η θέση της ποικίλει ανάλογα με τις παραμέτρους του προβλήματος.

Για την αριθμητική επίλυση του συστήματος των μερικών διαφορικών εξισώσεων, χρειάζεται η διακριτοποίηση του αρχικού PDE στο χώρο, Στην ουσία, ο τελικός αριθμός των εξισώσεων του ODE είναι ίσος με τον αριθμό των εξισώσεων του PDE πολλαπλασιασμένο με τον αριθμό των κόμβων του χωρικού πλέγματος.

Είναι λοιπόν αναγκαία η μείωση του μεγέθους του πλέγματος όσο το δυνατόν περισσότερο, χωρίς όμως να μειωθεί η ακρίβεια των υπολογισμών. Για να γίνει αυτό υιοθετείται ανομοιογενές πλέγμα, το οποίο είναι πυκνό σε περιοχές μεγάλων μεταβολών των μεταβλητών του συστήματος (π.χ. στην περιοχή της καύσης) και αραιό σε περιοχές μικρών μεταβολών (π.χ. κοντά στο εξωτερικό άκρο).

Στο έβδομο βήμα το σύστημα, που μετατράπηκε στο σύστημα συνήθων διαφορικών εξισώσεων (ODE), θα εισαχθεί στο Matlab και στο Matcont. Το Matlab δέχεται ODEs της μορφής:

$$\frac{\partial \phi}{\partial t} = F(\phi, a)$$

όπου ϕ είναι το διάνυσμα των άγνωστων μεταβλητών του συστήματος, a είναι το διάνυσμα των παραμέτρων και $F(\phi, a)$ είναι το διάνυσμα των εξισώσεων, που συνιστούν το ODE.

3 Θεωρητική Περιγραφή του Προβλήματος

3.1 Βασικές Εξισώσεις

Παρακάτω περιγράφονται οι εξισώσεις διατήρησης της μάζας, των συστατικών και της ενέργειας, οι οποίες συνιστούν το σύστημα εξισώσεων προς επίλυση (Christiansen, et al., 2003), :

Διατήρηση της μάζας:

$$A\frac{\partial \rho}{\partial t} + \frac{\partial \dot{m}}{\partial r} = 0 \tag{1}$$

όπου $A = 4\pi r^2$ είναι η επιφάνεια της σφαιρικής φλόγας με ακτίνα r, $\dot{m}$ είναι ο ρυθμός ροής μάζας του καυσίμου, ρ είναι η πυκνότητα του μίγματος και υ είναι η ταχύτητα της αντίδρασης.

Διατήρηση των συστατικών:

$$\rho A \frac{\partial Y_k}{\partial t} + \dot{m} \frac{\partial Y_k}{\partial r} + \frac{\partial}{\partial r}(\rho A Y_k V_k) - A\omega_k W_k = 0 \tag{2}$$

όπου Y_k είναι το κλάσμα μάζας του συστατικού k, V_k η ταχύτητα διάχυσης, ω_k ο μοριακός ρυθμός παραγωγής ανά μονάδα όγκου και W_k το μοριακό βάρος.

Διατήρηση της ενέργειας:

$$\rho A \frac{\partial T}{\partial t} + \dot{m} \frac{\partial T}{\partial r} - \frac{1}{c_p} \frac{\partial}{\partial r}\left(A\lambda \frac{\partial T}{\partial r}\right) + \frac{1}{c_p} \frac{\partial T}{\partial r} \sum_{k=1}^{K} (\rho A Y_k V_k c_{pk}) + \frac{1}{c_p} \sum_{k=1}^{K} (A\omega_k h_k W_k) + A \frac{q}{c_p} = 0 \tag{3}$$

όπου c_p είναι η ειδική θερμότητα υπό σταθερή πίεση, c_{pk} είναι η ειδική θερμότητα υπό σταθερή πίεση του συστατικού k, λ η θερμική αγωγιμότητα και q θερμική απώλεια ακτινοβολίας ανά μονάδα όγκου.

Ο ρυθμός ροής μάζας δίνεται από την παρακάτω σχέση:

$$\dot{m} = \rho \upsilon A \tag{4}$$

Είναι επίσης γνωστό από την καταστατική εξίσωση των αερίων ότι:

$$\rho = \frac{p\overline{W}}{RT} \tag{5}$$

όπου p είναι η πίεση, $\overline{W}$ το μέσο μοριακό βάρος του μίγματος και R η παγκόσμια σταθερά των αερίων.

Θεωρώντας ότι μίγμα αποτελείται από K συστατικά, οι εξισώσεις (2) και (3) αποτελούν ένα σύστημα μερικών διαφορικών εξισώσεων με K+2 αγνώστους, δηλαδή $\dot{m}$, T και Y_k. Αν και η πυκνότητα ρ του μίγματος έχει παράγωγο ως προς το χρόνο στην εξίσωση διατήρησης της μάζας (1), δεν είναι ανεξάρτητη μεταβλητή αλλά υπολογίζεται από την καταστατική εξίσωση (5).

3.2 Μηχανισμοί Αντίδρασης

Πέραν από τους αγνώστους, το ανωτέρω σύστημα περιλαμβάνει ιδιότητες που δεν είναι σταθερές αλλά εξαρτώνται από τους αγνώστους. Αυτές οι ιδιότητες είναι:

- οι ταχύτητες διάχυσης V_k που εξαρτώνται από τα Y_k και T
- οι ρυθμοί παραγωγής ω_k που εξαρτώνται από τα Y_k και T
- οι ειδικές θερμότητες c_{pk} και c_p που εξαρτώνται από το T
- οι ειδικές ενθαλπίες h_k που εξαρτώνται από το T
- η θερμική αγωγιμότητα λ που εξαρτάται από το T

Αυτό σημαίνει ότι στην επαναληπτική μέθοδο που περιγράφεται παρακάτω για την επίλυση του ανωτέρω PDE αυτές οι παράμετροι θα πρέπει να επανυπολογίζονται σε κάθε βήμα της διαδικασίας.

3.2.1 Κινητικές Ιδιότητες

3.2.1.1 Ταχύτητες Διάχυσης

Για τις ταχύτητες διάχυσης V_k, ισχύει:

$$V_k = v_k + w_k + V_c \tag{6}$$

Στην παραπάνω σχέση, η v_k είναι η συνήθης ταχύτητα διάχυσης, που υπολογίζεται με βάση την προσέγγιση των Curtiss-Hirschfelder από την εξίσωση (Bird, et al., 1960):

$$v_k = -\frac{D_{km}}{X_k}\frac{\partial X_k}{\partial r} \tag{7a}$$

όπου X_k είναι το μοριακό κλάσμα και D_{km} είναι η σταθερά διάχυσης μεθοδολογίας mixture average, που ορίζεται ως:

$$D_{km} = \frac{\sum_{j\neq k}^{K} \hat{X}_j W_j}{\bar{W} \sum_{j\neq k}^{K} \frac{\hat{X}_j}{\mathcal{D}_{jk}}} \tag{7b}$$

όπου $\hat{X}_j = X_j + \delta$ και δ είναι ένας πολύ μικρός αριθμός, της τάξης του 10^{-12}, που χρησιμοποιείται ώστε η εξίσωση να μπορεί να οριστεί ακόμα και στην περίπτωση που το μίγμα γίνει ένα και μόνο συστατικό. Το $\mathcal{D}_{jk}$ είναι η δυαδική σταθερά διάχυσης, που είναι συνάρτηση της θερμοκρασίας T και της πίεσης p σύμφωνα με την παρακάτω σχέση:

$$\left[\mathcal{D}_{jk}\right]_1 = \frac{3}{16}\frac{\sqrt{2\pi k_B^3 T^3 m_{jk}^{-1}}}{p\pi\sigma_{jk}^2 \Omega^{(1,1)*}} \tag{7c}$$

όπου k_B είναι η σταθερά του Boltzmann, m_{jk} η ανηγμένη μοριακή μάζα για το ζεύγος συστατικών (j,k), σ_{jk} είναι η ανηγμένη διάμετρος σύγκρουσης και $\Omega^{(1,1)*}$ είναι το ολοκλήρωμα σύγκρουσης που βασίζεται στα δυναμικά του Stockmayer (Monchick & Mason, 1961). Η παραπάνω σχέση δίνει την πρώτη προσέγγιση $\left[\mathcal{D}_{jk}\right]_1$ του $\mathcal{D}_{jk}$, ενώ η προσέγγιση δεύτερης τάξης $\left[\mathcal{D}_{jk}\right]_2$ δίνεται από τη σχέση:

$$\mathcal{D}_{jk} \cong \left[\mathcal{D}_{jk}\right]_2 \equiv \left[\mathcal{D}_{jk}\right]_1 \left(1 + \Delta_{jk}\right) \tag{7d}$$

όπου

$$\Delta_{jk} = \frac{\left(6C_{jk}^* - 5\right)^2}{10} \left(\frac{X_j^2 P_j + X_k^2 P_k + X_j X_k P_{jk}}{X_j^2 Q_j + X_k^2 Q_k + X_j X_k Q_{jk}}\right) \tag{7e}$$

ενώ οι παράμετροι P και Q είναι περίπλοκες εκφράσεις, που περιλαμβάνουν ολοκληρώματα συγκρούσεων (Marrero & Mason, 1972).

Λόγω της πολυπλοκότητας υπολογισμού των $\mathcal{D}_{jk}$ με αυτό τον τρόπο, υπολογίζουμε τα $\mathcal{D}_{jk}$ χρησιμοποιώντας πολυωνυμική προσέγγιση της μορφής:

$$\ln \mathcal{D}_{jk} = \sum_{n=1}^{N} d_{n,jk} (\ln T)^{n-1} \tag{7f}$$

Οι σταθερές $d_{n,jk}$ υπολογίζονται για κάθε ζεύγος συστατικών (j,k) πριν την επίλυση του PDE με επαναληπτική διαδικασία χρησιμοποιώντας κατάλληλες ρουτίνες του Matlab. Οι ρουτίνες αυτές είναι ίδιες με αυτές που χρησιμοποιούνται και από το CHEMKIN TRANSPORT για τη δημιουργία των προσεγγίσεων. Σημειώνεται ότι οι σταθερές διάχυσης εξαρτώνται

από την πίεση και οι προσεγγίσεις έχουν υπολογιστεί με βάση πίεση 1 atm. Συνεπώς, όταν υπολογιστούν οι σταθερές διάχυσης από τις πολυωνυμικές προσεγγίσεις, θα πρέπει το αποτέλεσμα να διαιρεθεί με την πραγματική πίεση.

Μετά από συνδυασμό των σχέσεων (7f), (7b) και (7a) η συνθήκη ταχύτητας διάχυσης v_k δίνεται από τη σχέση:

$$v_k = -\frac{\sum_{j\neq k}^{K}(X_j+\delta)W_j}{\sum_{j\neq k}^{K}\frac{X_j+\delta}{\exp\sum_{n=1}^{N} d_{n,jk}(\ln T)^{n-1}}}\frac{1}{\bar{W}X_k}\frac{\partial X_k}{\partial r} \tag{7g}$$

Στην προηγούμενη σχέση (6), η w_k είναι η θερμική ταχύτητα διάχυσης, που ορίζεται ως:

$$w_k = -\frac{D_{km}\Theta_k}{X_k}\frac{1}{T}\frac{\partial T}{\partial r} \tag{8a}$$

Θεωρούμε ότι θερμική ταχύτητα διάχυσης υπάρχει μόνο για τα ελαφριά συστατικά, δηλαδή αυτά που έχουν μοριακό βάρος W_k μικρότερο από 5, ενώ για τις άλλες περιπτώσεις θεωρούμε ότι $w_k = 0$. Το Θ_k είναι ο βαθμός θερμικής διάχυσης, που δίνεται από τη σχέση:

$$\Theta_k = \sum_{j\neq k}^{K}\theta_{kj} \tag{8b}$$

Όπου

$$\theta_{kj} = \frac{15}{2}\frac{(2A_{kj}^*+5)(6C_{kj}^*-5)}{A_{kj}^*(16A_{kj}^*-12B_{kj}^*+55)}\frac{W_k-W_j}{W_k+W_j}X_kX_j \tag{8c}$$

όπου οι όροι A_{kj}^*, B_{kj}^* και C_{kj}^* περιέχουν και αυτοί ολοκληρώματα συγκρούσεων και δίνονται από πίνακες (Monchick & Mason, 1961). Συνεπώς η σχέση (9b) είναι περίπλοκη και αντί αυτής θα χρησιμοποιηθεί πολυωνυμική προσέγγιση για τον όρο $\theta_{kj}^* = \theta_{kj}/(X_j X_k)$ ως εξής:

$$\theta_{kj}^* = \sum_{n=1}^{N} c_{n,kj} T^{n-1} \tag{8d}$$

όπου και πάλι οι σταθερές $c_{n,jk}$ υπολογίζονται για κάθε ζεύγος συστατικών (j,k) πριν την επίλυση του PDE με επαναληπτική διαδικασία χρησιμοποιώντας κατάλληλες ρουτίνες του Matlab. Οι ρουτίνες αυτές είναι ίδιες με αυτές που χρησιμοποιούνται και από το CHEMKIN TRANSPORT για τη δημιουργία των προσεγγίσεων. Σημειώνεται ότι δεν χρησιμοποιείται λογαριθμική προσέγγιση, όπως προηγουμένως, διότι ο ρυθμός θερμικής διάχυσης εξαρτάται ασθενώς από τη θερμοκρασία.

Κατόπιν συνδυασμού των (8d) και (8b) η θερμική ταχύτητα διάχυσης δίνεται από τη σχέση:

$$w_k = \begin{cases} -\dfrac{\sum_{j\neq k}^{K}(X_j+\delta)W_j}{\sum_{j\neq k}^{K}\dfrac{X_j+\delta}{\exp\sum_{n=1}^{N} d_{n,jk}(\ln T)^{n-1}}} \displaystyle\sum_{j\neq k}^{K}\left(X_j X_k \sum_{n=1}^{N} c_{n,kj} T^{n-1}\right)\frac{1}{X_k T}\frac{\partial T}{\partial r} & W_k \leq 5 \\ 0 & W_k > 5 \end{cases} \tag{8e}$$

Ο όρος V_c στη σχέση (6) είναι η ταχύτητα διόρθωσης, που συμπεριλαμβάνεται για να διασφαλίσει ότι το άθροισμα των κλασμάτων μάζας όλων των συστατικών είναι μονάδα (ή ισοδύναμα $\sum_{k=1}^{K} Y_k V_k = 0$), δηλαδή:

$$\sum_{k=1}^{K} Y_k(v_k + w_k + V_c) = 0 \Leftrightarrow V_c = -\sum_{k=1}^{K} Y_k(v_k + w_k) \tag{9}$$

3.2.1.2 Θερμικές Αγωγιμότητες

Η ολική θερμική αγωγιμότητα του μίγματος λ στην περίπτωση μεθοδολογίας mixture-average δίνεται από τη σχέση:

$$\lambda = \frac{1}{2}\left(\sum_{k=1}^{K} X_k \lambda_k + \frac{1}{\sum_{k=1}^{K} X_k/\lambda_k}\right) \tag{10a}$$

όπου οι θερμικές αγωγιμότητες των συστατικών αποτελούνται από συνεισφορές μεταφοράς, περιστροφής και δόνησης ως εξής:

$$\lambda_k = \frac{\eta_k}{W_k}\left(f_{\text{trans.}} C_{\upsilon,\text{trans.}} + f_{\text{rot.}} C_{\upsilon,\text{rot.}} + f_{\text{vib.}} C_{\upsilon,\text{vib.}}\right) \tag{10b}$$

Οι όροι της σχέσης (10b) είναι πολύ περίπλοκοι για να υπολογίζονται σε κάθε βήμα της επαναληπτικής διαδικασίας επίλυσης, συνεπώς το λ_k υπολογίζεται από πολυωνυμική προσέγγιση με βάση τη σχέση:

$$\ln \lambda_k = \sum_{n=1}^{N} b_{n,k} (\ln T)^{n-1} \tag{10c}$$

Οι σταθερές $b_{n,k}$, όπως και προηγουμένως, υπολογίζονται για κάθε συστατικό k πριν την επαναληπτική διαδικασία επίλυσης του PDE χρησιμοποιώντας κατάλληλες ρουτίνες του Matlab. Οι ρουτίνες αυτές είναι ίδιες με αυτές που χρησιμοποιεί το CHEMKIN TRANSPORT για να δημιουργήσει τις προσεγγίσεις.

3.2.1.3 Απώλεια Θερμότητας από Ακτινοβολία

Η απώλεια θερμότητας από ακτινοβολία ανά μονάδα όγκου q, και με την παραδοχή οπτικά λεπτής φλόγας, δίνεται από τη σχέση (Siegel & Howell, 1981):

$$q = 4a_p\sigma(T^4 - T_\infty^4) \tag{11a}$$

όπου σ είναι η σταθερά Stefan-Boltzmann, T_∞ είναι η θερμοκρασία μακριά από τη φλόγα και a_p είναι η συνολική μέση σταθερά απορρόφησης του Planck. Η συνολική μέση σταθερά απορρόφησης του Planck προκύπτει από την απορρόφηση και εκπομπή των αέριων συστατικών CO, CO_2 και H_2O και δίνεται από τη σχέση:

$$a_p = p\left(X_{CO}a_{p,CO} + X_{CO_2}a_{p,CO_2} + X_{H_2O}a_{p,H_2O}\right) \tag{11b}$$

όπου $a_{p,XX}$ είναι η μέση σταθερά απορρόφησης του συστατικού XX, που δίνεται από τη σχέση (Tien, 1967):

$$a_{p,XX} = \sum_j \frac{\pi \bar{I}_{b\,j,XX} a_{j,XX}}{\sigma T^4} \tag{11c}$$

όπου η άθροιση γίνεται για όλες τις ταινίες δόνησης-περιστροφής j της ακτινοβολίας του συστατικού XX, $\bar{I}_{bj,XX}$ είναι η μέση ένταση μαύρου σώματος για την ταινία j και $a_{j,XX}$ είναι η ολική ένταση της ταινίας j. Η μέση ένταση μαύρου σώματος δίνεται από το νόμο του Planck ως εξής:

$$\bar{I}_{bj,XX} = \frac{2hc_0^2\tilde{v}_{j,XX}^3}{e^{hc_0\tilde{v}_{j,XX}/(k_BT)} - 1} \tag{11}$$

όπου h είναι η σταθερά του Planck, c_0 είναι η ταχύτητα του φωτός στο κενό και $\tilde{v}_{j,XX}$ είναι ο κυματάριθμος της ταινίας j, που ισούται με το κέντρο της ταινίας. Οι ποσότητες $a_{j,XX}$ και $\tilde{v}_{j,XX}$ δίνονται από πίνακες για τα συστατικά CO, CO_2 και H_2O. Έπειτα από συνδυασμό των (11a) έως (11d) προκύπτει:

$$q = 8\pi hc_0^2 p\left(1 - \frac{T_\infty^4}{T^4}\right)\left[\sum_j \frac{\tilde{v}_{j,CO}^3 a_{j,CO}}{e^{hc_0\tilde{v}_{j,CO}/(k_BT)} - 1} + \sum_j \frac{\tilde{v}_{j,CO_2}^3 a_{j,CO_2}}{e^{hc_0\tilde{v}_{j,CO_2}/(k_BT)} - 1} + \sum_j \frac{\tilde{v}_{j,H_2O}^3 a_{j,H_2O}}{e^{hc_0\tilde{v}_{j,H_2O}/(k_BT)} - 1}\right] \tag{11e}$$

3.2.2 Θερμοδυναμικές Ιδιότητες

3.2.2.1 Ρυθμοί Παραγωγής

Ο ρυθμός παραγωγής ω_k, όπου για λόγους απλότητας παραλήφθηκε το σύμβολο της τελείας πάνω από το ω, μπορεί να γραφεί ως το άθροισμα των εντατικών ταχυτήτων όλων των αντιδράσεων που περιλαμβάνουν το συστατικό k:

$$\omega_k = \sum_{i=1}^{I} \upsilon_{ki} q_i \tag{12}$$

όπου $\upsilon_{ki} = \upsilon''_{ki} - \upsilon'_{ki}$ είναι ο στοιχειομετρικός αριθμός (στοιχειομετρικός συντελεστής προϊόντος μείον στοιχειομετρικός συντελεστής αντιδρώντος) του συστατικού k στην αντίδραση i, και q_i είναι η εντατική ταχύτητα της αντίδρασης i που εκφράζεται ως η διαφορά της θετικής και αρνητικής ταχύτητας ως εξής:

$$q_i = k_{f_i} \prod_{k=1}^{K} [X_k]^{\upsilon'_{ki}} - k_{r_i} \prod_{k=1}^{K} [X_k]^{\upsilon''_{ki}} \tag{13}$$

όπου $[X_k]$ είναι η μοριακή συγκέντρωση του συστατικού k και k_{f_i} και k_{r_i} είναι οι κανονικοί και αντίστροφοι συντελεστές ταχύτητας της αντίδρασης i.

Ο κανονικός συντελεστής ταχύτητας k_{f_i} υπολογίζεται γενικά με βάση την εξίσωση του Arrhenius:

$$k_{f_i} = A_i T^{\beta_i} \exp\left(\frac{-E_i}{R_c T}\right) \tag{14}$$

όπου ο όρος A_i, ο θερμοκρασιακός εκθέτης β_i και η ενέργεια ενεργοποίησης E_i είναι δεδομένες για κάθε αντίδραση και R_c είναι η παγκόσμια σταθερά των αερίων, εκπεφρασμένη σε calories/mole.

Ο αντίστροφος συντελεστής ταχύτητας k_{r_i} υπολογίζεται γενικά από τη σχέση:

$$k_{r_i} = \frac{k_{f_i}}{K_{c_i}} \tag{15a}$$

όπου K_{c_i} είναι η σταθερά ισορροπίας, που δίνεται από τη σχέση:

$$K_{c_i} = K_{p_i} \left(\frac{P_{atm}}{RT}\right)^{\sum_{k=1}^{K} \upsilon_{ki}} \tag{15b}$$

ενώ η σταθερά ισορροπίας K_{p_i} υπολογίζεται από τη σχέση:

$$K_{p_i} = \exp\left(\frac{\Delta S_i^0}{R} - \frac{\Delta H_i^0}{RT}\right) \tag{15c}$$

όπου οι όροι μέσα στην παρένθεση δίνονται από τις σχέσεις:

$$\frac{\Delta S_i^0}{R} = \sum_{k=1}^{K} \upsilon_{ki} \frac{S_k^0}{R} \tag{15d}$$

$$\frac{\Delta H_i^0}{RT} = \sum_{k=1}^{K} \upsilon_{ki} \frac{H_k^0}{RT} \tag{15e}$$

ενώ οιεκφράσεις S_k^0/R και $H_k^0/(RT)$ υπολογίζονται από πολυώνυμα προσέγγισης ως εξής:

$$\frac{S_k^0}{R} = a_{1k} \ln T + a_{2k} T + \frac{a_{3k}}{2} T^2 + \frac{a_{4k}}{3} T^3 + \frac{a_{5k}}{4} T^4 + a_{7k} \tag{15f}$$

$$\frac{H_k^0}{RT} = a_{1k} + \frac{a_{2k}}{2}T + \frac{a_{3k}}{3}T^2 + \frac{a_{4k}}{4}T^3 + \frac{a_{5k}}{5}T^4 + \frac{a_{6k}}{T} \tag{15g}$$

Στις εκφράσεις αυτές, οι συντελεστές a_{1k} έως a_{7k} δίνονται από θερμοδυναμικούς πίνακες για κάθε συστατικό.

Οι παραπάνω σχέσεις (14) και (15a) έως (15g) χρησιμοποιούνται σε απλές περιπτώσεις, όταν η αντίδραση δεν εξαρτάται από την πίεση ή όταν δεν περιλαμβάνει τρίτο σώμα ή πιο ειδικές πληροφορίες σχετικά με τους κανονικούς ή αντίστροφους ρυθμούς και τους στοιχειομετρικούς συντελεστές. Όμως, όταν υπάρχουν τρίτα σώματα ή εξαρτήσεις από την πίεση τότε ο υπολογισμός των συντελεστών ταχύτητας διαφέρει.

3.2.2.2 Ειδικές Θερμότητες

Οι θερμοδυναμικές ιδιότητες των αντιδράσεων, c_p και c_{pk} μπορούν να υπολογιστούν από τις παρακάτω σχέσεις:

$$c_{pk} = \frac{R}{W_k}(a_{1k} + a_{2k}T + a_{3k}T^2 + a_{4k}T^3 + a_{5k}T^4) \tag{16}$$

όπου οι σταθερές $a_{jk}, j = 1 \dots 5$ είναι οι ίδιες με αυτές των σχέσεων (14f) και (14g).

Η μέση ειδική θερμότητα σε σταθερή πίεση c_p υπολογίζεται από τη σχέση:

$$c_p = \sum_{k=1}^{K}(c_{pk}Y_k) \tag{17}$$

3.2.2.3 Ειδικές Ενθαλπίες

Οι ειδικές ενθαλπίες h_k υπολογίζονται από τη σχέση (15g) ως εξής:

$$h_k = \frac{H_k}{W_k} = \frac{RT}{W_k}\left(a_{1k} + \frac{a_{2k}}{2}T + \frac{a_{3k}}{3}T^2 + \frac{a_{4k}}{4}T^3 + \frac{a_{5k}}{5}T^4 + \frac{a_{6k}}{T}\right) \tag{18}$$

4 Αριθμητική Περιγραφή του Προβλήματος

Οι προαναφερόμενες διαφορικές εξισώσεις δεν μπορούν να επιλυθούν αναλυτικά και θα μοντελοποιηθούν χρησιμοποιώντας τη μέθοδο Πεπερασμένων Διαφορών, όπου τόσο το χρονικό όσο και το χωρικό πεδίο διακριτοποιούνται με σκοπό τη μετατροπή των μερικών διαφορικών σε απλές μαθηματικές παραστάσεις, που μπορούν να επιλυθούν με τη βοήθεια ηλέκτρονικού υπολογιστή.

Αρχικά συγκροτείται το πλήρες σύστημα PDE μαζί με τις συνοριακές συνθήκες, τόσο για την κατάσταση ισορροπίας όσο και για τη μεταβατική κατάσταση και χρησιμοποιείται μία μεθοδολογία επίλυσης που στηρίζεται σε επαναληπτικές προσεγγιστικές τεχνικές, όπως η Newton-Raphson.

Το σύστημα PDE εκφράζεται με τέτοιο τρόπο ώστε μελλοντικά να μπορεί να μετατραπεί σε σύστημα ODE της μορφής $\partial\phi/\partial t = F(\phi, a)$, κατάλληλο για το Matcont.

4.1 Συνοριακές Συνθήκες

Στο πρόβλημα σφαιρικών φλογών διάχυσης, καύσιμο μίγμα εισέρχεται από σημειακή πηγή σε θάλαμο που περιέχει οξειδωτικό με ρυθμό ροής μάζας $\dot{m}_0$ και θερμοκρασία T_0. Συνεπώς, στην μεριά της εισόδου, οι συνοριακές συνθήκες του συστήματος είναι οι εξής (Christiansen, et al., 2003):

$$\left.\begin{array}{c}\dot{m} = \dot{m}_0 \\ Y_k = \epsilon_k \\ T = T_0\end{array}\right|_{r=0} \tag{19a}$$

όπου ϵ_k είναι το κλάσμα μάζας του συστατικού k στο αρχικό μίγμα.

Στην άλλη πλευρά, όπου $r = r_\infty$, οι συνοριακές συνθήκες του συστήματος είναι οι εξής:

$$\left.\begin{array}{c}\frac{\partial \dot{m}}{\partial r} = 0 \\ Y_k = Y_{k,\infty} \\ T = T_\infty\end{array}\right|_{r=r_\infty} \tag{19b}$$

όπου $Y_{k,\infty}$ είναι το κλάσμα μάζας του συστατικού k του οξειδωτικού και T_∞ είναι η θερμοκρασία μακριά από την αντίδραση.

4.2 Χωρική Διακριτοποίηση

Οι διαφορικές εξισώσεις που διέπουν το πρόβλημα διακριτοποιούνται χρησιμοποιώντας ανομοιογενές πλέγμα για τη χωρική συντεταγμένη r. Επιλέγεται ανομοιογενές πλέγμα λόγω των μεγάλων μεταβολών των βασικών μεταβλητών (θερμοκρασία, κλάσματα μάζας) κατά μήκος του πεδίου του r. Περιοχές όπου συμβαίνουν μεγάλες μεταβολές απαιτούν πιο πυκνό πλέγμα προκειμένου να μη χαθεί ακρίβεια και πληροφορία εκεί. Η αναθεώρηση του πλέγματος και η τοποθέτηση περισσότερων κόμβων δυναμικά γίνεται επιλύοντας το πρόβλημα ξανά και ξανά, μέχρι οι κλίσεις των μεταβλητών T, $\dot{m}$ και Y_k να είναι μικρότερες από κάποιο όριο. Αν και αυτή η μεθοδολογία είναι περίπλοκη, οδηγεί σε μικρότερο συνολικό αριθμό κόμβων από τη χρήση ενός ομοιογενούς πυκνού πλέγματος.

Στην εξίσωση (1), το χωρικό διαφορικό διακριτοποιείται χρησιμοποιώντας πεπερασμένες διαφορές προς τα πίσω, λόγω του υπερβολικού της χαρακτήρα:

$$\left.\frac{\partial \dot{m}}{\partial r}\right|_i = \frac{\dot{m}_i - \dot{m}_{i-1}}{h_i} + O(h_i) \tag{20a}$$

όπου $h_i = r_i - r_{i-1}$, οπότε η εξίσωση (1) γίνεται:

$$\frac{\partial \rho_i}{\partial t} = -\frac{\dot{m}_i - \dot{m}_{i-1}}{A_i h_i} \tag{20b}$$

Θα πρέπει να σημειωθεί ότι η πυκνότητα ρ δεν είναι ανεξάρτητη μεταβλητή, αλλά εξαρτάται από τις μεταβλητές Y_k και T σύμφωνα με την εξίσωση (5). Συνεπώς, η χρονική παράγωγος της πυκνότητας μπορεί να γραφεί ως εξής:

$$\frac{\partial \rho_i}{\partial t} = \frac{\partial}{\partial t}\left(\frac{p\overline{\overline{W}}_i}{RT_i}\right) = \frac{\partial}{\partial t}\left(\frac{p}{RT_i \sum_{k=1}^{K} Y_{k,i}/W_{k,i}}\right) \Leftrightarrow$$

$$\frac{\partial \rho_i}{\partial t} = -\rho_i \overline{W}_i \sum_{k=1}^{K}\left(\frac{1}{W_{k,i}}\frac{\partial Y_{k,i}}{\partial t}\right) - \frac{\rho_i}{T_i}\frac{\partial T_i}{\partial t} \tag{20c}$$

Η εξίσωση (20c) δεν είναι εύκολη στον υπολογισμό της ρ_i και είναι πιο εύκολη η (5). Παρ' όλα αυτά, η έκφραση αυτή είναι χρήσιμη μελλοντικά κατά την μετατροπή όλου του συστήματος PDE σε ODE με σκοπό να χρησιμοποιηθεί από το Matcont.

Στην εξίσωση (2), η χωρική μερική παράγωγος των κλασμάτων μάζας των συστατικών διακριτοποιείται χρησιμοποιώντας κεντρικές διαφορές ως εξής:

$$\dot{m}\frac{\partial Y_k}{\partial r}\bigg|_i = \dot{m}_i\left[\frac{h_i}{h_{i+1}(h_{i+1}+h_i)}(Y_k)_{i+1} + \frac{h_{i+1}-h_i}{h_{i+1}h_i}(Y_k)_i - \frac{h_{i+1}}{h_i(h_{i+1}+h_i)}(Y_k)_{i-1} + O(h_i^2)\right] \tag{21a}$$

όπου σημειώνεται ότι τα κομβικά βήματα h_i διαφέρουν για κάθε i.

Στην εξίσωση (2) ο όρος $\frac{\partial}{\partial r}(\rho A Y_k V_k)$ διακριτοποιείται σύμφωνα με τη σχέση:

$$\frac{\partial}{\partial r}(\rho A Y_k V_k)\bigg|_i = \frac{(\rho A Y_k V_k)_{i+\frac{1}{2}} - (\rho A Y_k V_k)_{i-\frac{1}{2}}}{h_{i+\frac{1}{2}}} + O(h_i^2) \tag{21b}$$

όπου ρ, A, Y_k και V_k υπολογίζονται στην ενδιάμεση θέση $i \pm \frac{1}{2}$ ως μέσος όρος των δύο διπλανών κόμβων. Η έκφραση $(Y_k V_k)_j$, όπου $j = i \pm \frac{1}{2}$, υπολογίζεται με βάση τις ακόλουθες σχέσεις γνωρίζοντας ότι $Y_k = (X_k W_k)/\overline{W}$:

$$(Y_k v_k)_j = -\left(\frac{W_k D_{km}}{\overline{W}}\right)_j \left(\frac{X_{k,j+\frac{1}{2}} - X_{k,j-\frac{1}{2}}}{h_{j+\frac{1}{2}}}\right) \tag{21c}$$

$$(Y_k w_k)_j = -\left(\frac{W_k D_{km} \Theta_k}{\overline{W} T}\right)_j \left(\frac{T_{j+\frac{1}{2}} - T_{j-\frac{1}{2}}}{h_{j+\frac{1}{2}}}\right) \tag{21d}$$

όπου οι όροι v_k και w_k είναι οι επιμέρους ταχύτητες διάχυσης της σχέσης (4) ενώ η V_c υπολογίζεται με βάση τη σχέση (9).

Ο τελευταίος όρος της (2) περιλαμβάνει το ρυθμός παραγωγής ω_k που εξαρτάται από τις άγνωστες μεταβλητές και υπολογίζεται με βάση αυτές στον κόμβο i, συνεπώς η εξίσωση (2) γίνεται:

$$\frac{\partial Y_{k,i}}{\partial t} = \frac{\omega_{k,i}}{\rho_i} W_{k,i} - \frac{\dot{m}_i}{A_i \rho_i}\left[\frac{h_i}{h_{i+1}(h_{i+1}+h_i)} Y_{k,i+1} + \frac{h_{i+1}-h_i}{h_{i+1}h_i} Y_{k,i} - \frac{h_{i+1}}{h_i(h_{i+1}+h_i)} Y_{k,i-1}\right] - \frac{(A\rho Y_k V_k)_{i+\frac{1}{2}} - (A\rho Y_k V_k)_{i-\frac{1}{2}}}{A_i \rho_i h_{i+\frac{1}{2}}} \tag{21e}$$

Στην εξίσωση διατήρησης της ενέργειας (3) η χωρική παράγωγος της θερμοκρασίας διακριτοποιείται με κεντρικές διαφορές σύμφωνα με την παρακάτω σχέση:

$$\dot{m}\frac{\partial T}{\partial r}\bigg|_i = \dot{m}_i\left[\frac{h_i}{h_{i+1}(h_{i+1}+h_i)} T_{i+1} + \frac{h_{i+1}-h_i}{h_{i+1}h_i} T_i - \frac{h_{i+1}}{h_i(h_{i+1}+h_i)} T_{i-1} + O(h_i^2)\right] \tag{22a}$$

Ενώ ο επόμενος όρος διακριτοποιείται σύμφωνα με μία έκφραση κεντρικών διαφορών δεύτερης τάξης ως εξής:

$$\frac{\partial}{\partial r}\left(A\lambda\frac{\partial T}{\partial r}\right)\bigg|_i = \frac{2}{h_{i+1}+h_i}\left[(A\lambda)_{i+\frac{1}{2}}\left(\frac{T_{i+1}-T_i}{h_{j+1}}\right) - (A\lambda)_{i-\frac{1}{2}}\left(\frac{T_i - T_{i-1}}{h_j}\right)\right] + O(h_i^2) \tag{22b}$$

Η θερμική αγωγιμότητα λ υπολογίζεται χρησιμοποιώντας το μέσο όρο των μεταβλητών από τις οποίες εξαρτάται μεταξύ των κόμβων:

$$\lambda_{j+\frac{1}{2}} = \lambda\left(\frac{T_{j+1}+T_j}{2}, \frac{Y_{k,j+1}+Y_{k,j}}{2}\right) \tag{22c}$$

Οι υπόλοιποι όροι της εξίσωσης (3) εξαρτώνται και αυτοί από τις άγνωστες μεταβλητές και υπολογίζονται στον κόμβο i, οπότε τελικά η εξίσωση (3) γίνεται:

$$\frac{\partial T_i}{\partial t} = -\frac{1}{\rho_i c_{p,i}}\left\{\left[\frac{\dot{m}_i c_{p,i}}{A_i} + \rho_i \sum_{k=1}^{K}\left(Y_k V_k c_{pk}\right)_i\right]\left[\frac{h_i}{h_{i+1}(h_{i+1}+h_i)}T_{i+1} + \frac{h_{i+1}-h_i}{h_{i+1}h_i}T_i\right.\right.$$
$$\left. - \frac{h_{i+1}}{h_i(h_{i+1}+h_i)}T_{i-1}\right] - \frac{2}{A_i(h_{i+1}+h_i)}\left[(A\lambda)_{i+\frac{1}{2}}\left(\frac{T_{i+1}-T_i}{h_{i+1}}\right) - (A\lambda)_{i-\frac{1}{2}}\left(\frac{T_i - T_{i-1}}{h_i}\right)\right] \tag{22d}$$
$$\left. + \sum_{k=1}^{K}(\omega_k h_k W_k)_i + q_i\right\}$$

4.3 Κατάσταση Ισορροπίας

Κατόπιν μοντελοποίησης, οι εξισώσεις που διέπουν το σύστημα, εκπεφρασμένες σε διακριτοποιημένη μορφή, είναι οι (18b), (19e) και (20d). Αν K είναι ο αριθμός των συστατικών και N είναι ο αριθμός των αρχικά ορισμένων κόμβων, τότε το σύστημα αποτελείται από $(K+2)\times N$ εξισώσεις με $(K+2)\times N$ εγνώστους.

Στην περίπτωση της κατάστασης ισορροπίας, τα αριστερά μέρη των εξισώσεων (18b), (19e) και (20d) είναι μηδέν και το σύστημα μπορεί να εκφραστεί στη μορφή:

$$F(\phi) = 0 \tag{23}$$

όπου ϕ είναι το διάνυσμα των $(K+2)\times N$ αγνώστων, που ορίζεται ως:

$$\phi = [T_1 \quad Y_{1,1} \quad \cdots \quad Y_{K,1} \quad \dot{m}_1 \quad \cdots \quad \cdots \quad T_N \quad Y_{1,N} \quad \cdots \quad Y_{K,N} \quad \dot{m}_N]^T \tag{24}$$

και F είναι το διάνυσμα των $(K+2)\times N$ εξισώσεων. Αν λάβουμε υπόψη τις συνοριακές συνθήκες (19a) και (19b), τότε το F ορίζεται ώς:

$$
F = \begin{bmatrix}
-\dot{m}_1 + \dot{m}_0 \\
-Y_{1,1} + \epsilon_1 \\
\vdots \\
-Y_{K,1} + \epsilon_K \\
-T_1 + T_0 \\
-\dfrac{\dot{m}_2 - \dot{m}_1}{h_2 A_2} \\
\dfrac{\omega_{1,2}}{\rho_2} W_{1,2} - \dfrac{\dot{m}_2}{\rho_2 A_2}\left[\dfrac{h_2}{h_3(h_3+h_2)} Y_{1,3} + \dfrac{h_3 - h_2}{h_3 h_2} Y_{1,2} - \dfrac{h_3}{h_2(h_3+h_2)} Y_{1,1}\right] - \dfrac{(\rho A Y_1 V_1)_{\frac{5}{2}} - (\rho A Y_1 V_1)_{\frac{3}{2}}}{\rho_2 A_2 h_{\frac{5}{2}}} \\
\vdots \\
\dfrac{\omega_{K,2}}{\rho_2} W_{K,2} - \dfrac{\dot{m}_2}{\rho_2 A_2}\left[\dfrac{h_2}{h_3(h_3+h_2)} Y_{K,3} + \dfrac{h_3 - h_2}{h_3 h_2} Y_{K,2} - \dfrac{h_3}{h_2(h_3+h_2)} Y_{K,1}\right] - \dfrac{(\rho A Y_K V_K)_{\frac{5}{2}} - (\rho A Y_K V_K)_{\frac{3}{2}}}{\rho_2 A_2 h_{\frac{5}{2}}} \\
-\dfrac{1}{\rho_2 c_{p,2}}\left\{\left[\dfrac{\dot{m}_2 c_{p,2}}{A_2} + \rho_2 \sum_{k=1}^{K} \left(Y_k V_k c_{pk}\right)_2\right]\left[\dfrac{h_2}{h_3(h_3+h_2)} T_3 + \dfrac{h_3 - h_2}{h_3 h_2} T_2 - \dfrac{h_3}{h_2(h_3+h_2)} T_1\right] - \right. \\
\left. \dfrac{2}{(h_3+h_2)A_2}\left[(A\lambda)_{\frac{5}{2}}\left(\dfrac{T_3 - T_2}{h_3}\right) - (A\lambda)_{\frac{3}{2}}\left(\dfrac{T_2 - T_1}{h_2}\right)\right] + \sum_{k=1}^{K} (\omega_k h_k W_k)_2 + q_2\right\} \\
\vdots \\
\vdots \\
-\dfrac{\dot{m}_{N-1} - \dot{m}_{N-2}}{h_{N-1} A_{N-1}} \\
\dfrac{\omega_{1,N-1}}{\rho_{N-1}} W_{1,N-1} - \dfrac{\dot{m}_{N-1}}{\rho_{N-1} A_{N-1}}\left[\dfrac{h_{N-1}}{h_N(h_N+h_{N-1})} Y_{1,N} + \dfrac{h_N - h_{N-1}}{h_N h_{N-1}} Y_{1,N-1} - \dfrac{h_N}{h_{N-1}(h_N+h_{N-1})} Y_{1,N-2}\right] - \dfrac{(\rho A Y_1 V_1)_{N-\frac{1}{2}} - (\rho A Y_1 V_1)_{N-\frac{3}{2}}}{\rho_{N-1} A_{N-1} h_{N-\frac{1}{2}}} \\
\vdots \\
\dfrac{\omega_{K,N-1}}{\rho_{N-1}} W_{K,N-1} - \dfrac{\dot{m}_{N-1}}{\rho_{N-1} A_{N-1}}\left[\dfrac{h_{N-1}}{h_N(h_N+h_{N-1})} Y_{K,N} + \dfrac{h_N - h_{N-1}}{h_N h_{N-1}} Y_{K,N-1} - \dfrac{h_N}{h_{N-1}(h_N+h_{N-1})} Y_{K,N-2}\right] - \dfrac{(\rho A Y_K V_K)_{N-\frac{1}{2}} - (\rho A Y_K V_K)_{N-\frac{3}{2}}}{\rho_{N-1} A_{N-1} h_{N-\frac{1}{2}}} \\
-\dfrac{1}{\rho_{N-1} c_{p,N-1}}\left\{\left[\dfrac{\dot{m}_{N-1} c_{p,N-1}}{A_{N-1}} + \rho_{N-1} \sum_{k=1}^{K} \left(Y_k V_k c_{pk}\right)_{N-1}\right]\left[\dfrac{h_{N-1}}{h_N(h_N+h_{N-1})} T_N + \dfrac{h_N - h_{N-1}}{h_N h_{N-1}} T_{N-1} - \dfrac{h_N}{h_2(h_N+h_{N-1})} T_{N-2}\right] - \right. \\
\left. \dfrac{2}{(h_N+h_{N-1})A_{N-1}}\left[(A\lambda)_{N-\frac{1}{2}}\left(\dfrac{T_N - T_{N-1}}{h_N}\right) - (A\lambda)_{N-\frac{3}{2}}\left(\dfrac{T_{N-1} - T_{N-2}}{h_{N-1}}\right)\right] + \sum_{k=1}^{K} (\omega_k h_k W_k)_{N-1} + q_{N-1}\right\} \\
-\dfrac{\dot{m}_N - \dot{m}_{N-1}}{h_N A_N} \\
-Y_{1,N} + Y_{1,\infty} \\
\vdots \\
-Y_{K,N} + Y_{K,\infty} \\
-T_N + T_\infty
\end{bmatrix} \tag{25}
$$

4.4 Τροποποιημένη Μέθοδος Newton Με Απόσβεση

Το ανωτέρω σύστημα (23) μπορεί να επιλυθεί χρησιμοποιώντας την επαναληπτική μέθοδο Newton. Αν ϕ^n είναι η λύση του συστήματος στο βήμα n και ϕ^{n+1} είναι η λύση του συστήματος στο επόμενο βήμα, που δεν είναι ακόμα γνωστή, τότε (Belytschko, et al., 2000):

$$\phi^{n+1} = \phi^n - \left(\frac{\partial F}{\partial \phi}\right)_{\phi^n}^{-1} F(\phi^n) \tag{26}$$

Η παραπάνω διαδικασία είναι αργή διότι η Ιακωβιανή $J = \partial F / \partial \phi$ πρέπει να υπολογίζεται αριθμητικά σε κάθε βήμα. Λόγω της πολυπλοκότητας του συστήματος εξισώσεων δεν μπορεί να δοθεί αναλυτική περιγραφή της Ιακωβιανής. Επιπλέον, το παραπάνω σύστημα απαιτεί μία καλή αρχική εκτίμηση της λύσης ώστε να συγκλίνει.

Για την αντιμετώπιση των παραπάνω προβλημάτων υιοθετείται μία μέθοδος Newton με απόσβεση, χρησιμοποιώντας μία παράμετρο απόσβεσης λ^n ενώ επίσης η Ιακωβιανή δεν υπολογίζεται σε κάθε βήμα, οπότε η εξίσωση (26) γίνεται (Wriggers, 2008):

$$\phi^{n+1} = \phi^n - \lambda^n \left(\frac{\partial F}{\partial \phi}\right)_{\phi^s}^{-1} F(\phi^n) = \phi^n - \lambda^n (J^s)^{-1} F(\phi^n) \Leftrightarrow$$

$$J^s \Delta\phi^n = -\lambda^n F(\phi^n) \tag{27}$$

όπου $\Delta\phi^n = \phi^{n+1} - \phi^n$ και ο δείκτης s, όπου $s \leq n$, στην Ιακωβιανή J^s, υποδηλώνει ότι χρησιμοποιείται η Ιακωβιανή ενός προηγούμενου βήματος s. Η παράμετρος απόσβεσης παίρνει τιμές στην περιοχή $0 < \lambda^n \leq 1$.

Η παράμετρος απόσβεσης λ^n επιλέγεται αρχικά να είναι όσο το δυνατόν μεγαλύτερη. Θεωρώντας μία τιμή για το λ^n υπολογίζουμε το ϕ^{n+1} από την (27). Για να είναι αποδεκτή η

λύση τα μη αποσβενυμένα βήματα $(J^s)^{-1}F(\phi^n)$ θα πρέπει να είναι μικρότερα σε μέγεθος καθώς προχωράμε, δηλαδή:

$$\|(J^s)^{-1}F(\phi^{n+1})\| < \|(J^s)^{-1}F(\phi^n)\|$$

ή ισοδύναμα:

$$\|\phi^{n+1} - \phi^n\| < \|\phi^n - \phi^{n-1}\| \tag{28a}$$

Επιπλέον, η λύση θα πρέπει να έχει φυσικό νόημα, δηλαδή θα πρέπει επίσης να ικανοποιούνται τα παρακάτω κριτήρια:

$$\begin{gathered} \dot{m}^{n+1} \geq 0 \\ 0 \leq Y_k^{n+1} \leq 1 \\ T^{n+1} > 0 \end{gathered} \tag{28b}$$

Αν τα παραπάνω κριτήρια δεν ικανοποιούνται, τότε το λ^n μειώνεται μέχρι να ικανοποιηθούν. Αν το λ^n γίνει πολύ μικρό και η μετάβαση στην επόμενη προσέγγιση προχωρά με πολύ αργό ρυθμό, τότε η Ιακωβιανή επανυπολογίζεται. Αν και πάλι δεν ικανοποιούνται τα κριτήρια, τότε υιοθετείται μία τεχνική χρονικού βηματισμού. Σε αυτή οι χρονικές παράγωγοι των κλασμάτων μάζας των συστατικών καθώς και της θερμοκρασίας υπεισέρχονται στις εξισώσεις (21e) και (22d) και το σύστημα επιλύεται ξανά με την επαναληπτική μέθοδο Newton για μία σειρά χρονικών βημάτων. Η μέθοδος χρονικού βηματισμού περιγράφεται πιο αναλυτικά στις επόμενες παραγράφους, όπου λύνεται το πρόβλημα μεταβατικής κατάστασης. Οι χρονικές παράγωγοι των κλασμάτων μάζας των συστατικών καθώς και της θερμοκρασίας δίνονται από τις σχέσεις:

$$\frac{\partial Y_{k,i}}{\partial t} = \frac{Y_{k,i}^{(n)} - Y_{k,i}^{(n-1)}}{\Delta t}, \frac{\partial T_i}{\partial t} = \frac{T_i^{(n)} - T_i^{(n-1)}}{\Delta t} \tag{29}$$

όπου οι άνω δείκτες (n) και $(n-1)$ υποδηλώνουν τιμές του παρόντος και προηγούμενου χρονικού βήματος αντίστοιχα. Το Δt επιλέγεται μεταξύ 1 έως 10 μsec. Οι εξισώσεις χρονικού βηματισμού είναι πιο εύκολο να συγκλίνουν, ειδικά όσο $\Delta t \to 0$ όπου τότε $\phi^{n+1} \to \phi^n$, όμως επιλέγονται χρονικά βήματα αρκετά μεγάλα ώστε να προχωρά η λύση. Αν ο χρονικός βηματισμός αποτύχει (δε συγκλίνει) τότε επιλέγονται μικρότερα χρονικά βήματα και αν αποτύχει και τότε θα πρέπει να επιλεχθεί μία διαφορετική αρχική λύση εκκίνησης. Σε περίπτωση επιτυχίας του χρονικού βηματισμού για έναν συγκεκριμένο αριθμό βημάτων, τότε η εξίσωση (27) επιχειρείται να επιλυθεί ξανά.

Η επαναληπτική διαδικασία τελειώνει όταν η λύση συγκλίνει, δηλαδή:

$$\|\Delta\phi^n\| \leq \max(A_t, R^t\|\phi^n\|) \tag{30}$$

όπου A_t είναι η απόλυτη ανοχή και R_t είναι η σχετική ανοχή, που καθορίζονται από το χρήστη.

Όσον αφορά την Ιακωβιανή του συστήματος, παρατηρούμε από την (25) ότι οι συναρτήσεις f_i της F για έναν κόμβο i εξαρτώνται από τις άγνωστες μεταβλητές ϕ_{i-1}, ϕ_i και ϕ_{i+1}, συνεπώς μόνο τα αντίστοιχα στοιχεία θα υπάρχουν στην Ιακωβιανή. Αυτό σημαίνει ότι η Ιακωβιανή είναι ένας τρι-διαγώνιος πίνακας και η εξίσωση (27) μπορεί να επιλυθεί με τη μέθοδο απαλοιφής του Gauss. Επιπλέον, η Ιακωβιανή μπορεί να υπολογιστεί αριθμητικά με πεπερασμένες διαφορές:

$$J_{i,j} \approx \frac{f_i(\phi_j + \delta) - f_i(\phi_j)}{\delta} \tag{30}$$

όπου $\delta = r\phi_j + a$ και r και a είναι οι σχετικές και απόλυτες μεταβολές, που αντιστοιχούν στο όριο στρογγυλοποίησης του ηλεκτρονικού υπολογιστή.

4.5 Μεταβατική Κατάσταση

Η μεταβατική κατάσταση είναι αυτή κατά την οποία οι μεταβλητές του συστήματος μεταβάλλονται με το χρόνο. Συνεπώς στις εξισώσεις που διέπουν το σύστημα θα πρέπει να ληφθούν υπόψη και οι χρονικές παράγωγοι των άγνωστων μεταβλητών.

Θα χρησιμοποιηθούν δύο προσεγγίσεις για την επίλυση της μεταβατικής κατάστασης, μία χρησιμοποιώντας το αρχικά διαμορφωμένο PDE, που περιγράφεται από τις εξισώσεις (20b), (21e) και (22d) και μία μετατρέποντας το PDE σε ODE και επιλύοντάς το με το Matcont. Ο λόγος επιλογής δύο προσεγγίσεων είναι για τη σύγκριση μεταξύ τους και επιβεβαίωση της ορθής μετατροπής του PDE σε ODE, διότι εν τέλει αυτό που θέλουμε είναι να εκμεταλλευτούμε το Matcont, που λειτουργεί με ODE και τις δυνατότητές του στον εντοπισμό σημείων διακλάδωσης.

4.5.1 Επίλυση Μεταβατικής Κατάστασης με PDE

Η παρατήρηση της μεταβατικής κατάστασης ενός προβλήματος είναι χρήσιμη, διότι μας βοηθά να κατανοήσουμε πως συμπεριφέρεται ένα σύστημα όταν μεταβληθεί κάποια παράμετρός του. Ενώ το σύστημα βρίσκεται σε κατάσταση ισορροπίας, αυτό που κάνουμε στην ουσία είναι να μεταβάλλουμε ελαφρώς μία παράμετρό του (π.χ. το ρυθμό ροής μάζας του καυσίμου) και παρατηρούμε πως εξελίσσεται το σύστημα.

Για τη μοντελοποίηση του συστήματος στη μεταβατική κατάσταση, θα χρησιμοποιηθεί η πεπλεγμένη (implicit) μέθοδος του Euler, η οποία είναι της μορφής:

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = F(\phi^{n+1}) \tag{31}$$

Η παραπάνω μορφή είναι συμβατή με τις εξισώσεις (21e) και (22d), όπου τα αριστερά τους μέλη είναι οι χρονικές παράγωγοι των Y_k και T και οι οποίες σε διακριτοποιημένη μορφή δίνονται από τις σχέσεις (29). Όσον αφορά το ρυθμό ροής μάζας $\dot{m}$, παρατηρούμε ότι δεν έχει χρονική παράγωγο, ούτε στις δύο προαναφερόμενες εξισώσεις, ούτε στην (20b). Η πυκνότητα, που εμφανίζεται με χρονική παράγωγο στην (20b) δεν είναι ανεξάρτητη μεταβλητή αλλά εξαρτάται από τα Y_k και T και υπολογίζεται βάσει της (5). Το γεγονός ότι υπάρχει ο όρος της πυκνότητας στην εξίσωση (20b) και ο οποίος είναι διάφορος του μηδενός στην περίπτωση της μεταβατικής κατάστασης, σημαίνει ότι ο ρυθμός ροής μάζας δεν είναι σταθερός, όπως στην περίπτωση της κατάστασης ισορροπίας, σε όλο το μήκος του πλέγματος, αλλά μεταβάλεται, δηλαδή $\dot{m}_i \neq \dot{m}_{i-1}$.

Στην παρούσα εργασία εξετάζονται δύο περιπτώσεις, μία όπου ο ρυθμός ροής μάζας θεωρείται σταθερός σε όλο το μήκος του πλέγματος, ακόμα και για το πρόβλημα μεταβατικής κατάστασης και η εξίσωση (20b) δεν λαμβάνεται υπόψη και μία όπου θεωρείται μεταβλητός με βάση τη σχέση (20b).

Το σύστημα εξισώσεων στη διακριτοποιημένη του μορφή, αντίστοιχο με αυτό της έκφρασης (25), όπου έχουν συμπεριληφθεί και οι χρονικές παράγωγοι του αριστερού μέλους των (21e) και (22d) ενώ οι εκφράσεις των δεξιών μελών έχουν μεταφερθεί και αυτές στα αριστερά ώστε το σύνολο να ισούται με μηδέν, είναι το εξής:

$$
F = \begin{bmatrix}
\dot{m}_1 - \dot{m}_0 \\
Y_{1,1} - \epsilon_1 \\
\vdots \\
Y_{K,1} - \epsilon_K \\
T_1 + T_0 \\
\dfrac{\dot{m}_2 - \dot{m}_1}{h_2 A_2} \\
\dfrac{Y_{1,2}^{(n+1)} - Y_{1,2}^{(n+1)}}{\Delta t} + \left\{ -\dfrac{\omega_{1,2}}{\rho_2} W_{1,2} + \dfrac{\dot{m}_2}{\rho_2 A_2} \left[\dfrac{h_2}{h_3(h_3+h_2)} Y_{1,3} + \dfrac{h_3 - h_2}{h_3 h_2} Y_{1,2} - \dfrac{h_3}{h_2(h_3+h_2)} Y_{1,1} \right] + \dfrac{(\rho A Y_1 V_1)_{\frac{5}{2}} - (\rho A Y_1 V_1)_{\frac{3}{2}}}{\rho_2 A_2 h_{\frac{5}{2}}} \right\}^{(n+1)} \\
\vdots \\
\dfrac{Y_{K,2}^{(n+1)} - Y_{K,2}^{(n+1)}}{\Delta t} + \left\{ -\dfrac{\omega_{K,2}}{\rho_2} W_{K,2} + \dfrac{\dot{m}_2}{\rho_2 A_2} \left[\dfrac{h_2}{h_3(h_3+h_2)} Y_{K,3} + \dfrac{h_3 - h_2}{h_3 h_2} Y_{K,2} - \dfrac{h_3}{h_2(h_3+h_2)} Y_{K,1} \right] + \dfrac{(\rho A Y_K V_K)_{\frac{5}{2}} - (\rho A Y_K V_K)_{\frac{3}{2}}}{\rho_2 A_2 h_{\frac{5}{2}}} \right\}^{(n+1)} \\
\dfrac{T_2^{(n+1)} - T_2^{(n+1)}}{\Delta t} + \dfrac{1}{\rho_2^{(n+1)} c_{p,2}^{(n+1)}} \left\{ \left[\dfrac{\dot{m}_2 c_{p,2}}{A_2} + \rho_2 \sum_{k=1}^{K} \left(Y_k V_k c_{pk} \right)_2 \right] \left[\dfrac{h_2}{h_3(h_3+h_2)} T_3 + \dfrac{h_3 - h_2}{h_3 h_2} T_2 - \dfrac{h_3}{h_2(h_3+h_2)} T_1 \right] - \right. \\
\left. \dfrac{2}{(h_3+h_2) A_2} \left[(A\lambda)_{\frac{5}{2}} \left(\dfrac{T_3 - T_2}{h_3} \right) - (A\lambda)_{\frac{3}{2}} \left(\dfrac{T_2 - T_1}{h_2} \right) \right] + \sum_{k=1}^{K} (\omega_k h_k W_k)_2 + q_2 \right\}^{(n+1)} \\
\vdots \\
\vdots \\
\dfrac{\dot{m}_{N-1} - \dot{m}_{N-2}}{h_{N-1} A_{N-1}} \\
\dfrac{Y_{1,N-1}^{(n+1)} - Y_{1,N-1}^{(n+1)}}{\Delta t} + \left\{ -\dfrac{\omega_{1,N-1}}{\rho_{N-1}} W_{1,N-1} + \dfrac{\dot{m}_{N-1}}{\rho_{N-1} A_{N-1}} \left[\dfrac{h_{N-1}}{h_N(h_N+h_{N-1})} Y_{1,N} + \dfrac{h_N - h_{N-1}}{h_N h_{N-1}} Y_{1,N-1} - \dfrac{h_N}{h_{N-1}(h_N+h_{N-1})} Y_{1,N-2} \right] + \dfrac{(\rho A Y_1 V_1)_{N-\frac{1}{2}} - (\rho A Y_1 V_1)_{N-\frac{3}{2}}}{\rho_{N-1} A_{N-1} h_{N-\frac{1}{2}}} \right\}^{(n+1)} \\
\vdots \\
\dfrac{Y_{K,N-1}^{(n+1)} - Y_{K,N-1}^{(n+1)}}{\Delta t} + \left\{ -\dfrac{\omega_{K,N-1}}{\rho_{N-1}} W_{K,N-1} + \dfrac{\dot{m}_{N-1}}{\rho_{N-1} A_{N-1}} \left[\dfrac{h_{N-1}}{h_N(h_N+h_{N-1})} Y_{K,N} + \dfrac{h_N - h_{N-1}}{h_N h_{N-1}} Y_{K,N-1} - \dfrac{h_N}{h_{N-1}(h_N+h_{N-1})} Y_{K,N-2} \right] + \dfrac{(\rho A Y_K V_K)_{N-\frac{1}{2}} - (\rho A Y_K V_K)_{N-\frac{3}{2}}}{\rho_{N-1} A_{N-1} h_{N-\frac{1}{2}}} \right\}^{(n+1)} \\
\dfrac{T_{N-1}^{(n+1)} - T_{N-1}^{(n+1)}}{\Delta t} + \dfrac{1}{\rho_{N-1}^{(n+1)} c_{p,N-1}^{(n+1)}} \left\{ \left[\dfrac{\dot{m}_{N-1} c_{p,N-1}}{A_{N-1}} + \rho_{N-1} \sum_{k=1}^{K} \left(Y_k V_k c_{pk} \right)_{N-1} \right] \left[\dfrac{h_{N-1}}{h_N(h_N+h_{N-1})} T_N + \dfrac{h_N - h_{N-1}}{h_N h_{N-1}} T_{N-1} - \dfrac{h_N}{h_2(h_N+h_{N-1})} T_{N-2} \right] - \right. \\
\left. \dfrac{2}{(h_N+h_{N-1}) A_{N-1}} \left[(A\lambda)_{N-\frac{1}{2}} \left(\dfrac{T_N - T_{N-1}}{h_N} \right) - (A\lambda)_{N-\frac{3}{2}} \left(\dfrac{T_{N-1} - T_{N-2}}{h_{N-1}} \right) \right] + \sum_{k=1}^{K} (\omega_k h_k W_k)_{N-1} + q_{N-1} \right\}^{(n+1)} \\
\dfrac{\dot{m}_N - \dot{m}_{N-1}}{h_N A_N} \\
Y_{1,N} + Y_{1,\infty} \\
\vdots \\
Y_{K,N} - Y_{K,\infty} \\
T_N - T_\infty
\end{bmatrix} \tag{32}
$$

Στο παραπάνω σύστημα οι ρυθμοί ροής μάζας έχουν θεωρηθεί σταθερές, ενώ ο άνω δείκτης n μέσα σε παρένθεση που εμφανίζεται στους όρους υποδεικνύει το χρονικό βήμα,

όπου $(n+1)$ είναι το επόμενο χρονικό βήμα, που είναι άγνωστο και (n) είναι το τελευταίο γνωστό.

Το σύστημα εξισώσεων (32) επιλύεται για κάθε χρονικό βήμα χρησιμοποιώντας την τροποποιημένη μέθοδο Newton με απόσβεση. Εφόσον βρεθεί λύση, τότε αυτή χρησιμοποιείται ως προηγούμενη για τον υπολογισμό της λύσης του επόμενου βήματος και ούτω καθεξής.

Σημειώνται ότι το σύστημα (32) είναι ίδιο με αυτό που χρησιμοποιήθηκε και στην τεχνική του χρονικού βηματισμού. Όπως προαναφέρθηκε, η λύση είναι πιο εύκολο να συγκλίνει στην περίπτωση που συμπεριλαμβάνονται στις εξισώσεις και οι χρονικές παράγωγοι. Αυτό οφείλεται στο γεγονός ότι λόγω των χρονικών παραγώγων εμφανίζονται στη διαγώνιο της Ιακωβιανής οι όροι $\frac{1}{\Delta t}$, οι οποίοι είναι πολύ μεγάλοι και εξασφαλίζουν ότι το σύστημα θα επιλυθεί.

Στην περίπτωση που θεωρήσουμε ότι ο ρυθμός ροής μάζας μεταβάλλεται με βάση την εξίσωση (20b), τότε οι εξισώσεις $f_i = \dot{m}_i - \dot{m}_{i-1}$ μετατρέπονται στις εξής:

$$\frac{\rho_i^{(n+1)} - \rho_i^{(n)}}{\Delta t} + \frac{\dot{m}_i^{(n+1)} - \dot{m}_{i-1}^{(n+1)}}{A_i h_i} = 0 \tag{33}$$

όπου ο όρος $\rho_i^{(n)}$ λαμβάνεται από τη λύση του προηγούμενου βήματος.

4.5.2 Επίλυση Μεταβατικής Κατάστασης με ODEs

Η μετατροπή του συστήματος PDEs σε ODEs αφορά στην ουσία στην απαλοιφή από το σύστημα (25) των μεταβλητών εκείνων, οι οποίοι δεν έχουν χρονικές παραγώγους, καθώς και τις αντίστοιχες εξισώσεις.

Οι μεταβλητές αυτές είναι τα Y_k και T που ανήκουν στον πρώτο και τελευταίο κόμβο, τα οποία περιορίζονται από συνοριακές συνθήκες, καθώς και οι ρυθμοί ροής μάζας $\dot{m}_i$, που δεν έχουν χρονική παράγωγο.

Εφόσον περιορίζονται από συνοριακές συνθήκες και οι τιμές τους είναι ανεξάρτητες από το ίδιο το σύστημα, μπορούν να θεωρηθούν ως παράμετροι αυτού.

Ο Πίνακας 4 περιλαμβάνει αλγοριθμική διαδικασία, που περιγράφει τους κανόνες βάσει των οποίων οι μεταβλητές αυτές αφαιρούνται από το διάνυσμα των αγνώστων και θεωρούνται παράμετροι του συστήματος.

5 Ανάπτυξη Αλγόριθμων στο Matlab

Για την επίλυση του συστήματος στις διάφορες καταστάσεις του αναπτύχθηκε κώδικας του Matlab, παρόμοιος με αυτό που χρησιμοποιείται από το πολύ γνωστό λογισμικό CHEMKIN.

Για να είναι ο κώδικας χρήσιμος σε γενικότερο επίπεδο και να μπορεί να χρησιμοποιηθεί και σε άλλα προβλήματα, πέραν από το συγκεκριμένο πρόβλημα σφαιρικών φλογών διάχυσης, αποτελείται από διακριτά ξεχωριστά τμήματα, ενώ περιλαμβάνει επιλογές όσον αφορά διάφορες παραμέτρους, π.χ. επιλογή αδιαβατικής ή ελεύθερης καύσης, επιλογή συνοριακών συνθηκών Dirichlet ή Neuman (ή συνδυασμό), επιλογή αν θα υπολογίζεται η εξίσωση ενέργειας ή όχι κλπ.

Ο κώδικας αποτελείται από τέσσερις ομάδες ρουτινών:

1. Ρουτίνες ανάγνωσης χημικών δεδομένων. Οι ρουτίνες αυτές διαβάζουν αρχεία με θερμοδυναμικές και κινητικές ιδιότητες των συστατικών, αρχεία με μηχανισμούς αντίδρασης καθώς και δεδομένα ακτινοβολίας
2. Ρουτίνες πολυωνυμικής προσέγγισης κινητικών ιδιοτήτων για πιο γρήγορο υπολογισμό τους κατά τη διάρκεια επίλυσης.
3. Ρουτίνες υπολογισμού θερμοδυναμικών και κινητικών ιδιοτήτων. Οι ρουτίνες αυτές υπολογίζουν όποτε απαιτηθεί τις θερμοδυναμικές και κινητικές ιδιότητες των συστατικών και αντιδράσεων, οι οποίες υπεισέρχονται στις εξισώσεις του συστήματος.
4. Ρουτίνες επίλυσης. Οι ρουτίνες αυτές διαβάζουν την λεπτομερή περιγραφή του προβλήματος από κατάλληλο αρχείο, μαζί με συνοριακές και αρχικές συνθήκες και επιλύουν το πρόβλημα είτε για την κατάσταση ισορροπίας είτε για τη μεταβατική κατάσταση.

5.1 Ρουτίνες Ανάγνωσης Χημικών Δεδομένων

Ο παρακάτω Πίνακας 1 παρουσιάζει τις ρουτίνες που έχουν αναπτυχθεί στα πλαίσια της εργασίας αυτής για την ανάγνωση θερμοδυναμικών και κινητικών ιδιοτήτων των συστατικών, καθώς και μηχανισμών αντιδράσεων:

Πίνακας 1 - Ρουτίνες Ανάγνωσης Χημικών Δεδομένων

#	Ρουτίνα	Περιγραφή
1	ParseChemkinInput.m	Ρουτίνα που διαβάζει τα στοιχεία, τα συστατικά, τις θερμοδυναμικές και κινητικές ιδιότητες των συστατικών καθώς και τις αντιδράσεις από αρχεία εισόδου που χρησιμοποιούνται ευρέως στη Χημική Μηχανική
2	AtomicMass.m	Υπολογίζει τις ατομικές μάζες για όλα τα στοιχεία του Περιοδικού Πίνακα των Στοιχείων
3	ParseChemkinInput_Elements.m	Ρουτίνα που διαβάζει τα χημικά στοιχεία, που απαρτίζουν το χημικό σύστημα του προβλήματος, από αρχείο εισόδου που ακολουθεί το πρότυπο του CHEMKIN. Η ρουτίνα επιστρέφει τα χημικά στοιχεία καθώς και βασικές παραμέτρους αυτών (όπως ατομικά βάρη)
4	ParseChemkinInput_Reactions.m	Ρουτίνα που διαβάζει το μηχανισμό αντιδράσεων από αρχείο εισόδου που ακολουθεί το πρότυπο του CHEMKIN. Η ρουτίνα διαβάζει τις αντιδράσεις, τους συντελεστές Arrhenius αυτής καθώς και οποιασδήποτε άλλες επιπρόσθετες πληροφορίες, όπως τρίτα σώματα, αν εξαρτάται από την πίεση, συντελεστές Troe, SRI κλπ. Εκτελεί έλεγχο ορθότητας των αντιδράσεων και επιστρέφει όλες τις απαιτούμενες πληροφορίες σε δομημένη μορφή
5	ParseChemkinInput_Species.m	Ρουτίνα που διαβάζει τα συστατικά του συστήματος από αρχείο εισόδου που ακολουθεί το πρότυπο του CHEMKIN. Η ρουτίνα επιστρέφει τα συστατικά μαζί με επιπρόσθετες πληροφορίες αυτών (π.χ. μοριακά βάρη, χημική σύσταση κλπ).
6	ParseChemkinInput_Transport.m	Ρουτίνα που διαβάζει τις κινητικές ιδιότητες των συστατικών από αρχείο εισόδου που ακολουθεί το πρότυπο του CHEMKIN
7	ParseThermo.m	Ρουτίνα που διαβάζει τις θερμοδυναμικές ιδιότητες των συστατικών από αρχείο εισόδου που ακολουθεί το πρότυπο της NASA
8	ParseThermoFromFile.m	Προέκταση της ανωτέρω ρουτίνας για ανάγνωση απευθείας από αρχείο
9	SpeciesRadiationData.m	Περιέχει δεδομένα ακτινοβολίας, που χρησιμεύουν για τον υπολογισμό των μέσων σταθερών απορρόφησης συγκεκριμένων συστατικών

5.2 Ρουτίνες Πολυωνυμικής Προσέγγισης

Ο παρακάτω Πίνακας 2 παρουσιάζει τις ρουτίνες που έχουν αναπτυχθεί στα πλαίσια της εργασίας για την πολυωνυμική προσέγγιση των κινητικών ιδιοτήτων (σταθερές διάχυσης, βαθμοί θερμικής διάχυσης, συντελεστές θερμικής αγωγιμότητας και ιξώδους), προκειμένου

να είναι πιο γρήγορος ο υπολογισμός τους κατά τη διάρκεια επίλυσης του προβλήματος, καθόσον οι αναλυτικές μέθοδοι υπολογισμού τους είναι πολύ χρονοβόρες.

Πίνακας 2 - Ρουτίνες Πολυωνυμικής Προσέγγισης

#	Ρουτίνα	Περιγραφή
1	CalculateTransportFits.m	Ρουτίνα που υπολογίζει τους συντελεστές πολυωνυμικής προσέγγισης για όλες τις κινητικές ιδιότητες
2	CalculateBinaryDiffusion CoefficientFits.m	Υπολογίζει τους πολυωνυμικούς συντελεστές προσέγγισης $d_{n,jk}$ των δυαδικών σταθερών διάχυσης $\mathcal{D}_{jk}$ με βάση τη σχέση (7f). Ο κώδικας παρατίθεται στο Παράρτημα ΣΤ Πίνακας 21
3	CalculateThermalDiffusion RatioFits.m	Υπολογίζει τους συντελεστές πολυωνυμικής προσέγγισης $c_{n,kj}$ των ειδικών βαθμών θερμικής διάχυσης θ_{kj}^* για όλους τους συνδυασμούς k, j των συστατικών. Ο κώδικας παρατίθεται στο Παράρτημα ΣΤ Πίνακας 22
4	CalculateThermalConductivity andViscosityFits.m	Υπολογίζει τους συντελεστές πολυωνυμικής προσέγγισης των θερμικών αγωγιμοτήτων λ_k και ιξωδών η_k των συστατικών της αντίδρασης. Ο κώδικας παρατίθεται στο Παράρτημα ΣΤ Πίνακας 23
5	CalculateBinaryDiffusion CoefficientD12.m	Υπολογίζει την προσέγγιση δεύτερης τάξης Δ_{jk} των δυαδικών σταθερών διάχυσης $\mathcal{D}_{jk}$ με βάση τη σχέση (7e). Ο κώδικας παρατίθεται στο Παράρτημα ΣΤ Πίνακας 24
6	CalculateOmega12.m	Υπολογίζει τα ολοκληρώματα σύγκρουσης $\Omega^{(1,1)*}$ με βάση τα δυναμικά του Stockmayer. Ο αλγόριθμος αποτελεί μεταφορά του αντίστοιχου αλγόριθμου του CHEMKIN. Ο κώδικας παρατίθεται στο Παράρτημα ΣΤ Πίνακας 25
7	CalculateOmegaXX.m	Υπολογίζει τα ολοκληρώματα σύγκρουσης $\Omega_{11}^{(2,2)*}$, $\Omega_{12}^{(1,1)*}$ και τους λόγους ολοκληρωμάτων σύγκρουσης A_{12}^*, B_{12}^* και C_{12}^*, τα οποία χρησιμοποιούνται για τον υπολογισμό των παραμέτρων P και Q της σχέσης (7e). Ο κώδικας παρατίθεται στο Παράρτημα ΣΤ Πίνακας 26
8	CalculateRotationalCollision Numbers.m	Εκτελεί υπολογισμό των όρων οι A_{kj}^*, B_{kj}^* και C_{kj}^* που αφορούν ολοκληρώματα συγκρούσεων περιστροφής που χρησιμοποιούνται στη σχέση (8c) για τον υπολογισμό των βαθμών θερμικής διάχυσης Θ_k. Ο κώδικας παρατίθεται στο Παράρτημα ΣΤ Πίνακας 27
9	CalculateTaylorCoefficients.m	Υπολογίζει τους συντελεστές Taylor μίας μεταβλητής
10	Horner.m	Υπολογίζει μία πολυωνυμική συνάρτηση με τη μέθοδο Horner
11	PolynomialFitEvaluate.m	Ρουτίνα που χρησιμεύει στην εκτίμηση των συντελεστών πολυωνυμικής προσέγγισης
12	PolynomialsFit.m	Υπολογίζει τους συντελεστές πολυωνυμικής προσέγγισης μίας συνάρτησης
13	QuadraticInterpolation.m	Υπολογισμός ενδιάμεσων τιμών μίας συνάρτησης με πολυωνυμική παρεμβολή

Οι σημαντικότητες από τις παραπάνω ρουτίνες παρατίθενται στο Παράρτημα ΣΤ.

5.3 Ρουτίνες Υπολογισμού Θερμοδυναμικών και Κινητικών Ιδιοτήτων

Ο παρακάτω Πίνακας 3 παρουσιάζει τις ρουτίνες που έχουν αναπτυχθεί στα πλαίσια της εργασίας για τον υπολογισμό των θερμοδυναμικών και κινητικών ιδιοτήτων των συστατικών, κάθε φορά που αλλάζει η λύση, δηλαδή οι μεταβλητές Y_k και T, ή το πλέγμα ή όποτε το επιβάλει ο χρήστης. Οι ιδιότητες υπολογίζονται για κάθε κόμβο του πλέγματος και σε ορισμένες περιπτώσεις υπολογίζονται και οι ιδιότητες ενδιάμεσα των κόμβων.

Πίνακας 3 - Ρουτίνες Υπολογισμού Θερμοδυναμικών και Κινητικών Ιδιοτήτων

#	Ρουτίνα	Περιγραφή
1	RecalculateChemical Parameters.m	Επανυπολογίζει όλες τις θερμοδυναμικές και κινητικές ιδιότητες των συστατικών του συστήματος για όλο το πλέγμα
2	RecalculateChemical ParametersForSingleT.m	Επανυπολογίζει τις θερμοδυναμικές και κινητικές ιδιότητες των συστατικών του συστήματος που επηρεάζονται από την τιμή της θερμοκρασίας ενός συγκεκριμένου κόμβου
3	RecalculateChemical ParametersForSingleY.m	Επανυπολογίζει τις θερμοδυναμικές και κινητικές ιδιότητες των συστατικών του συστήματος που επηρεάζονται από την τιμή του κλάσματος μάζας ενός συγκεκριμένου συστατικού και ενός συγκεκριμένου κόμβου
4	CalculateArea.m	Υπολογίζει την επιφάνεια του μετώπου της καύσης σε συνάρτηση με τη σφαιρική συντεταγμένη για κάθε κόμβο. Στην περίπτωση που γίνεται καύση σε θάλαμο σταθερής διατομής, τότε η επιφάνεια ισούται με ένα
5	GetWMean.m	Υπολογίζει το μέσο μοριακό βάρος $\overline{W}$ του μίγματος για κάθε κόμβο αλλά και ενδιάμεσα αυτών από τη σχέση $\overline{W}^{-1} = \sum_{k=1}^{K} \frac{Y_k}{W_k}$
6	GetM.m	Υπολογίζει τη συγκέντρωση του μίγματος σε κάθε κόμβο και ενδιάμεσα αυτών, με βάση τη σχέση $M = {}^{p}/_{RT}$
7	GetDensity.m	Υπολογίζει την πυκνότητα του μίγματος για κάθε κόμβο με βάση τη σχέση (5)
8	GetXk.m	Υπολογίζει τα μοριακά κλάσματα X_k για κάθε συστατικό και κόμβο αλλά και ενδιάμεσα των κόμβων με βάση τη σχέση $X_k = \overline{W} \frac{Y_k}{W_k}$
9	GetXCk.m	Υπολογίζει τις μοριακές συγκεντρώσεις $[X_k]$ για κάθε συστατικό και κόμβο αλλά και ενδιάμεσα των κόμβων με βάση τη σχέση $[X_k] = \rho \frac{Y_k}{W_k}$
10	GetDk.m	Υπολογίζει τις σταθερές διάχυσης D_{km} για κάθε συστατικό και κόμβο από τη σχέση (7b), καθώς και ενδιάμεσα στους κόμβους
11	GetTHk.m	Υπολογίζει το βαθμό θερμικής διάχυσης Θ_k για κάθε συστατικό και κάθε κόμβο και ενδιάμεσα αυτών
12	Getvkfac.m	Υπολογίζει την ποσότητα $-\left(\frac{W_k D_{km}}{\overline{W}}\right)_j$ για κάθε συστατικό και κόμβο, καθώς και ενδιάμεσα αυτών, η οποία χρησιμοποιείται για τον υπολογισμό των ποσοτήτων $(Y_k v_k)_j$ σύμφωνα με τη σχέση (21c)
13	Getwkfac.m	Υπολογίζει την ποσότητα $-\left(\frac{W_k D_{km} \Theta_k}{\overline{W} T}\right)_j$ για κάθε συστατικό και κόμβο, καθώς και ενδιάμεσα αυτών, η οποία χρησιμοποιείται για τον υπολογισμό των ποσοτήτων $(Y_k w_k)_j$ σύμφωνα με τη σχέση (21d)

14	GetYVk.m	Υπολογίζει την ποσότητα $Y_k V_k$ από τη σχέση (6), για κάθε συστατικό και κόμβο, καθώς και ενδιάμεσα αυτών. Ο χρήστης έχει τη δυνατότητα αν θέλει να συμπεριλάβει τις θερμικές ταχύτητες διάχυσης στον υπολογισμό ή όχι
15	Getlk.m	Υπολογίζει τις θερμικές αγωγιμότητες λ_k για κάθε συστατικό και κόμβο, καθώς και ενδιάμεσα αυτών από τη σχέση (17c)
16	GetCpHSk.m	Υπολογίζει τις ειδικές θερμότητες ${C_p}/{R}$, ενθαλπίες ${H}/{RT}$ και εντροπίες S/R για κάθε συστατικό και κόμβο με βάση τις σχέσεις (14), (13f) και (13g)
17	GetEMi.m	Υπολογίζει τη δραστική συγκέντρωση $M_e = \sum_{k=1}^{K} a_k [X_k]$ κάθε αντίδρασης, η οποία χρησιμοποιείται στον υπολογισμό της εντατικής ταχύτητας της αντίδρασης όταν υπάρχει τρίτο σώμα, όπου $[X_k]$ είναι η μοριακή συγκέντρωση κάθε συστατικού και a_k ο συντελεστής συνεισφοράς του στην αντίδραση, ο οποίος δίνεται από το μηχανισμό της αντίδρασης. Η δραστική συγκέντρωση υπολογίζεται για κάθε κόμβο
18	GetKfi.m	Υπολογίζει τους εμπρόσθιους συντελεστές ταχύτητας k_f κάθε αντίδρασης σε κάθε κόμβο, από τον τύπο Arrhenius της σχέσης (12), ή από άλλες σχέσεις (π.χ. Troe, SRI, JAN, FIT1) ανάλογα τον τύπο της αντίδρασης. Ο κώδικας παρατίθεται στο Παράρτημα ΣΤ Πίνακας 28
19	GetKri.m	Υπολογίζει τους αντίστροφους συντελεστές ταχύτητας k_r κάθε αντίδρασης σε κάθε κόμβο από τις σχέσεις (13a) έως (13g). Ο κώδικας παρατίθεται στο Παράρτημα ΣΤ Πίνακας 29
20	GetQi.m	Υπολογίζει τις εντατικές ταχύτητες q_i κάθε αντίδρασης σε κάθε κόμβο από τη σχέση (11) ή στην περίπτωση που υπάρχει τρίτο σώμα λαμβάνει υπόψη και τη δραστική συγκέντρωση M_e
21	Getwdotk.m	Υπολογίζει τους ρυθμούς παραγωγής ω_k για κάθε συστατικό και κόμβο σύμφωνα με τη σχέση (10)
22	Getqri.m	Υπολογίζει τις θερμικές απώλειες ακτινοβολίας q για κάθε κόμβο από τη σχέση (18a). Ο κώδικας παρατίθεται στο Παράρτημα ΣΤ Πίνακας 30

Οι πιο χαρακτηριστικές από τις παραπάνω ρουτίνες παρατίθενται στο Παράρτημα ΣΤ.

5.4 Ρουτίνες Επίλυσης

Οι ρουτίνες αυτές αρχικοποιούν το πρόβλημα και βασικές παραμέτρους και σταθερές, διαβάζουν την λεπτομερή περιγραφή του προβλήματος από κατάλληλο αρχείο, μαζί με συνοριακές, υπολογίζουν μία αρχική υποθετική λύση εκκίνησης, που είναι απαραίτητη για να ξεκινήσει η επίλυση και επιλύουν το πρόβλημα είτε για την κατάσταση ισορροπίας είτε για τη μεταβατική κατάσταση. Οι ρουτίνες αυτές παρατίθενται ως Πίνακας 4.

Πίνακας 4 - Ρουτίνες Επίλυσης

#	Ρουτίνα	Περιγραφή

1	ProblemSolver.m	Ρουτίνα που επιλύει το σύστημα με βάση τις παραμέτρους που έχει ορίσει ο χρήστης. Τα αποτελέσματα της επίλυσης αποθηκεύονται σε αρχεία μορφής Matlab, έτσι ώστε να είναι δυνατή κατόπιν η εκκίνηση ενός νέου προβλήματος από κάποιο προηγούμενο. Επίσης ο αρχικός υπολογισμός τον πολυωνυμικών προσεγγίσεων των κινητικών ιδιοτήτων δεν απαιτείται να γίνεται κάθε φορά που ξαναξεκινάει το πρόβλημα, αλλά μόνο κατά την πρώτη φορά.
2	InitializeConstants.m	Αρχικοποιεί ορισμένες σταθερές, οι οποίες χρησιμοποιούνται στους υπολογισμούς, π.χ. P_{atm}, R, k_B κλπ ενώ επίσης αναθέτει αρχικές τιμές σε όλες τις παραμέτρους του προβλήματος
3	InitializeGlobals.m	Ρουτίνα που αρχικοποιεί τις μεταβλητές του Matlab, οι οποίες έχουν παγκόσμιο χαρακτήρα (global scope), δηλαδή είναι προσβάσιμες σε κάθε συνάρτηση χωρίς να περνούν ως παράμετροι της συνάρτησης. Η τεχνική των global variables χρησιμοποιείται σε μεταβλητές μεγάλου μεγέθους (π.χ. ο πίνακας της Ιακωβιανής) για την επιτάχυνση των υπολογισμών, λόγω του τρόπο με τον οποίο το Matlab μεταχειρίζεται τις μεταβλητές
4	SetProblemParameters.m	Ρουτίνα που διαβάζει αρχείο του χρήστη με τις βασικές παραμέτρους του συστήματος, όπως περιγράφονται με λεπτομέρεια παρακάτω καθώς και συνοριακές συνθήκες
5	ProduceInitialGuess.m	Ρουτίνα που υπολογίζει μία αρχική λύση εκκίνησης ϕ^0, εφόσον απαιτείται, έτσι ώστε να ξεκινήσει η επαναλητική διαδικασία επίλυσης
6	SolveSteadyStateSystem.m	Ρουτίνα η οποία επιλύει το σύστημα για την κατάσταση ισορροπίας, αποφασίζοντας πότε θα χρησιμοποιείται η Newton, πότε ο χρονικός βηματισμός και πότε θα γίνεται αναθεώρηση των κόμβων του πλέγματος. Το ροϊκό διάγραμμα της διαδικασίας αυτής παρατίθεται παρακάτω.
7	SolveTransientSystem.m	Ρουτίνα που επιλύει το σύστημα στη μεταβατική του κατάσταση χρησιμοποιώντας τον αλγόριθμο χρονικού βηματισμού.
8	SolveNewton.m	Ρουτίνα που επιλύει το σύστημα με την τροποποιημένη επαναληπτική μέθοδο Newton με απόσβεση. Λεπτομερής περιγραφή του τρόπου με τον οποίο λειτουργεί η ρουτίνα παρατίθεται στο Διάγραμμα 2.
9	TimeStepper.m	Ρουτίνα που επιλύει το σύστημα με χρονικό βηματισμό. Λεπτομερής περιγραφή του τρόπου με τον οποίο λειτουργεί η ρουτίνα παρατίθεται στο Διάγραμμα 3.
10	RefineGrid.m	Ρουτίνα που ελέγχει στη λύση που προέκυψε αν υπάρχουν μεταβλητές που παρουσιάζουν μεγάλες κλίσεις σε ορισμένους κόμβους και με βάση κάποια κριτήρια δημιουργεί επιπλέον κόμβους πυκνώνοντας το πλέγμα εκεί που απαιτείται
11	SteadyStateFunction.m	Βασική ρουτίνα, η οποία περιλαμβάνει τις εξισώσεις του συστήματος και τις υπολογίζει για κάθε λύση ϕ^n. Η ρουτίνα αυτή είναι γενική και καλύπτει όλες τις περιπτώσεις, κατάστασης ισορροπίας ή μεταβατικής κατάστασης, του συστήματος PDE ή του συστήματος ODE, με ή χωρίς χρονικές παραγώγους, με διάφορες επιλογές συνοριακών συνθηκών (Dirichlet, Neuman ή συνδυασμό), για αδιαβατική ή ελεύθερη ροή κλπ. Λεπτομερής περιγραφή παρατίθεται παρακάτω.
12	CalculateJacobian.m	Υπολογίζει την Ιακωβιανή $J_{i,j}$ με πεπερασμένες διαφορές σύμφωνα με τη σχέση (30)

13	FindMaxDumping Coefficient.m	Υπολογίζει τη μέγιστη παράμετρο απόσβεσης λ^n για την οποία η λύση που προκύπτει από την τροποποιημένη μέθοδο Newton με απόσβεση σε κάποιο βήμα n και με βάση τη σχέση (27) έχει φυσικό νόημα, δηλαδή όλες οι μεταβλητές βρίσκονται εντός φυσικών ορίων σύμφωνα με τα κριτήρια (28b)
14	HasConverged.m	Ελέγχει αν η τρέχουσα λύση $\Delta\phi^n = \phi^{n+1} - \phi^n$ έχει συγκλίνει, δηλαδή ικανοποιεί τη σχέση (30) για κάθε μεταβλητή
15	IsSolutionAcceptable.m	Ελέγχει αν η λύση ϕ^n έχει φυσικό νόημα, δηλαδή αν ικανοποιεί τα κριτήρια (28b)
16	MatcontFunction.m	Ρουτίνα η οποία μετατρέπει το σύστημα εξισώσεων PDE σε ODE, αφαιρώντας από το διάνυσμα άγνωστων μεταβλητών αυτές που δεν έχουν χρονικές παραγώγους καθώς και τις αντίστοιχες εξισώσεις, έτσι ώστε το σύστημα να μπορεί να χρησιμοποιηθεί από το Matcont
17	SolveBoundaryFluxes.m	Κατά την επίλυση του συστήματος ODE από το Matcont δεν είναι δυνατή η επιβολή συνοριακών συνθηκών μέσω των εξισώσεων, έτσι ώστε οι μεταβλητές που περιορίζονται από συνοριακές συνθήκες να υπολογίζονται αυτόματα κατά την επίλυση, καθόσον δεν υπάρχουν χωρικές παράγωγοι. Συνεπώς οι συνοριακές συνθήκες πρέπει να υπολογίζονται ξεχωριστά κατόπιν κάθε βήματος επίλυσης. Αυτή η ρουτίνα υπολογίζει τις συνοριακές συνθήκες Dirichlet+Neuman στο αριστερό όριο (είσοδος καυσίμου) για τα κλάσματα μάζας των συστατικών Y_k

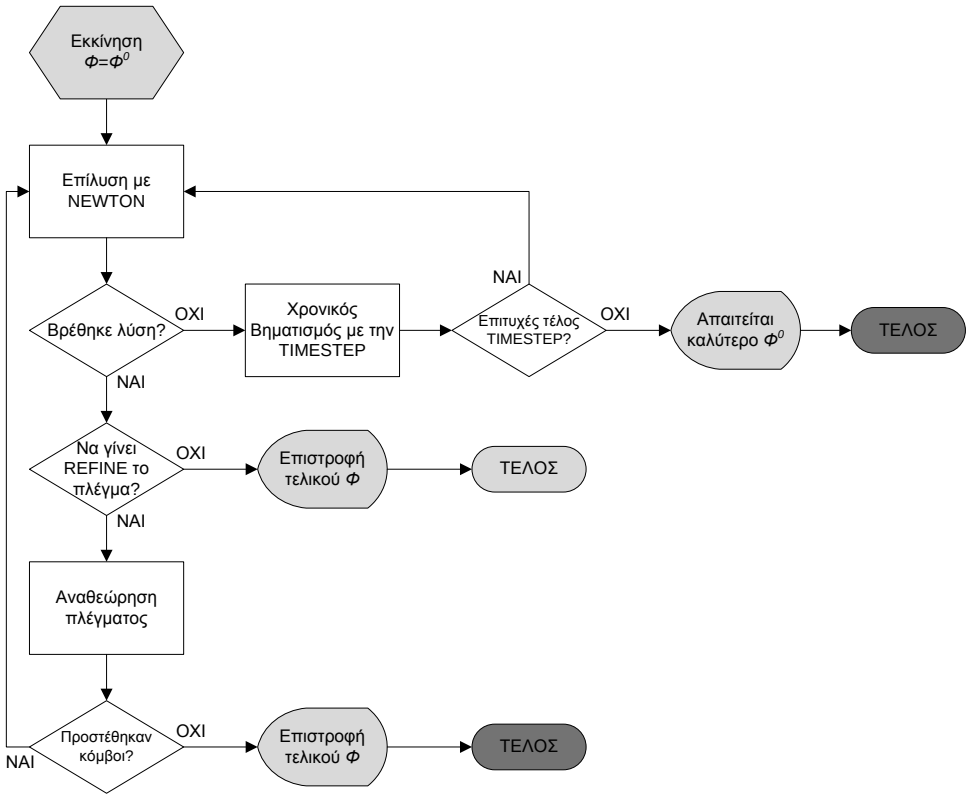
5.5 Διαθέσιμες Επιλογές του Αλγορίθμου

Παρότι οι προαναφερόμενες ρουτίνες κατασκευάστηκαν στα πλαίσια επίλυσης συγκεκριμένου προβλήματος φλογών διάχυσης, εντούτοις διατίθενται πολλές επιλογές, μέσω των οποίων ο χρήστης μπορεί να κατασκευάσει ένα πρόβλημα φλογών διάχυσης με ποικίλες συνθήκες. Οι επιλογές αυτές, που ορίζονται από αρχείο εισόδου παρουσιάζονται στο Παράρτημα Α, Πίνακας 15.

5.6 Ροϊκά Διαγράμματα Ρουτινών

Τα ροϊκά διαγράμματα, τα οποία περιγράφουν τον τρόπο λειτουργίας των δύο βασικών ρουτινών επίλυσης, της τροποποιημένης Newton με απόσβεση και του χρονικού βηματισμού, παρατίθεται στο Παράρτημα Β.

Στο Διάγραμμα 1 παρατίθεται το κεντρικό ροϊκό διάγραμμα που επιλύει το πρόβλημα στην κατάσταση ισορροπίας, χρησιμοποιώντας είτε Newton είτε χρονικό βηματισμό, ενώ περιλαμβάνει και αναθεώρηση των κόμβων:



Διάγραμμα 1 – Ροϊκό διάγραμμα αλγορίθμου επίλυσης κατάστασης ισορροπίας

5.7 Επιλογές Συστήματος Εξισώσεων

Το διακριτοποιημένο και μοντελοποιημένο σύστημα εξισώσεων του προβλήματος με τη ρουτίνα SteadyStateFunction.m χρησιμοποιεί τις παρακάτω εξισώσεις, ανάλογα με τις παραμέτρους του προβλήματος, ως Πίνακας 5:

Πίνακας 5 – Επιλογές Συστήματος Μοντελοποιημένων Εξισώσεων

Μετ	Συνθήκη	Εξίσωση
$\dot{m}_1$	adiab = true	$f_j = \dot{m}_2 - \dot{m}_1$
	adiab = false	$f_j = \dot{m}_1 - \dot{m}_0$
$Y_{k,1}$	keepflux = true	$f_j = \varepsilon_k - Y_k - \left(\frac{\rho A Y_k V_k}{\dot{m}_1}\right)_{1\frac{1}{2}}$
	keepflux = false	$f_j = Y_k - \varepsilon_k$
T_1		$f_j = T_1 - T_0$
$\dot{m}_i$	adiab = true	$f_j = \dot{m}_{i+1} - \dot{m}_i \quad i > i_{fixT}$ $f_j = \dot{m}_i - \dot{m}_{i-1} \quad i \leq i_{fixT}$ $f_j = T_i - T_{fixT} \quad i = i_{fixT}$
	adiab = false	$f_j = \dot{m}_i - \dot{m}_{i-1}$

$Y_{k,i}$	cendif = true timestep = false	$f_i = -\frac{\omega_{k,i}}{\rho_i} W_{k,i} + \frac{\dot{m}_i}{A_i \rho_i}\left[\frac{h_i}{h_{i+1}(h_{i+1}+h_i)} Y_{k,i+1} + \frac{h_{i+1}-h_i}{h_{i+1}h_i} Y_{k,i} - \frac{h_{i+1}}{h_i(h_{i+1}+h_i)} Y_{k,i-1}\right] + \frac{(A\rho Y_k V_k)_{i+\frac{1}{2}} - (A\rho Y_k V_k)_{i-\frac{1}{2}}}{A_i \rho_i h_{i+\frac{1}{2}}}$
	cendif = false timestep = false	$f_i = -\frac{\omega_{k,i}}{\rho_i} W_{k,i} + \frac{\dot{m}_i}{A_i \rho_i h_i}\left[Y_{k,i} - Y_{k,i-1}\right] + \frac{(A\rho Y_k V_k)_{i+\frac{1}{2}} - (A\rho Y_k V_k)_{i-\frac{1}{2}}}{A_i \rho_i h_{i+\frac{1}{2}}}$
	timestep = true	$f_i = f_i + \frac{\left[Y_{k,i} - Y_{k,i}^{(prev)}\right]}{\Delta t}$
T_i	enrg = true cendif = true timestep = false	$f_i = \frac{1}{\rho_i c_{p,i}}\left\{\left[\frac{\dot{m}_i c_{p,i}}{A_i} + \rho_i \sum_{k=1}^{K}\left(Y_k V_k c_{pk}\right)_i\right]\left[\frac{h_i}{h_{i+1}(h_{i+1}+h_i)} T_{i+1} + \frac{h_{i+1}-h_i}{h_{i+1}h_i} T_i - \frac{h_{i+1}}{h_i(h_{i+1}+h_i)} T_{i-1}\right] - \frac{2}{A_i(h_{i+1}+h_i)}\left[(A\lambda)_{i+\frac{1}{2}}\left(\frac{T_{i+1}-T_i}{h_{i+1}}\right) - (A\lambda)_{i-\frac{1}{2}}\left(\frac{T_i - T_{i-1}}{h_i}\right)\right] + \sum_{k=1}^{K}(\omega_k h_k W_k)_i + q_i\right\}$
	enrg = true cendif = false timestep = false	$f_i = \frac{1}{\rho_i c_{p,i}}\left\{\frac{\dot{m}_i c_{p,i}}{A_i h_i}(T_i - T_{i-1}) + \rho_i \sum_{k=1}^{K}\left(Y_k V_k c_{pk}\right)_i\left[\frac{h_i}{h_{i+1}(h_{i+1}+h_i)} T_{i+1} + \frac{h_{i+1}-h_i}{h_{i+1}h_i} T_i - \frac{h_{i+1}}{h_i(h_{i+1}+h_i)} T_{i-1}\right] - \frac{2}{A_i(h_{i+1}+h_i)}\left[(A\lambda)_{i+\frac{1}{2}}\left(\frac{T_{i+1}-T_i}{h_{i+1}}\right) - (A\lambda)_{i-\frac{1}{2}}\left(\frac{T_i - T_{i-1}}{h_i}\right)\right] + \sum_{k=1}^{K}(\omega_k h_k W_k)_i + q_i\right\}$
	enrg = false timestep = false	$f_j = T_i - T_{fix,i}$
	timestep = true	$f_i = f_i + \frac{\left[T_i - T_i^{(prev)}\right]}{\Delta t}$
$\dot{m}_N$		$f_j = \dot{m}_N - \dot{m}_{N-1}$
$Y_{k,N}$	gradvan = true	$f_j = Y_{k,N} - Y_{k,N-1}$
	gradvan = false	$f_j = Y_{k,N} - Y_\infty$
T_N	enrg = true gradvan = true	$f_j = T_N - T_{N-1}$
	enrg = false ή gradvan = false	$f_j = T_N - T_\infty$

5.8 Πιστοποίηση Αλγόριθμων

Βασικό στοιχείο πριν την χρήση των ρουτινών που αναπτύχθηκαν στο Matlab είναι η πιστοποίησή τους, λόγω της μεγάλης πολυπλοκότητάς τους και την πιθανότητας σφάλματος.

Η πιστοποίηση γίνεται σε δύο επίπεδα, ένα που επιβεβαιώνει αρχικά την ορθότητα των ρουτινών που διαβάζουν τα αρχεία με τις θερμοδυναμικές και κινητικές ιδιότητες των συστατικών και υπολογίζουν τις πολυωνυμικές προσεγγίσεις τους και το δεύτερο επίπεδο που επιβεβαιώνει τις ρουτίνες συνολικά, συγκρίνοντας την παραγόμενη λύση ενός προβλήματος με αυτή που παράγει το CHEMKIN.

5.8.1 Πιστοποίηση Πρώτου Επιπέδου

Για την πιστοποίηση πρώτου επιπέδου χρησιμοποιήθηκε το πρόβλημα καύσης προαναμεμιγμένης φλόγας (premixed flame) υδρογόνου-οξυγόνου που υπάρχει στο εγχειρίδιο του PREMIX. Το PREMIX (Kee, et al., 1998) είναι ένα πρόγραμμα γραμμένο σε FORTRAN που επιλύει προβλήματα φλογών διάχυσης με χρήση του CHEMKIN. Το εν λόγω πρόγραμμα είναι διαθέσιμο και έτσι μπορεί να γίνει αντιπαραβολή αποτελεσμάτων με αυτό.

Το συγκεκριμένο πρόβλημα αφορά καύση υδρογόνου με οξυγόνο και αργό, όπου το καύσιμο και το οξειδωτικό έχουν προαναμιχθεί. Τα αντιδρώντα, πριν αρχίσει η καύση έχουν μοριακά κλάσματα ως εξής: H_2=0.28; O_2=0.09; Ar=0.63. Η πίεση είναι 0.0329 atm και ο ρυθμός ροής μάζας που περνά από το θάλαμο καύσης είναι 4.63 x 10-3 gm/cm2-sec.

Ο μηχανισμός καύσης, από τον οποίο προκύπτουν τα συστατικά, φαίνεται ως ο ακόλουθος Πίνακας 6:

Πίνακας 6 – Μηχανισμός Καύσης Προαναμεμιγμένης Φλόγας H_2-O_2

```
ELEMENTS
H  O  AR
END
SPECIES
H2 O2 H O OH HO2 H2O2 H2O AR
END
REACTIONS
H2+O2=2OH                                  1.7E13    0.0    47780.
OH+H2=H2O+H      1.17E9    1.3     3626. !D-L$W

H+O2=OH+O        5.13E16  -0.816  16507. !JAM,JCP 1981

O+H2=OH+H                                  1.8E10    1.0     8826.
H+O2+M=HO2+M     2.1E18    -1.0       0. !SLACK

   H2O/21./   H2/3.3/  O2/0.0/

H+O2+O2=HO2+O2   6.7E19   -1.42       0. !SLACK,JAN
```

```
OH+HO2=H2O+O2                                5.0E13    0.0     1000.
H+HO2=2OH                                    2.5E14    0.0     1900.
O+HO2=O2+OH                                  4.8E13    0.0     1000.
2OH=O+H2O        6.0E+8    1.3        0. !COHEN-WEST.
H2+M=H+H+M                                   2.23E12   0.5    92600.
  H2O/6/   H/2/  H2/3/
O2+M=O+O+M                                   1.85E11   0.5    95560.
H+OH+M=H2O+M                                 7.5E23   -2.6        0.
  H2O/20/
H+HO2=H2+O2                                  2.5E13    0.0      700.
HO2+HO2=H2O2+O2                              2.0E12    0.0        0.
H2O2+M=OH+OH+M                               1.3E17    0.0    45500.
H2O2+H=HO2+H2                                1.6E12    0.0     3800.
H2O2+OH=H2O+HO2                              1.0E13    0.0     1800.
END
```

Το CHEMKIN διαβάζει τα παραπάνω δεδομένα από αρχείο, καθώς επίσης και τις θερμοδυναμικές και κινητικές ιδιότητες από πίνακες και με βάση αυτά τα στοιχεία κατασκευάζει τους πίνακες πολυωνυμικών προσεγγίσεων με τη χρήση του πακέτου TRANSPORT. Κατόπιν εκτέλεσης του προγράμματων με τα ανωτέρω στοιχεία, προέκυψε το αρχείο tran.out, του οποίου τα δεδομένα παρατίθενται στο Παράρτημα Γ, Πίνακας 16.

Για την επιβεβαίωση των ρουτινών του Matlab κατασκευάστηκε η ρουτίνα TestChemkinInput2.m, που περιλαμβάνεται ως Πίνακας 17 του Παραρτήματος Δ και η οποία χρησιμοποιώντας τις ρουτίνες αυτές διαβάζει τα ίδια αρχεία, όπως και το CHEMKIN, δημιουργεί τους πολυωνυμικούς συντελεστές και μετά δημιουργεί ένα αρχείο tran2.out με την ίδια μορφή όπως και το tran.out, προκειμένου να συγκρίνει τα περιεχόμενά τους.

Κατόπιν εκτέλεσης του TestChemkinInput2.m το αρχείο tran2.out που δημιουργήθηκε είναι πανομοιότυπο με το tran.out, που σημαίνει ότι οι ρουτίνες του Matlab παράγουν τα ίδια αποτελέσματα, όπως και το CHEMKIN, όσον αφορά τον υπολογισμό των πολυωνυμικών συντελεστών των κινητικών ιδιοτήτων.

5.8.2 Πιστοποίηση Δεύτερου Επιπέδου

Για την πιστοποίηση δεύτερου επιπέδου θα πρέπει να γίνει σύγκριση αποτελεσμάτων επίλυσης ολόκληρου προβλήματος. Χρησιμοποιείται και πάλι το πρόβλημα καύσης υδρογόνου με οξυγόνο και αργό του PREMIX αλλά αυτή τη φορά επιλύεται ολόκληρο, μέχρι να καταλήξει σε συγκεκριμένη λύση.

Το σύστημα αρχικά έχει 6 κόμβους και κατόπιν επίλυσής του με το PREMIX παράγει την εξής λύση, ως Πίνακας 7:

Πίνακας 7 – Αρχική Λύση PREMIX Προβλήματος Καύσης H_2-O_2

	X	T	V	RHO	H2	O2	H	O	OH	HO2	H2O2
1	.0000	3.737E+02	1.496E+02	3.095E-05	2.412E-01	8.539E-02	2.495E-02	2.454E-04	1.926E-04	5.314E-09	1.490E-08
2	2.0000	1.260E+03	4.940E+02	9.373E-06	8.645E-02	4.971E-03	8.654E-02	2.642E-03	2.112E-03	8.830E-08	2.494E-07
3	4.0000	1.390E+03	5.414E+02	8.553E-06	8.882E-02	2.170E-03	7.592E-02	2.248E-03	2.859E-03	3.127E-08	2.788E-07
4	6.0000	1.475E+03	5.720E+02	8.095E-06	9.284E-02	1.929E-03	6.822E-02	2.174E-03	3.373E-03	2.423E-08	2.596E-07
5	8.0000	1.500E+03	5.800E+02	7.983E-06	9.575E-02	1.627E-03	6.298E-02	1.889E-03	3.310E-03	1.985E-08	2.256E-07
6	10.0000	1.524E+03	5.895E+02	7.854E-06	9.575E-02	1.627E-03	6.298E-02	1.889E-03	3.310E-03	1.985E-08	2.256E-07

	X	H2O	AR
1	.0000	1.329E-02	6.347E-01
2	2.0000	1.684E-01	6.489E-01
3	4.0000	1.745E-01	6.534E-01
4	6.0000	1.753E-01	6.561E-01
5	8.0000	1.767E-01	6.578E-01
6	10.0000	1.767E-01	6.578E-01

Για να μπορέσει το PREMIX να επιλύσει το πρόβλημα μεσολάβησαν δύο εκτελέσεις χρονικού βηματισμού, πριν καταφέρει η Newton να συγκλίνει. Η επίλυση συνεχίζεται με αναθεώρηση των κόμβων, μέχρι οι μεταβολές των μεταβλητών του συστήματος να είναι κάτω από τα προκαθορισμένα όρια. Το σύστημα σταματάει στους 32 κόμβους και η λύση που προκύπτει είναι η παρακάτω, ως Πίνακας 8:

Πίνακας 8 – Τελική Λύση PREMIX Προβλήματος Καύσης H2-O2

	X	T	V	RHO	H2	O2	H	O	OH	HO2	H2O2
1	.0000	3.737E+02	1.488E+02	3.112E-05	2.683E-01	9.115E-02	3.913E-04	3.289E-08	3.681E-05	1.565E-05	6.797E-07
2	.0156	3.875E+02	1.542E+02	3.003E-05	2.676E-01	9.116E-02	4.049E-04	3.711E-08	4.163E-05	1.886E-05	8.202E-07
3	.0313	4.014E+02	1.596E+02	2.902E-05	2.669E-01	9.117E-02	4.812E-04	7.197E-08	4.653E-05	1.994E-05	9.383E-07
4	.0469	4.153E+02	1.649E+02	2.807E-05	2.660E-01	9.117E-02	6.058E-04	1.486E-07	5.138E-05	1.969E-05	1.034E-06
5	.0625	4.291E+02	1.703E+02	2.719E-05	2.652E-01	9.116E-02	7.742E-04	3.006E-07	5.594E-05	1.865E-05	1.108E-06
6	.0938	4.568E+02	1.810E+02	2.557E-05	2.633E-01	9.112E-02	1.235E-03	1.001E-06	6.352E-05	1.553E-05	1.201E-06
7	.1250	4.845E+02	1.917E+02	2.415E-05	2.612E-01	9.104E-02	1.857E-03	2.853E-06	6.845E-05	1.263E-05	1.243E-06
8	.1563	5.093E+02	2.013E+02	2.301E-05	2.589E-01	9.093E-02	2.632E-03	6.809E-06	7.154E-05	1.048E-05	1.250E-06
9	.1875	5.341E+02	2.107E+02	2.197E-05	2.563E-01	9.078E-02	3.559E-03	1.431E-05	7.321E-05	8.827E-06	1.232E-06
10	.2500	5.837E+02	2.297E+02	2.016E-05	2.507E-01	9.035E-02	5.848E-03	4.459E-05	7.420E-05	6.502E-06	1.143E-06
11	.3125	6.280E+02	2.463E+02	1.879E-05	2.442E-01	8.967E-02	8.660E-03	1.059E-04	7.818E-05	5.109E-06	1.016E-06
12	.3750	6.722E+02	2.630E+02	1.761E-05	2.368E-01	8.865E-02	1.196E-02	2.098E-04	8.791E-05	4.124E-06	8.709E-07
13	.4375	7.129E+02	2.781E+02	1.665E-05	2.286E-01	8.718E-02	1.571E-02	3.643E-04	1.084E-04	3.456E-06	7.237E-07
14	.5000	7.535E+02	2.932E+02	1.579E-05	2.196E-01	8.512E-02	1.986E-02	5.762E-04	1.432E-04	2.944E-06	5.870E-07
15	.5625	7.905E+02	3.069E+02	1.509E-05	2.097E-01	8.236E-02	2.435E-02	8.461E-04	1.957E-04	2.576E-06	4.706E-07
16	.6250	8.275E+02	3.206E+02	1.444E-05	1.991E-01	7.883E-02	2.913E-02	1.174E-03	2.690E-04	2.270E-06	3.795E-07
17	.6875	8.644E+02	3.344E+02	1.385E-05	1.880E-01	7.449E-02	3.410E-02	1.552E-03	3.652E-04	2.005E-06	3.150E-07
18	.7500	9.014E+02	3.482E+02	1.330E-05	1.765E-01	6.941E-02	3.914E-02	1.962E-03	4.836E-04	1.766E-06	2.748E-07
19	.8125	9.328E+02	3.600E+02	1.286E-05	1.649E-01	6.375E-02	4.414E-02	2.376E-03	6.127E-04	1.574E-06	2.543E-07
20	.8750	9.642E+02	3.718E+02	1.245E-05	1.535E-01	5.765E-02	4.900E-02	2.775E-03	7.538E-04	1.385E-06	2.475E-07
21	1.0000	1.027E+03	3.957E+02	1.170E-05	1.319E-01	4.483E-02	5.795E-02	3.459E-03	1.059E-03	1.024E-06	2.577E-07
22	1.1250	1.074E+03	4.133E+02	1.120E-05	1.136E-01	3.318E-02	6.523E-02	3.796E-03	1.301E-03	7.432E-07	2.704E-07
23	1.2500	1.120E+03	4.309E+02	1.074E-05	9.927E-02	2.359E-02	7.069E-02	3.840E-03	1.524E-03	5.086E-07	2.777E-07
24	1.3750	1.152E+03	4.429E+02	1.045E-05	8.876E-02	1.649E-02	7.444E-02	3.640E-03	1.668E-03	3.482E-07	2.789E-07
25	1.5000	1.184E+03	4.548E+02	1.018E-05	8.131E-02	1.142E-02	7.685E-02	3.365E-03	1.815E-03	2.331E-07	2.796E-07
26	2.0000	1.260E+03	4.827E+02	9.592E-06	7.306E-02	4.567E-03	7.713E-02	2.650E-03	2.200E-03	8.372E-08	2.924E-07
27	2.5000	1.304E+03	4.986E+02	9.286E-06	7.321E-02	3.170E-03	7.419E-02	2.445E-03	2.451E-03	5.407E-08	3.066E-07
28	3.0000	1.348E+03	5.147E+02	8.996E-06	7.475E-02	2.893E-03	7.113E-02	2.459E-03	2.759E-03	4.557E-08	3.241E-07
29	4.0000	1.390E+03	5.295E+02	8.744E-06	7.717E-02	2.632E-03	6.633E-02	2.340E-03	2.975E-03	3.863E-08	3.215E-07
30	6.0000	1.475E+03	5.598E+02	8.270E-06	8.054E-02	2.430E-03	5.977E-02	2.296E-03	3.529E-03	3.098E-08	2.960E-07
31	8.0000	1.500E+03	5.679E+02	8.153E-06	8.282E-02	2.087E-03	5.539E-02	2.023E-03	3.486E-03	2.579E-08	2.585E-07
32	10.0000	1.524E+03	5.772E+02	8.022E-06	8.282E-02	2.087E-03	5.539E-02	2.023E-03	3.486E-03	2.579E-08	2.585E-07

	X	H2O	AR
1	.0000	9.558E-04	6.391E-01
2	.0156	1.124E-03	6.396E-01
3	.0313	1.311E-03	6.401E-01
4	.0469	1.518E-03	6.406E-01
5	.0625	1.746E-03	6.411E-01
6	.0938	2.266E-03	6.420E-01
7	.1250	2.875E-03	6.430E-01
8	.1563	3.580E-03	6.439E-01
9	.1875	4.390E-03	6.448E-01
10	.2500	6.348E-03	6.466E-01
11	.3125	8.889E-03	6.484E-01
12	.3750	1.218E-02	6.501E-01
13	.4375	1.640E-02	6.516E-01
14	.5000	2.174E-02	6.530E-01
15	.5625	2.834E-02	6.542E-01
16	.6250	3.627E-02	6.552E-01
17	.6875	4.551E-02	6.560E-01
18	.7500	5.587E-02	6.566E-01
19	.8125	6.707E-02	6.571E-01
20	.8750	7.880E-02	6.575E-01
21	1.0000	1.026E-01	6.582E-01
22	1.1250	1.240E-01	6.589E-01
23	1.2500	1.414E-01	6.597E-01
24	1.3750	1.544E-01	6.606E-01
25	1.5000	1.639E-01	6.614E-01
26	2.0000	1.769E-01	6.635E-01
27	2.5000	1.797E-01	6.648E-01

```
 28   3.0000  1.802E-01  6.658E-01
 29   4.0000  1.811E-01  6.675E-01
 30   6.0000  1.817E-01  6.698E-01
 31   8.0000  1.830E-01  6.712E-01
 32  10.0000  1.830E-01  6.712E-01
```

Στη συνέχεια εκτελείται το ίδιο πρόβλημα στο Matlab. Για να εκτελεστεί, δημιουργείται αρχικά αρχείο με τις βασικές παραμέτρους του συστήματος, το οποίο παρατίθεται παρακάτω, ως Πίνακας 9:

Πίνακας 9 – Αρχείο Προβλήματος Καύσης Προαναμεμιγμένης Φλόγας H2-O2

```
NSTART=6
NEWTON_ATOL=1E-10
NEWTON_RTOL=1E-4
TSTEP_ATOL=1E-5
TSTEP_RTOL=1E-5
MDOT0=4.63E-3
DIST=10.0
ENRG=false
CENDIF=true
KEEPFLUX=true
GRADVAN=true
AREA1=true
GSIGMA=0.15
P=33335.925
INIT_XSTR=0.0
INIT_XEND=10.0
INIT_XCEN=5.0
INIT_WMIX=10.0
REFINE=true
REFINE_GRAD=0.2
REFINE_CURV=0.5
XBOUND_0=[0.28 0.09 0 0 0 0 0 0 0.63]
XBOUND_INF=[0.15 0 0.01 0.02 0.02 0 0 0.12 0.68]
INIT_INTM=[0.01 0 0 0 0 0.001 0.00001 0 0]
T_PROFILE=[0.0  373.7;  0.1250  484.5;  0.250  583.7;  0.375  672.2;  0.5  753.5;  0.75
901.4; 1.0 1027; 1.25 1120; 1.5 1184; 2.0 1260; 3.0 1348; 6.0 1475; 10.0 1524]
```

Οι ρουτίνες του Matlab μαζί με το παραπάνω αρχείο διαβάζουν επίσης τα αρχεία θερμοδυναμικών και κινητικών ιδιοτήτων, καθώς και το μηχανισμό αντίδρασης που παρουσιάστηκε προηγουμένως και επιλύει το σύστημα αυτόματα. Η λύση που επιστρέφουν για τους 6 κόμβους είναι η εξής ως Πίνακας 10:

Πίνακας 10 - Αρχική Λύση Matlab Προβλήματος Καύσης H2-O2

T	H2	O2	H	O	OH	HO2	H2O2	H2O	AR
3,737E+02	2,412E-01	8,539E-02	2,495E-02	2,454E-04	1,926E-04	5,317E-09	1,490E-08	1,329E-02	6,347E-01
1,260E+03	8,645E-02	4,971E-03	8,654E-02	2,642E-03	2,112E-03	8,835E-08	2,494E-07	1,684E-01	6,489E-01
1,390E+03	8,882E-02	2,170E-03	7,592E-02	2,248E-03	2,859E-03	3,127E-08	2,788E-07	1,745E-01	6,534E-01
1,475E+03	9,284E-02	1,929E-03	6,822E-02	2,174E-03	3,373E-03	2,423E-08	2,596E-07	1,753E-01	6,561E-01
1,500E+03	9,576E-02	1,627E-03	6,298E-02	1,889E-03	3,310E-03	1,985E-08	2,256E-07	1,767E-01	6,578E-01
1,524E+03	9,576E-02	1,627E-03	6,298E-02	1,889E-03	3,310E-03	1,985E-08	2,256E-07	1,767E-01	6,578E-01

Συγκρίνοντας τα αποτελέσματα με αυτά του PREMIX, παρατηρούμε ότι είναι ίδια. Συνεχίζοντας η λύση τελικά καταλήγει και αυτή σε 32 κόμβους με τα παρακάτω αποτελέσματα ως Πίνακας 11:

Πίνακας 11 - Τελική Λύση Matlab Προβλήματος Καύσης H2-O2

T	H2	O2	H	O	OH	HO2	H2O2	H2O	AR
3,737E+02	2,683E-01	9,115E-02	3,913E-04	3,289E-08	3,681E-05	1,565E-05	6,797E-07	9,558E-04	6,391E-01
3,876E+02	2,676E-01	9,116E-02	4,050E-04	3,711E-08	4,163E-05	1,886E-05	8,203E-07	1,124E-03	6,396E-01
4,014E+02	2,669E-01	9,117E-02	4,812E-04	7,197E-08	4,653E-05	1,994E-05	9,384E-07	1,311E-03	6,401E-01
4,153E+02	2,660E-01	9,117E-02	6,058E-04	1,486E-07	5,138E-05	1,969E-05	1,034E-06	1,518E-03	6,406E-01
4,291E+02	2,652E-01	9,116E-02	7,742E-04	3,006E-07	5,594E-05	1,865E-05	1,108E-06	1,746E-03	6,411E-01
4,568E+02	2,633E-01	9,112E-02	1,235E-03	1,001E-06	6,352E-05	1,553E-05	1,201E-06	2,266E-03	6,420E-01
4,845E+02	2,612E-01	9,104E-02	1,857E-03	2,853E-06	6,845E-05	1,263E-05	1,243E-06	2,875E-03	6,430E-01
5,093E+02	2,589E-01	9,093E-02	2,632E-03	6,809E-06	7,154E-05	1,048E-05	1,250E-06	3,580E-03	6,439E-01
5,341E+02	2,563E-01	9,078E-02	3,559E-03	1,431E-05	7,321E-05	8,827E-06	1,232E-06	4,390E-03	6,448E-01
5,837E+02	2,507E-01	9,035E-02	5,848E-03	4,459E-05	7,420E-05	6,502E-06	1,143E-06	6,348E-03	6,466E-01
6,280E+02	2,442E-01	8,967E-02	8,660E-03	1,059E-04	7,818E-05	5,109E-06	1,016E-06	8,889E-03	6,484E-01
6,722E+02	2,368E-01	8,865E-02	1,196E-02	2,098E-04	8,791E-05	4,124E-06	8,709E-07	1,218E-02	6,501E-01
7,129E+02	2,286E-01	8,718E-02	1,571E-02	3,643E-04	1,084E-04	3,456E-06	7,237E-07	1,640E-02	6,516E-01
7,535E+02	2,196E-01	8,512E-02	1,986E-02	5,762E-04	1,432E-04	2,944E-06	5,870E-07	2,174E-02	6,530E-01
7,905E+02	2,097E-01	8,236E-02	2,435E-02	8,461E-04	1,957E-04	2,576E-06	4,706E-07	2,834E-02	6,542E-01
8,275E+02	1,991E-01	7,883E-02	2,913E-02	1,174E-03	2,690E-04	2,270E-06	3,795E-07	3,627E-02	6,552E-01
8,644E+02	1,880E-01	7,449E-02	3,410E-02	1,552E-03	3,652E-04	2,005E-06	3,150E-07	4,551E-02	6,560E-01
9,014E+02	1,765E-01	6,941E-02	3,914E-02	1,962E-03	4,836E-04	1,766E-06	2,748E-07	5,587E-02	6,566E-01
9,328E+02	1,649E-01	6,375E-02	4,414E-02	2,376E-03	6,127E-04	1,574E-06	2,543E-07	6,707E-02	6,571E-01
9,642E+02	1,535E-01	5,765E-02	4,900E-02	2,775E-03	7,538E-04	1,385E-06	2,475E-07	7,880E-02	6,575E-01
1,027E+03	1,319E-01	4,483E-02	5,795E-02	3,459E-03	1,059E-03	1,024E-06	2,577E-07	1,026E-01	6,582E-01
1,074E+03	1,136E-01	3,318E-02	6,523E-02	3,796E-03	1,301E-03	7,432E-07	2,704E-07	1,240E-01	6,589E-01
1,120E+03	9,927E-02	2,359E-02	7,069E-02	3,840E-03	1,524E-03	5,086E-07	2,777E-07	1,414E-01	6,597E-01
1,152E+03	8,876E-02	1,649E-02	7,444E-02	3,640E-03	1,668E-03	3,482E-07	2,789E-07	1,544E-01	6,606E-01
1,184E+03	8,131E-02	1,142E-02	7,685E-02	3,365E-03	1,815E-03	2,331E-07	2,796E-07	1,639E-01	6,614E-01
1,260E+03	7,306E-02	4,567E-03	7,712E-02	2,650E-03	2,200E-03	8,372E-08	2,924E-07	1,769E-01	6,635E-01
1,304E+03	7,321E-02	3,170E-03	7,419E-02	2,445E-03	2,451E-03	5,407E-08	3,066E-07	1,797E-01	6,648E-01
1,348E+03	7,475E-02	2,893E-03	7,113E-02	2,459E-03	2,759E-03	4,557E-08	3,241E-07	1,802E-01	6,658E-01
1,390E+03	7,717E-02	2,632E-03	6,633E-02	2,340E-03	2,975E-03	3,863E-08	3,215E-07	1,811E-01	6,675E-01
1,475E+03	8,054E-02	2,430E-03	5,977E-02	2,296E-03	3,529E-03	3,098E-08	2,960E-07	1,817E-01	6,698E-01
1,500E+03	8,282E-02	2,087E-03	5,539E-02	2,023E-03	3,486E-03	2,579E-08	2,585E-07	1,830E-01	6,712E-01
1,524E+03	8,282E-02	2,087E-03	5,539E-02	2,023E-03	3,486E-03	2,579E-08	2,585E-07	1,830E-01	6,712E-01

Συγκρινόμενα με τα τελικά αποτελέσματα του CHEMKIN, παρατηρούμε ότι είναι ίδια. Είναι επίσης αξιοσημείωτο το γεγονός ότι ο τρόπος με τον οποίο το Matlab κατέληξε στη λύση είναι ίδιος με αυτό του CHEMKIN.

6 Επίλυση Προβλημάτων στο Matlab

Κατόπιν επιβεβαίωσης της αξιοπιστίας των αλγορίθμων, το επόμενο βήμα είναι η επίλυση προβλημάτων με τη χρήση αυτών. Παρακάτω παρατίθενται προβλήματα φλογών διάχυσης, η μεθοδολογία επίλυσής τους στην κατάσταση ισορροπίας αλλά και στη μεταβατική κατάσταση τους καθώς και διαγράμματα με τα αποτελέσματα και σχολιασμός αυτών.

6.1 Πρόβλημα 1 – Προαναμεμιγμένες Φλόγες Μεθανίου-Αέρα

Το συγκεκριμένο πρόβλημα αφορά τη μελέτη αδιαβατικής φλόγας μεθανίου-αέρα σε ατμοσφαιρική πίεση, που εξελίσσεται ελεύθερα χρησιμοποιώντας αναλυτική χημεία.

Το μίγμα 9,5%$CH_4$19%$O_2$71,5%N_2 εισέρχεται στο θάλαμο με ρυθμό 40 mg/cm^{2}sec και στο σημείο που η θερμοκρασία είναι 400°C θεωρούμε ότι δεν μεταβάλλεται (θεωρείται ως θερμοκρασία αναφοράς). Ο μηχανισμός αντιδράσεων περιλαμβάνει 17 συστατικά και 59 αντιδράσεις.

Αρχικά θα βρούμε τη λύση ισορροπίας του συστήματος και κατόπιν θα μεταβάλλουμε το ρυθμό ροής μάζας και θα εξετάσουμε τη δυναμική συμπεριφορά του συστήματος από το σημείο εκείνο και μετά.

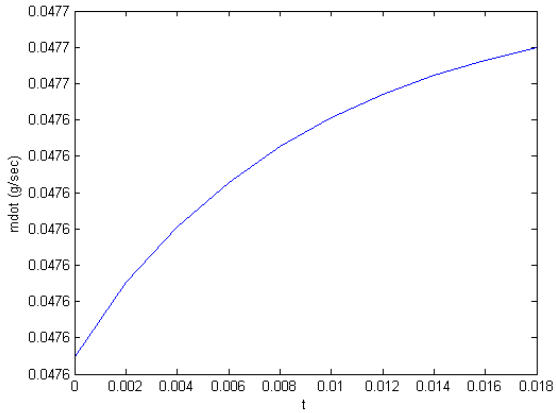
Η εύρεση της λύσης ισορροπίας παρουσιάζει δυσκολίες και για το λόγο αυτό θα γίνει τμηματικά:

- Αρχικά επιλέγεται ένα μικρό χωρικό πεδίο μήκους 0,3cm και το πρόβλημα λύνεται θεωρώντας ένα συγκεκριμένο προφίλ θερμοκρασιών, αντί να συμπεριλάβουμε την εξίσωση ενέργειας τους υπολογισμούς.
- Αφού βρεθεί λύση, συμπεριλαμβάνουμε την εξίσωση της ενέργειας, ξαναλύνουμε το σύστημα και κατόπιν εύρεσης νέας λύσης επιτρέπουμε στον αλγόριθμο να κάνει δυναμική πύκνωση του πλέγματος.
- Στη συνέχεια αυξάνουμε το χωρικό πεδίο στα 8,5cm και ξαναλύνουμε το σύστημα και αφού βρεθεί εκ νέου λύση πυκνώνουμε ακόμη περισσότερο το πλέγμα.
- Στη συνέχεια αυξάνουμε και πάλι το χωρικό πεδίο στα 12cm και ξαναλύνουμε το σύστημα. Αφού βρεθεί εκ νέου λύση ξαναπυκνώνουμε το πλέγμα.

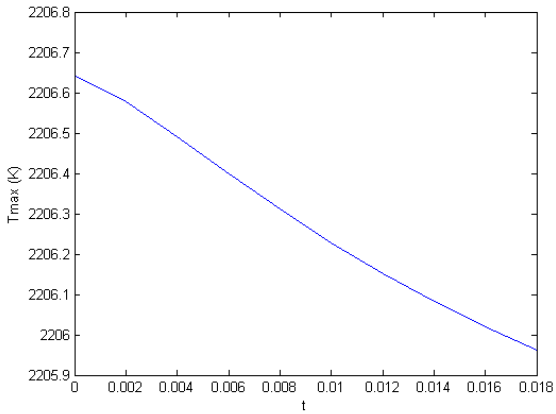
Το αρχείο με το μηχανισμό αντίδρασης, καθώς και τα αρχεία εισόδου που χρησιμοποιεί ο αλγόριθμος του Matlab για να λύσει το σύστημα διαδοχικά περιλαμβάνονται στο Παράρτημα Ε Πίνακας 18 και Πίνακας 19 αντίστοιχα, όπως επίσης και η λύση στην κατάσταση ισορροπίας, Πίνακας 20. Η λύση ισορροπίας περιλαμβάνει μετά την πύκνωση 27 κόμβους.

Ξεκινώντας από την κατάσταση ισορροπίας, επιβάλλουμε μία αύξηση 100°C στη θερμοκρασία αναφοράς και παρατηρούμε πως εξελίσσεται το σύστημα στη μεταβατική του κατάσταση.

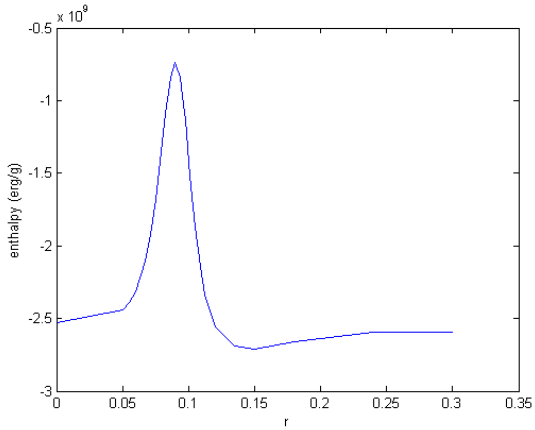
Κατόπιν χρονικής εξέλιξης για 20msec, λαμβάνουμε τα παρακάτω διαγράμματα μεταβολής του ρυθμού ροής καυσίμου, της μέγιστης θερμοκρασίας και της ενθαλπίας:



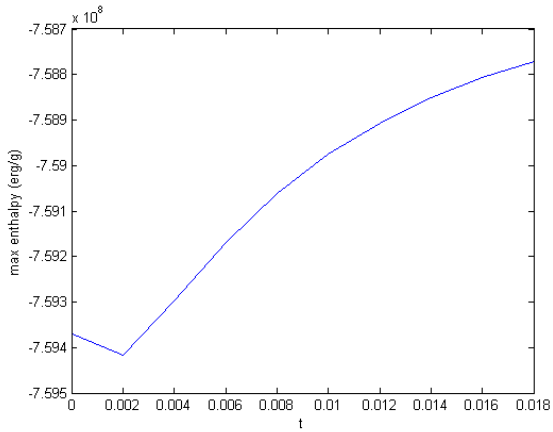
Εικόνα 1 – Πρόβλημα 1: Μεταβολή Ρυθμού Ροής Μάζας στο Χρόνο



Εικόνα 2 – Πρόβλημα 1: Μεταβολή Μέγιστης Θερμοκρασίας στο Χρόνο



Εικόνα 3 – Πρόβλημα 1: Μεταβολή Ενθαλπίας Μίγματος στο Χώρο



Εικόνα 4 – Πρόβλημα 1: Μεταβολή Μέγιστης Ενθαλπίας Μίγματος στο Χρόνο

Στην Εικόνα 1, όπου παρατίθεται η μεταβολή του ρυθμού ροής μάζας σε συνάρτηση με το χρόνο, παρατηρούμε ότι με την αύξηση της θερμοκρασίας αναφοράς αυξάνει ομαλά και ο ρυθμός ροής μάζας, χωρίς ανώμαλα σημεία. Ανάλογα αποτελέσματα παρατηρούμε και στην Εικόνα 2, όπου παρουσίαζεται η μέγιστη θερμοκρασία σε συνάρτηση με το χρόνο και η οποία μειώνεται ομαλά (πιθανώς εξαιτίας της αύξησης της ροής).

Στην Εικόνα 3 παρατίθεται η μεταβολή της ενθαλπίας σε συνάρτηση με τη χωρική συντεταγμένη. Παρατηρούμε πως η κορυφή, δηλαδή το κέντρο της καύσης, βρίσκεται στο σημείο r=0.9cm. Η επόμενη Εικόνα 4, στην οποία παρατίθεται η μεταβολή της μέγιστης ενθαλπίας, που βρίσκεται συνέχεια στο σημείο r=0.9cm, ως προς το χρόνο, δείχνει ότι η μέγιστη ενθαλπία παρουσιάζει αύξηση από τα 2msec και μετά. Η μείωση που φαίνεται αρχικά ενδεχομένως να οφείλεται σε σφάλμα ακρίβειας.

6.2 Πρόβλημα 2 – Φλόγες Διάχυσης Υδρογόνου

Το συγκεκριμένο πρόβλημα αφορά τη μεταβατική συμπεριφορά οπτικά λεπτών σφαιρικών φλογών διάχυσης (Christiansen, et al., 2003). Έχει δειχθεί αναλυτικά ότι μπορεί να εμφανιστούν ταλαντώσεις κοντά στα όρια φλογών διάχυσης με μη μοναδιαίους αριθμούς Lewis Le (Kirkby & Schmitz, 1966). Ένα μίγμα καυσίμου σε αέρια φάση, που περιέχει ήλιο ως διαλύτη ώστε να επιτευχθούν μη μοναδιαίοι αριθμοί Lewis Le, εισέρχεται από σημειακή πηγή στο οξειδωτικό περιβάλλον. Η αέρια φάση εξασφαλίζει ότι δε θα υπάρχει επιστροφή μεταξύ της φλόγας και της πηγής του καυσίμου, η οποία θα μπορούσε να εμποδίσει το μηχανισμό ταλάντωσης. Χρησιμοποιείται μίγμα καυσίμου 100%H_2 ($Le = 0.3$) σε 21%O_2/79%He ($Le = 1.9$).

Μελετώνται δύο περιπτώσεις, μία με απώλεια θερμότητας από ακτινοβολία, όπου το σύστημα θεωρείται ισοβαρές και μία αδιαβατική. Προηγούμενες μελέτες έχουν καταδείξει ότι η χρήση απλής χημείας και κινητικών μοντέλων μπορεί να οδηγήσουν σε ανακριβή αποτελέσματα κατά τη μελέτη της μεταβατικής κατάστασης (Christiansen, et al., 2001). Συνεπώς, θα χρησιμοποιηθεί αναλυτική χημεία και κινητικά μοντέλα για την παραγωγή ενός συστήματος, όσο το δυνατόν πιο ρεαλιστικού, χρησιμοποιώντας εξισώσεις mixture-average. Για τις φλόγες μεθανίου υιοθετείται ο μηχανισμός GRI-Mech 1.2 (Frenklach, et al., 1995), που αποτελείται από 31 συστατικά και 175 αντιδράσεις. Για τις φλόγες υδρογόνου χρησιμοποιείται ένας άλλος πιο κατάλληλος μηχανισμός (Mueller, et al., 1999), που αποτελείται από 9 συστατικά και 21 αντιδράσεις.

Ο παρακάτω Πίνακας 12 συνοψίζει την περιγραφή του προβλήματος, περιλαμβάνοντας και τις επιβαλλόμενες συνοριακές συνθήκες:

Πίνακας 12 – Περιγραφή Προβλήματος Φλογών Διάχυσης

Μελετούμενη διεργασία	Φλόγες διάχυσης
Μελετούμενη κατάσταση	Ισορροπίας και μεταβατική
Τύπος μελέτης	Αριθμητική εξομοίωση
Γεωμετρία	Σφαιρική (μονοδιάστατη)
Υπόθεση πάχους φλόγας	Οπτικά λεπτή
Χημεία και κινητικό μοντέλο	Λεπτομερές, με εξισώσεις mixture average
Θερμοδυναμική κατάσταση	1. Ισοβαρής με απώλεια θερμότητας από ακτινοβολία 2. Αδιαβατική

Μίγμα καυσίμου	100%H_2
Οξειδωτικό	21%O_2/79%He
Συστατικά με ακτινοβολία	CO, CO_2 και H_2O
Μέγεθος περιοχής υπολογισμού	500cm
Πίεση	1atm
Συνοριακές συνθήκες	Στην πλευρά εισόδου του καυσίμου: • $\dot{m} = \dot{m}_0$ (σταθερός ρυθμός ροής μάζας) • $Y_k = \epsilon_k$ (συγκεκριμένα κλάσματα μάζας συστατικών) • $T = T_0 = 300$ K Στην ελεύθερη πλευρά: • $Y_k = Y_\infty$ (κλάσματα μάζας συστατικών ίδια με αυτά του οξειδωτικού) • $T = T_\infty = 300$ K

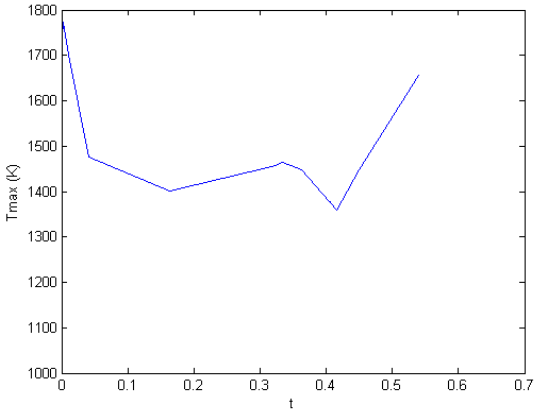
Λόγω του μεγάλου μεγέθους του χωρικού πεδίου, 500cm, δημιουργούμε ένα ανομοιογενές χωρικό πλέγμα, το οποίο είναι πυκνό μέχρι τα 3cm και αραιό στην υπόλοιπη περιοχή. Ο Πίνακας 13 παρουσίαζει τα χαρακτηριστικά της αρχικής λύσης που επιλέξαμε για εκκίνηση εξετάζοντας το πρόβλημα αδιαβατικής κατάστασης:

Πίνακας 13 – Αρχείο Παραμέτρων Προβλήματος Φλογών Διάχυσης Υδρογόνου (α)

```
NSTART=25
NEWTON_ATOL=1E-10
NEWTON_RTOL=1E-4
TSTEP_ATOL=1E-5
TSTEP_RTOL=1E-5
MDOT0=0.200E-3
ADIAB=false
ENRG=true
CENDIF=true
KEEPFLUX=false
GRADVAN=false
AREA1=false
GSIGMA=0.10
INIT_XSTR=0.0
INIT_XEND=500.0
INIT_XCEN=1.0
INIT_WMIX=0.1
REFINE_GRAD=0.9
REFINE_CURV=0.9
XBOUND_0=[1 0 0 0 0 0 0 0 0]
XBOUND_INF=[0 0.21 0 0 0 0 0 0 0.79]
INIT_INTM=[0.01 0.05 0.0001 0.001 0.001 0.0001 0.00001 0.05 0.93]
GRID=[0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8 3 4 5 10 20 50 100 200 350 500]
```

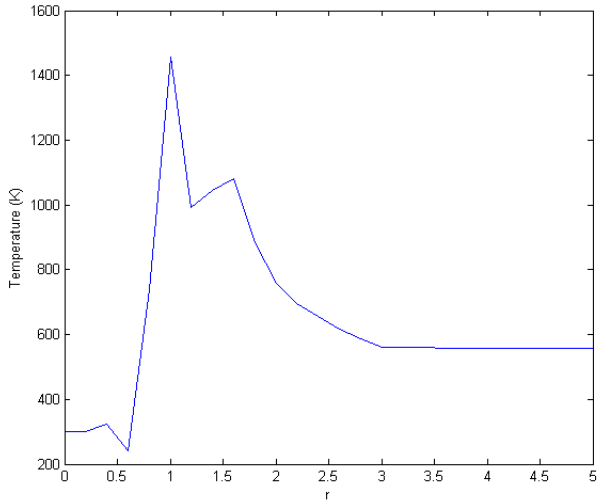
```
T_PROFILE=[0 300; 0.2 760; 0.4 925; 0.6 990; 0.8 1025; 1 1040; 1.2 1000; 1.4 900; 1.6 825; 1.8 760; 2 710; 2.2 660; 2.4 625; 2.6 590; 2.8 560; 3 535; 4 535; 5 534; 10 532; 20 527; 50 513; 100 489; 200 442; 350 371; 500 300]
TRANSIENT=true
TRAN_STEPS=20
TSTEP_DT=1E-6
TSTEP_DTMAX=1E-2
TSTEP_MAXSTEPS=5
TSTEP_DTMUL=2
TSTEP_DTDIV=2.2
TSTEP_MAXSAMESTEPS=20
MDOTVAR=false
PREMIX=false
```

Το πρόβλημα έτρεξε σε transient ανάλυση για συνολικό χρόνο 0,54sec χρησιμοποιώντας τους αλγόριθμους του Matlab. Η μεταβολή της μέγιστης θερμοκρασίας σε συνάρτηση με το χρόνο παρατίθεται στην Εικόνα 5.



Εικόνα 5 – Πρόβλημα 2(α): Μεταβολή Μέγιστης Θερμοκρασίας στο Χρόνο

Παρατηρούμε ότι οι θερμοκρασίες κυμαίνονται μεταξύ 1350 και 1800 βαθμών C, που είναι υψηλότερες από αυτές άλλων μελετών (Christiansen, et al., 2003), λόγω της αρχικής λύσης εκκίνησης που επιλέξαμε. Επίσης λόγω της μικρής κλίμακας διακριτοποίησης, για λόγους υπολογιστικής ισχύος, η καμπύλη δεν είναι ομαλή. Παρ' όλα αυτά, παρατηρείται μία αυξομείωση της μέγιστης θερμοκρασίας, η οποία μπορεί να υποδηλώνει ταλάντωση. Η ταλάντωση αυτή φαίνεται να έχει συχνότητα 0,4 sec.



Εικόνα 6 – Πρόβλημα 2(α): Μεταβολή Θερμοκρασίας στο Χώρο

Η Εικόνα 6 παρουσίαζει τη μεταβολή της θερμοκρασίας σε συνάρτηση με τη χωρική συντεταγμένη. Αν και το διάγραμμα δεν είναι ομαλό, λόγω χαμηλής διακριτοποίησης, αλλά και του γεγονότος ότι δεν ξεκίνησε από λύση ισορροπίας, εντούτοις η μορφή του ακολουθεί τα αποτελέσματα άλλων μελετών. Επισημαίνεται ότι στο παραπάνω σχήμα φαίνονται μόνο τα πρώτα πέντε (5) cm του χωρικού πεδίου, καθόσον αυτή είναι η κρίσιμη περιοχή με τις μεγάλες μεταβολές στις μεταβλητές.

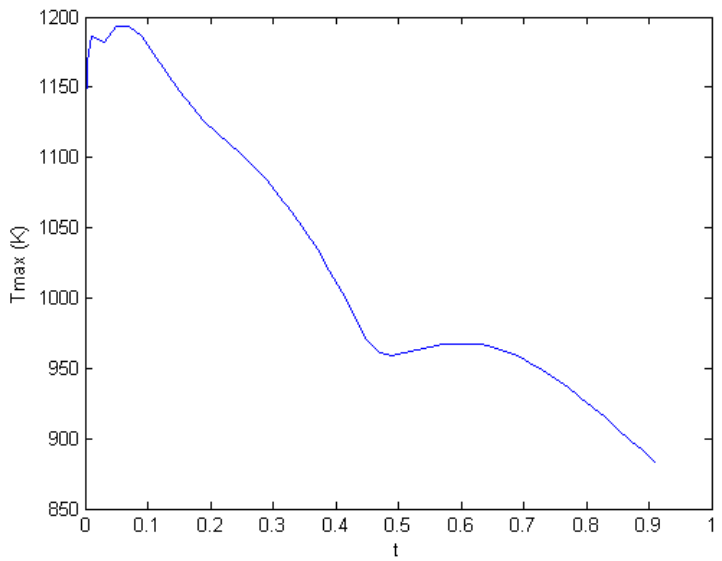
Στη συνέχεια αλλάζουμε τις παραμέτρους του συστήματος και εξετάζουμε καύση υδρογόνου που περιλαμβάνει θερμικές απώλειες από ακτινοβολία. Ο Πίνακας 14 περιλαμβάνει τις βασικές παραμέτρους του προβλήματος:

Πίνακας 14 – Αρχείο Παραμέτρων Προβλήματος Φλογών Διάχυσης Υδρογόνου (β)

```
NSTART=25
NEWTON_ATOL=1E-10
NEWTON_RTOL=1E-4
TSTEP_ATOL=1E-5
TSTEP_RTOL=1E-5
MDOT0=0.250E-3
ADIAB=false
ENRG=true
CENDIF=true
KEEPFLUX=false
GRADVAN=false
AREA1=false
GSIGMA=0.10
INIT_XSTR=0.0
INIT_XEND=500.0
```

```
INIT_XCEN=1.0
INIT_WMIX=0.1
REFINE_GRAD=0.9
REFINE_CURV=0.9
XBOUND_0=[1 0 0 0 0 0 0 0 0]
XBOUND_INF=[0 0.21 0 0 0 0 0 0 0.79]
INIT_INTM=[0.01 0.005 0.0001 0.001 0.001 0.0001 0.00001 0.01 0.99]
GRID=[0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8 3 4 5 10 20 50 100 200 350 500]
T_PROFILE=[0 300; 0.2 760; 0.4 925; 0.6 990; 0.8 1025; 1 1040; 1.2 1000; 1.4 900; 1.6 825; 1.8 760; 2 710; 2.2 660; 2.4 625; 2.6 590; 2.8 560; 3 535; 4 535; 5 534; 10 532; 20 527; 50 513; 100 489; 200 442; 350 371; 500 300]
TRANSIENT=true
TRAN_STEPS=20
TSTEP_DT=1E-6
TSTEP_DTMAX=1E-2
TSTEP_MAXSTEPS=5
TSTEP_DTMUL=2
TSTEP_DTDIV=2.2
TSTEP_MAXSAMESTEPS=20
MDOTVAR=false
PREMIX=false
```

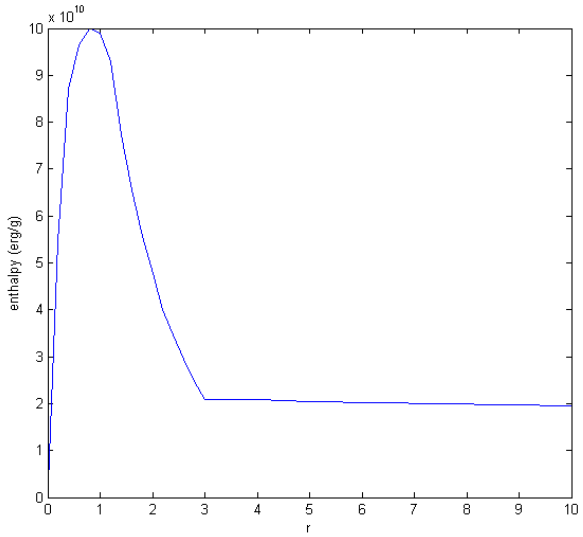
Το πρόβλημα έτρεξε σε transient ανάλυση για συνολικό χρόνο 0,91sec χρησιμοποιώντας τους αλγόριθμους του Matlab. Η μεταβολή της μέγιστης θερμοκρασίας σε συνάρτηση με το χρόνο παρουσιάζεται στην Εικόνα 7.



Εικόνα 7 – Πρόβλημα 2(β): Μεταβολή Μέγιστης Θερμοκρασίας στο Χρόνο

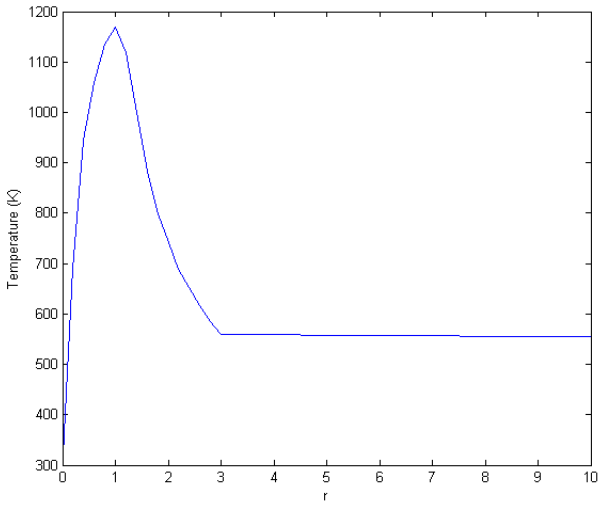
Η παραπάνω είκονα παρουσιάζει ένα ομαλό διάγραμμα μεταβολής της μέγιστης θερμοκρασίας, κάτι το οποίο σημαίνει ότι η αρχική λύση που επιλέχθηκε ήταν καλή. Η

μέγιστη θερμοκρασία παρουσιάζει μία αυξομείωση στο σημείο των 0.5sec, η οποία υποδεικνύει ταλάντωση. Επίσης φαίνεται πως έπειτα από κάποιο διάστημα η φλόγα θα αποσβεστεί.



Εικόνα 8 – Πρόβλημα 2(β): Μεταβολή Ενθαλπίας στο Χώρο

Η Εικόνα 8 παρουσιάζει τη μεταβολή της ενθαλπίας του μίγματος ως προς τη χωρική συντεταγμένη. Το μέγιστο σημείο είναι στο 1cm, όπως είχαμε υποθέσει και στην αρχική εκτίμηση της λύσης εκκίνησης. Η καμπύλη είναι ομαλή, αλλά παρουσιάζει μία απότομη καμπή στο σημείο όπου $r = 3$cm. Αυτό εξηγείται από το γεγονός ότι από το σημείο εκείνο και μετά το χωρικό πλέγμα αραιώνει απότομα.



Εικόνα 9 – Πρόβλημα 2(β): Μεταβολή Θερμοκρασίας στο Χώρο

Η Εικόνα 9 παρουσιάζει τη μεταβολή της θερμοκρασίας του μίγματος ως προς τη χωρική συντεταγμένη, στο χρονικό σημείο $t = 0.026\text{sec}$. Παρατηρούμε όπως και προηγουμένως μία ομαλή καμπύλη, η οποία παρουσιάζει καμπή στο σημείο που αραιώνει το πλέγμα.

7 Συμπεράσματα

Σκοπός της παρούσας εργασίας ήταν να μοντελοποιήσει και να λύσει τις μερικές διαφορικές εξισώσεις που διέπουν το πρόβλημα καύσης φλογών διάχυσης και προαναμεμιγμένων φλογών, συμπεριλαμβάνοντας τον πλήρη μηχανισμό της καύσης, στο περιβάλλον του Matlab, με σκοπό να μπορεί να χρησιμοποιηθεί μελλοντικά για την ανάλυση διακλάδωσης τέτοιων συστημάτων από το Matcont.

Το πλήρες σύστημα εξισώσεων και υπολογισμού κινητικών και θερμοδυναμικών ιδιοτήτων καταστρώθηκε με επιτυχία και οι αλγόριθμοι που αναπτύχθηκαν επιβεβαιώθηκαν ως προς την ακρίβειά τους συγκρίνοντας τα αποτελέσματα που παράγουν με αυτά του CHEMKIN.

Στα πλαίσια εφαρμογής των αλγορίθμων αυτών, έγινε μοντελοποίηση και επίλυση δύο χαρακτηριστικών προβλημάτων καύσης, ένα με προαναμεμιγμένη και ένα με φλόγα διάχυσης. Τα προβλήματα αυτά εξετάστηκαν στην κατάσταση ισορροπίας αλλά και στη δυναμική (μεταβατική) τους κατάσταση. Τα αποτελέσματα που προέκυψαν συγκρίθηκαν με αυτά άλλων μελετών.

7.1 Μελλοντική Έρευνα

Η μοντελοποίηση στο Matlab των εξισώσεων και του πλήρους μηχανισμού αντιδράσεων καύσεων φλογών διάχυσης και προαναμεμιγμένων φλογών θα συνεχιστεί και θα βελτιωθεί ακόμα περισσότερο κατόπιν ολοκλήρωσης της παρούσας εργασίας.

Το πιο βασικό είναι ο συνδυασμός των αλγορίθμων που αναπτύχθηκαν με το Matcont, έτσι ώστε να είναι δυνατή η συστηματική μελέτη διακλάδωσης τέτοιων προβλημάτων.

Επιπρόσθετα, θα γίνουν βελτιώσεις στον ίδιο τον κώδικα, προκειμένου αφενός να επιταχυνθεί η λειτουργία του αλλά και για να προστεθούν νέες δυνατότητες, όπως η

ενσωμάτωση της μεθόδου επίλυσης Arc-Length (πέραν από τη Newton), η δυνατότητα υπολογισμού των κινητικών και θερμοδυναμικών ιδιοτήτων με πολυσυστατική (multicomponent) μεθοδολογία (πέραν από τη mixture-average) αλλά και περισσότερες δυνατότητες παρουσίασης των αποτελεσμάτων.

ΠΑΡΑΡΤΗΜΑ Α

Διαθέσιμες Επιλογές του Αλγορίθμου

Πίνακας 15 – Διαθέσιμες Επιλογές Αλγορίθμου Matlab

#	Επιλογή	Αρχική Τιμή	Περιγραφή
1	nstart	10	αρχικός αριθμός κόμβων
2	newton_atol	1e-9	απόλυτο σφάλμα σύγκλισης για τη μέθοδο Newton
3	newton_rtol	1e-4	σχετικό σφάλμα σύγκλισης για τη μέθοδο Newton
4	tstep_atol	1e-9	απόλυτο σφάλμα σύγκλισης για τη μέθοδο χρονικού βηματισμού
5	tstep_rtol	1e-4	σχετικό σφάλμα σύγκλισης για τη μέθοδο χρονικού βηματισμού
6	mdot0	0.0	ρυθμός ροής μάζας εισόδου του καυσίμου
7	dist	0.0	μήκος του πεδίου καύσης
8	matcont	false	αν είναι true τότε οι ρουτίνες αφορούν επίλυση στο Matcont
9	restart	false	αν είναι true τότε η επίλυση αφορά επανεκκίνηση από προηγούμενη λύση
10	adiab	false	αν είναι true τότε έχουμε αδιαβατική καύση αλλιώς ελεύθερη
11	enrg	true	αν είναι true τότε θα συμπεριληφθεί η εξίσωση ενέργειας στην επίλυση, αλλιώς οι θερμοκρασίες θα θεωρηθούν αμετάβλητες και θα ληφθούν από συγκεκριμένο προφίλ που θα ορίσει ο χρήστης
12	ltdif	false	αν είναι true τότε θα στον υπολογισμό των ταχυτήτων διάχυσης θα συμπεριληφθούν και οι θερμικές ταχύτητες διάχυσης
13	cendif	true	αν είναι true τότε θα χρησιμοποιηθούν κεντρικές διαφορές για τη διακριτοποίηση των μεταβλητών ως προς τη χωρική συντεταγμένη, αλλιώς πεπερασμένες διαφορές προς τα πίσω (backward differences)
14	keepflux	true	αν είναι true τότε η συνοριακή συνθήκη για τα κλάσματα μάζας των συστατικών στην είσοδο του καυσίμου είναι η: $\varepsilon_{k,1} - Y_{k,1} - \left(\frac{\rho A Y_k V_k}{\dot{m}}\right)_{1\frac{1}{2}} = 0$ αλλιώς δίνεται από συνθήκη Dirichlet: $Y_k = \varepsilon_k$
15	gradvan	true	αν είναι true τότε η συνοριακή συνθήκη για τα κλάσματα μάζας των συστατικών και τη θερμοκρασία στο άκρο μακριά από την καύση είναι συνθήκη Neuman: $\left.\frac{\partial Y_k}{\partial r}\right\|_{i=N} = \left.\frac{\partial T}{\partial r}\right\|_{i=N} = 0$, αλλιώς είναι συνθήκη Dirichlet: $Y_{k,N} = Y_\infty$ και $T_N = T_\infty$
16	area1	false	αν είναι true τότε η επιφάνεια μετώπου της καύσης σε κάθε κόμβο ισούται με τη μονάδα, αλλιώς ισούται με σφαιρική επιφάνεια ακτίνας r
17	refine	false	αν είναι true τότε κατόπιν επίλυσης του προβλήματος θα γίνει αναθεώρηση του πλέγματος ώστε να πυκνώσει σε σημεία μεγάλων μεταβολών των μεταβλητών
18	gsigma	0.10	παράμετρος της εξίσωσης Gauss που χρησιμοποιείται για την αρχική εκτίμηση της κατανομής των μοριακών κλασμάτων των συστατικών
19	lobound_mdot	0.0	το κάτω επιτρεπτό όριο για το ρυθμό ροής μάζας
20	hibound_mdot	10000	το άνω επιτρεπτό όριο για το ρυθμό ροής μάζας
21	lobound_Y	-1e-3	το κάτω επιτρεπτό όριο για τα κλάσματα μάζας των συστατικών
22	hibound_Y	10	το άνω επιτρεπτό όριο για τα κλάσματα μάζας των συστατικών
23	lobound_T	50	το κάτω επιτρεπτό όριο για τη θερμοκρασία

24	hibound_T	6000	το άνω επιτρεπτό όριο για τη θερμοκρασία
25	newton_jage	20	μέγιστος αριθμός βημάτων της μεθόδου Newton πριν επανυπολογιστεί η Ιακωβιανή
26	tstep_jage	20	μέγιστος αριθμός βημάτων της μεθόδου χρονικού βηματισμού πριν επανυπολογιστεί η Ιακωβιανή
27	tstep_maxsamesteps	50	μέγιστος αριθμός ίδιων βημάτων της μεθόδου χρονικού βηματισμού για να αλλάξει το μέγεθος του βήματος
28	tstep_newtonsteps	100	μέγιστος επιτρεπτός αριθμός βημάτων της μεθόδου Newton
29	newton_lreductions	5	μέγιστος επιτρεπτός αριθμός υποδιπλασιασμών της παραμέτρου απόσβεσης λ_n
30	tstep_maxtrials	10	μέγιστος αριθμός δοκιμαστικών βημάτων της μεθόδου χρονικού βηματισμού
31	tstep_dt	1e-6	χρονικό βήμα της μεθόδου χρονικού βηματισμού
32	tstep_dtmin	1e-10	ελάχιστο χρονικό βήμα
33	tstep_dtmax	1e-4	μέγιστο χρονικό βήμα
34	tstep_dtmul	2.0	παράγοντας αύξησης του χρονικού βήματος
35	tstep_dtdiv	2.2	παράγοντας μείωσης του χρονικού βήματος
36	tstep_ftol	1e-10	κριτήριο συνέχισης της μεθόδου χρονικού βηματισμού (αν η νόρμα $\|F\|_\infty$ είναι μεγαλύτερη από αυτό το όριο)
37	tstep_maxsteps	100	μέγιστος αριθμός βημάτων της μεθόδου χρονικού βηματισμού, πριν επιστρέψουμε σε κανονική επίλυση με Newton
38	refine_magfac	1e-8	συντελεστής που καθορίζει ποιες μεταβολές στις τιμές των μεταβλητών από κόμβο σε κόμβο θεωρούνται μεγάλες
39	refine_grad	0.1	συντελεστής που καθορίζει κάτω από ποιες κλίσεις (πρώτη παράγωγος) των μεταβλητών απαιτείται προσθήκη κόμβων
40	refine_curv	0.5	συντελεστής που καθορίζει κάτω από ποιες καμπυλώσεις (δεύτερη παράγωγος) των μεταβλητών απαιτείται προσθήκη κόμβων
41	maxpoints	100	μέγιστος αριθμός κόμβων του πλέγματος
42	refine_maxadd	20	μέγιστος αριθμών κόμβων που μπορούν να προστεθούν κάθε φορά που εκτελείται η ρουτίνα αναθεώρησης του πλέγματος
43	p	1013250	η πίεση του συστήματος
44	init_xstr	0	η συντεταγμένη του πρώτου κόμβου
45	init_xend	0	η συντεταγμένη του τελευταίου κόμβου
46	init_xcen	0	η συντεταγμένη που βρίσκεται το κέντρο της καύσης
47	init_wmix	0	το πάχος της περιοχής καύσης
48	xbound_0	[]	διάνυσμα με τα μοριακά κλάσματα του καυσίμου στην είσοδο του καυσίμου (στο σημείο του πρώτου κόμβου), οι οποίες χρησιμοποιούνται για τον υπολογισμό της αρχικής λύσης αλλά και ως συνοριακές συνθήκες Dirichlet για τα κλάσματα μάζας των συστατικών κατά την επίλυση
49	xbound_inf	[]	διάνυσμα με τα μοριακά κλάσματα του οξειδωτικού μακριά από την καύση (στο σημείο του τελευταίου κόμβου), οι οποίες χρησιμοποιούνται για τον υπολογισμό της αρχικής λύσης αλλά και ως συνοριακές συνθήκες Dirichlet για τα κλάσματα μάζας των συστατικών κατά την επίλυση
50	tbound_0	0	η τιμή της θερμοκρασίας στην είσοδο του καυσίμου (στο σημείο του πρώτου κόμβου), η οποία χρησιμοποιούνται και ως συνοριακή συνθήκη Dirichlet
51	tbound_inf	0	η τιμή της θερμοκρασίας μακριά από την καύση (στο σημείο του τελευταίου κόμβου), η οποία χρησιμοποιούνται και ως συνοριακή συνθήκη Dirichlet

52	init_intm	[]	διάνυσμα με τα μοριακά κλάσματα των ενδιάμεσων συστατικών στο κέντρο της καύσης, το οποίο χρησιμοποιείται για την εκτίμηση της αρχικής λύσης
53	t_profile	[]	προφίλ θερμοκρασιών που χρησιμοποιείται όταν δε χρησιμοποιούμε την εξίσωση ενέργειας αλλά θεωρούμε πως οι θερμοκρασίες είναι αμετάβλητες ως προς το χρόνο
54	tfixt	0	θερμοκρασία αναφοράς σε αδιαβατική καύση
55	nfixt	0	κόμβος στον οποίο αντιστοιχεί η θερμοκρασία αναφοράς σε αδιαβατική καύση
56	grid	[]	διάνυσμα με τις συντεταγμένες κάθε κόμβου, το οποίο θα χρησιμοποιηθεί στη θέση του αυτόματα υπολογισμένου ομοιογενούς πλέγματος

Ο χρήστης μπορεί να ορίσει τις τιμές των ανωτέρω επιλογών από αρχείο, όπου σε κάθε γραμμή υπάρχει το όνομα της επιλογής και η τιμή της στη μορφή:

ΕΠΙΛΟΓΗ=ΤΙΜΗ

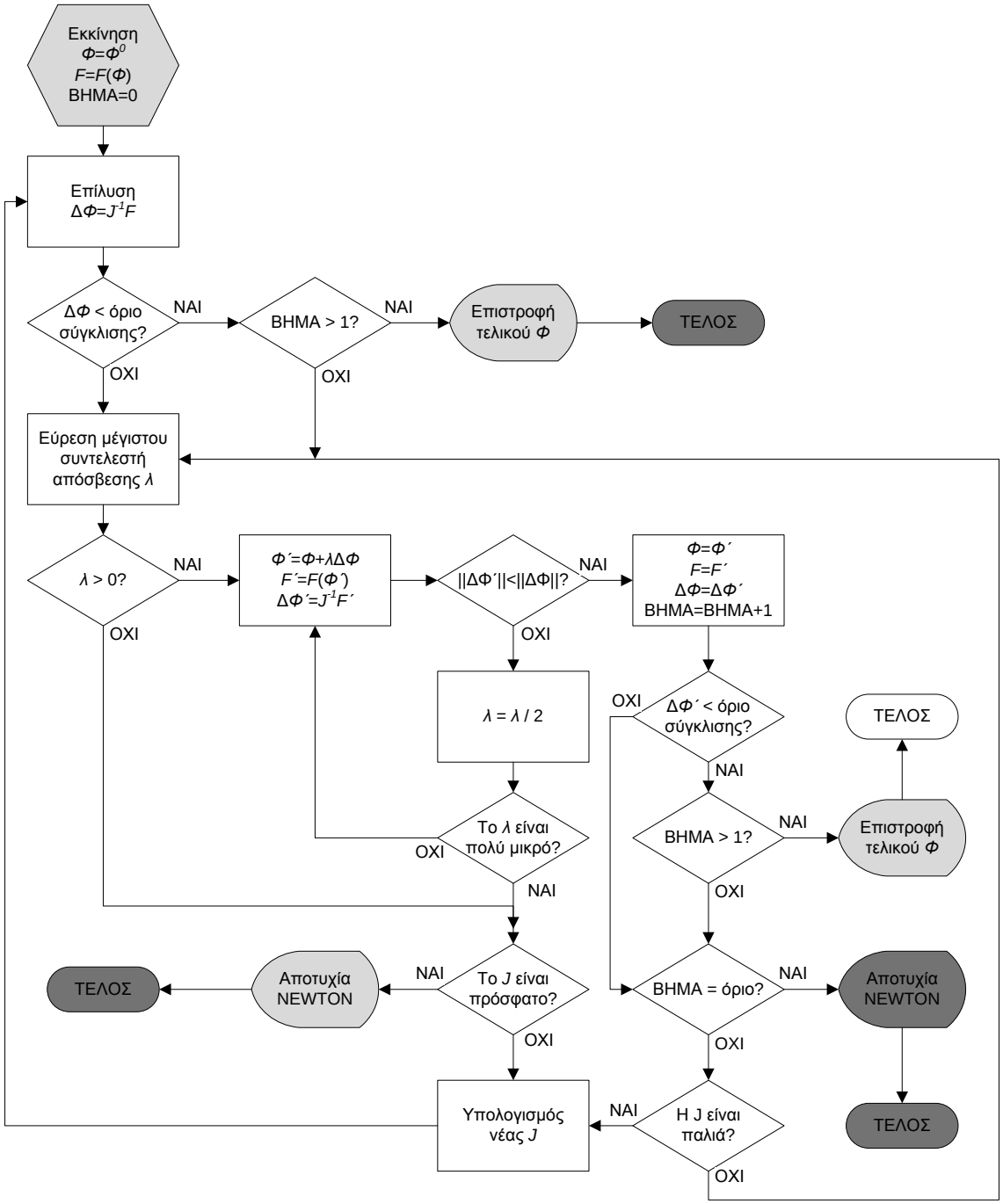
ΕΠΙΛΟΓΗ=ΤΙΜΗ

...κοκ...

ΠΑΡΑΡΤΗΜΑ Β

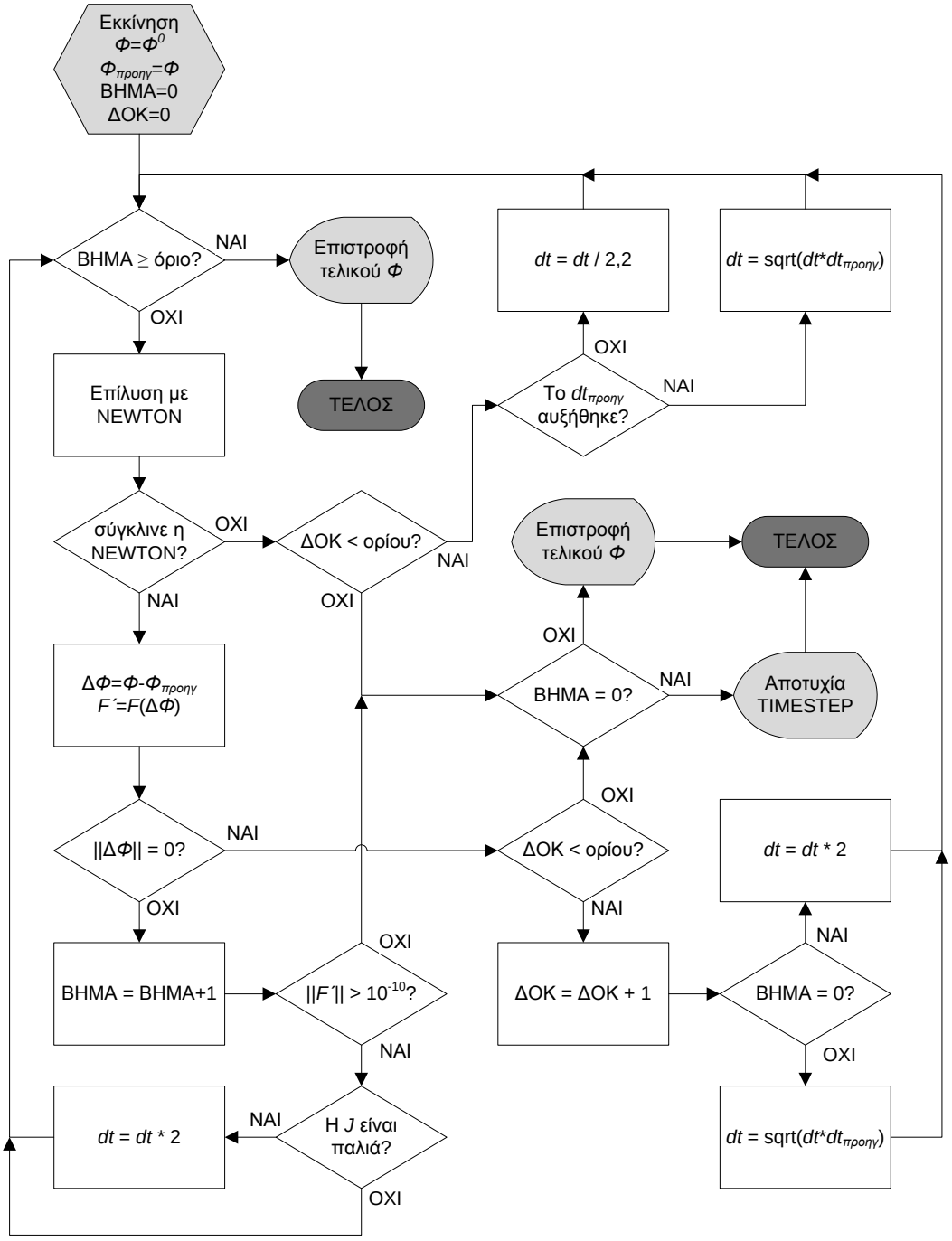
Ροϊκά Διαγράμματα Αλγορίθμων

Ροϊκό Διάγραμμα Αλγορίθμου Newton



Διάγραμμα 2 – Ροϊκό Διάγραμμα Αλγορίθμου Newton

Ροικό Διάγραμμα Αλγορίθμου Χρονικού Βηματισμού



Διάγραμμα 3 – Ροϊκό Διάγραμμα Αλγόριθμου Χρονικού Βηματισμού

ΠΑΡΑΡΤΗΜΑ Γ

Δεδομένα Αρχείου tran.out

Τα παρακάτω δεδομένα αφορούν το πρόβλημα καύσης φλογών υδρογόνου με οξυγόνο και αργό, το οποίο περιλαμβάνεται στο εγχειρίδιο του PREMIX.

Πίνακας 16 – Πολυωνυμικοί Συντελεστές Κινητικών Ιδιοτήτων Καύσης Φλογών H_2-O_2

```
TRANFT:  Transport property fitting,
          CHEMKIN-II Version 3.0, November 1990
          DOUBLE PRECISION



OUTPUT FROM THE TRANSPORT PROPERTY FITTING PACKAGE



CKLIB: CHEMKIN-III GAS-PHASE CHEMICAL KINETICS LIBRARY,
DOUBLE PRECISION Vers. 5.12 97/10/28
Copyright 1995, Sandia Corporation.
The U.S. Government retains a limited license in this software.



DATA HAS BEEN FIT OVER THE RANGE:  TLOW=   300.00, THIGH=  3500.00



TRANSPORT PARAMETERS FROM DATA BASE:

AR                 0   136.500     3.330     0.000     0.000     0.000
H                  0   145.000     2.050     0.000     0.000     0.000
H2                 1    38.000     2.920     0.000     0.790   280.000
H2O                2   572.400     2.605     1.844     0.000     4.000
H2O2               2   107.400     3.458     0.000     0.000     3.800
HO2                2   107.400     3.458     0.000     0.000     1.000  !(*)
O                  0    80.000     2.750     0.000     0.000     0.000
O2                 1   107.400     3.458     0.000     1.600     3.800
OH                 1    80.000     2.750     0.000     0.000     0.000


COEFFICIENTS FOR SPECIES CONDUCTIVITIES

 MAXIMUM FITTING ERROR =    5.371E-03

 H2            .116E+02   -.152E+01    .272E+00   -.103E-01
 O2           -.301E+01    .338E+01   -.343E+00    .151E-01
 H            -.325E+00    .342E+01   -.363E+00    .159E-01
 O             .197E+01    .180E+01   -.155E+00    .691E-02
 OH            .153E+02   -.378E+01    .613E+00   -.272E-01
 HO2           .555E+00    .159E+01   -.528E-01    .407E-03
 H2O2          .625E+00    .143E+01    .175E-02   -.355E-02
 H2O           .228E+02   -.870E+01    .149E+01   -.741E-01
 AR           -.273E+01    .327E+01   -.345E+00    .151E-01
```

```
COEFFICIENTS FOR SPECIES VISCOSITIES

 MAXIMUM FITTING ERROR =    2.087E-03

 H2           -.138E+02    .971E+00   -.453E-01    .210E-02
 O2           -.168E+02    .252E+01   -.249E+00    .110E-01
 H            -.199E+02    .342E+01   -.363E+00    .159E-01
 O            -.148E+02    .180E+01   -.155E+00    .691E-02
 OH           -.148E+02    .180E+01   -.155E+00    .691E-02
 HO2          -.168E+02    .252E+01   -.249E+00    .110E-01
 H2O2         -.168E+02    .252E+01   -.249E+00    .110E-01
 H2O          -.119E+02   -.788E+00    .334E+00   -.199E-01
 AR           -.186E+02    .327E+01   -.345E+00    .151E-01


COEFFICIENTS FOR SPECIES DIFFUSION COEFFICIENTS

 MAXIMUM FITTING ERROR =    2.046E-03

 H2        H2           -.102E+02    .215E+01   -.697E-01    .323E-02

 O2        H2           -.123E+02    .274E+01   -.146E+00    .650E-02
 O2        O2           -.158E+02    .357E+01   -.252E+00    .110E-01

 H         H2           -.117E+02    .288E+01   -.164E+00    .727E-02
 H         O2           -.172E+02    .486E+01   -.422E+00    .185E-01
 H         H            -.148E+02    .420E+01   -.328E+00    .141E-01

 O         H2           -.106E+02    .216E+01   -.652E-01    .281E-02
 O         O2           -.147E+02    .335E+01   -.225E+00    .991E-02
 O         H            -.150E+02    .413E+01   -.328E+00    .144E-01
 O         O            -.133E+02    .294E+01   -.171E+00    .755E-02

 OH        H2           -.106E+02    .216E+01   -.652E-01    .281E-02
 OH        O2           -.147E+02    .334E+01   -.223E+00    .983E-02
 OH        H            -.150E+02    .414E+01   -.328E+00    .145E-01
 OH        O            -.133E+02    .294E+01   -.171E+00    .755E-02
 OH        OH           -.133E+02    .294E+01   -.171E+00    .755E-02

 HO2       H2           -.123E+02    .274E+01   -.146E+00    .650E-02
 HO2       O2           -.158E+02    .357E+01   -.252E+00    .110E-01
 HO2       H            -.172E+02    .487E+01   -.422E+00    .185E-01
 HO2       O            -.148E+02    .336E+01   -.225E+00    .995E-02
 HO2       OH           -.148E+02    .334E+01   -.224E+00    .987E-02
 HO2       HO2          -.158E+02    .357E+01   -.252E+00    .110E-01

 H2O2      H2           -.123E+02    .274E+01   -.146E+00    .651E-02
 H2O2      O2           -.158E+02    .357E+01   -.252E+00    .110E-01
 H2O2      H            -.172E+02    .487E+01   -.423E+00    .185E-01
 H2O2      O            -.148E+02    .336E+01   -.226E+00    .999E-02
 H2O2      OH           -.148E+02    .335E+01   -.225E+00    .991E-02
 H2O2      HO2          -.158E+02    .357E+01   -.252E+00    .110E-01
 H2O2      H2O2         -.158E+02    .357E+01   -.252E+00    .110E-01

 H2O       H2           -.179E+02    .491E+01   -.417E+00    .179E-01
 H2O       O2           -.204E+02    .520E+01   -.430E+00    .174E-01
 H2O       H            -.170E+02    .442E+01   -.311E+00    .116E-01
 H2O       O            -.189E+02    .494E+01   -.411E+00    .172E-01
```

```
 H2O        OH            -.190E+02    .494E+01   -.411E+00    .172E-01
 H2O        HO2           -.197E+02    .499E+01   -.409E+00    .167E-01
 H2O        H2O2          -.197E+02    .499E+01   -.408E+00    .167E-01
 H2O        H2O           -.130E+02    .143E+01    .166E+00   -.121E-01

 AR         H2            -.127E+02    .292E+01   -.170E+00    .758E-02
 AR         O2            -.165E+02    .381E+01   -.282E+00    .122E-01
 AR         H             -.172E+02    .480E+01   -.406E+00    .175E-01
 AR         O             -.156E+02    .366E+01   -.264E+00    .116E-01
 AR         OH            -.155E+02    .364E+01   -.262E+00    .115E-01
 AR         HO2           -.165E+02    .381E+01   -.281E+00    .122E-01
 AR         H2O2          -.165E+02    .381E+01   -.281E+00    .122E-01
 AR         H2O           -.200E+02    .503E+01   -.406E+00    .163E-01
 AR         AR            -.174E+02    .413E+01   -.322E+00    .139E-01



COEFFICIENTS FOR THERMAL DIFFUSION RATIOS

 MAXIMUM FITTING ERROR =    1.562E-02

 H2         H2             .000E+00    .000E+00    .000E+00    .000E+00
 H2         O2             .443E+00    .712E-04   -.385E-07    .686E-11
 H2         H             -.153E+00   -.546E-04    .293E-07   -.487E-11
 H2         O              .416E+00    .110E-04   -.396E-08    .114E-11
 H2         OH             .422E+00    .111E-04   -.402E-08    .116E-11
 H2         HO2            .444E+00    .715E-04   -.386E-07    .689E-11
 H2         H2O2           .446E+00    .717E-04   -.388E-07    .691E-11
 H2         H2O            .602E-01    .562E-03   -.255E-06    .363E-10
 H2         AR             .423E+00    .132E-03   -.712E-07    .120E-10

 H          H2             .153E+00    .546E-04   -.293E-07    .487E-11
 H          O2             .220E+00    .480E-03   -.233E-06    .346E-10
 H          H              .000E+00    .000E+00    .000E+00    .000E+00
 H          O              .270E+00    .362E-03   -.181E-06    .275E-10
 H          OH             .272E+00    .364E-03   -.182E-06    .277E-10
 H          HO2            .221E+00    .481E-03   -.233E-06    .347E-10
 H          H2O2           .221E+00    .482E-03   -.234E-06    .348E-10
 H          H2O           -.142E+00    .767E-03   -.307E-06    .403E-10
 H          AR             .165E+00    .561E-03   -.266E-06    .388E-10
Stop - Program terminated.
```

ΠΑΡΑΡΤΗΜΑ Δ

Κώδικας ρουτίνας TestChemkinInput2.m

Πίνακας 17 – Κώδικας TestChemkinInput2.m

```
function [ output ] = TestChemkinInput2( )
    %define functions to format output the same as fortran
    form = @(x) (strrep(regexprep(regexprep(regexprep(sprintf('%.2E',x)...
        ,'\.',''),'[0-9]','.$&','once'),'...$',regexp(sprintf('%.2E',...
        x*10),'..$','match')),'-00','+00'));
    form2 = @(x) (regexprep(sprintf('%.3E', x),'...$',...
        regexp(sprintf('%.3E', x),'..$','match')));
    %read input data files
    output = ParseChemkinInput(...
        'D:\PhD\Πτυχιακή Εργασία\Matlab\Problem2\chem.inp', ...
        'D:\PhD\Πτυχιακή Εργασία\Matlab\Problem2\therm.dat', ...
        'D:\PhD\Πτυχιακή Εργασία\Matlab\Problem2\tran.dat', '');
    %set transport output file and original transport output file
    outf = 'D:\PhD\Πτυχιακή Εργασία\Matlab\Problem2\tran2.out';
    origf = 'D:\PhD\Πτυχιακή Εργασία\Matlab\Problem2\tran.out';
    %initialize constants
    consts = InitializeConstants();
    %calculate transport parameter fits
    [output.fits status error_in] = ...
        CalculateTransportFits(consts, output, consts.Patm);
    if ~status
        output.status_Ok = false;
        output.error_in = error_in;
        return;
    end
    %output results to file
    fileId = fopen(outf, 'w');
    kk = length(output.species.item);
    %output header
    header = ['\r\n'...
'TRANFT:  Transport property fitting,\r\n'...
'          CHEMKIN-II Version 3.0, November 1990\r\n'...
'          DOUBLE PRECISION\r\n'...
'\r\n\r\n\r\n'...
'OUTPUT FROM THE TRANSPORT PROPERTY FITTING PACKAGE\r\n'...
'\r\n\r\n\r\n'...
'CKLIB: CHEMKIN-III GAS-PHASE CHEMICAL KINETICS LIBRARY,\r\n'...
'DOUBLE PRECISION Vers. 5.12 97/10/28        \r\n'...
'Copyright 1995, Sandia Corporation.\r\n'...
'The U.S. Government retains a limited license in this software.\r\n'...
'\r\n\r\n\r\n'...
'DATA HAS BEEN FIT OVER THE RANGE:  TLOW=%9.2f, THIGH=%9.2f\r\n'...
'\r\n\r\n\r\n'];
    fprintf(fileId, header, output.fits.Tlow, output.fits.Thigh);
    %output transport parameters from database
    fprintf(fileId, 'TRANSPORT PARAMETERS FROM DATA BASE:\r\n\r\n');
    transport_raw = [output.species.item.transport_raw];
    ind = sortrows([[transport_raw.index]' [1:kk]'], 1);
    for k = 1:kk
        fprintf(fileId, '%-80s\r\n', transport_raw(ind(k, 2)).line);
    end
    fprintf(fileId, '\r\n\r\n\r\n');
    %output species conductivities
    fprintf(fileId, ['COEFFICIENTS FOR SPECIES CONDUCTIVITIES\r\n\r\n'...
        ' MAXIMUM FITTING ERROR = %12s\r\n \r\n'],...
        form2(output.fits.emaxL));
    for k = 1:kk
        fprintf(fileId, ' %-10s%12s%12s%12s%12s\r\n', ...
            output.species.item(k).formula, ...
```

```
            form(output.fits.cofL(1, k)), form(output.fits.cofL(2, k)), ...
            form(output.fits.cofL(3, k)), form(output.fits.cofL(4, k)));
    end
    fprintf(fileId, '\r\n\r\n\r\n');
    %output species viscosities
    fprintf(fileId, ['COEFFICIENTS FOR SPECIES VISCOSITIES\r\n\r\n'...
        ' MAXIMUM FITTING ERROR = %12s\r\n \r\n'],...
        form2(output.fits.emaxH));
    for k = 1:kk
        fprintf(fileId, ' %-10s%12s%12s%12s%12s\r\n', ...
            output.species.item(k).formula, ...
            form(output.fits.cofH(1, k)), form(output.fits.cofH(2, k)), ...
            form(output.fits.cofH(3, k)), form(output.fits.cofH(4, k)));
    end
    fprintf(fileId, '\r\n\r\n\r\n');
    %output species diffusion coefficients
    fprintf(fileId, ['COEFFICIENTS FOR SPECIES DIFFUSION COEFFICIENTS'...
        '\r\n\r\n MAXIMUM FITTING ERROR = %12s\r\n'],...
        form2(output.fits.emaxD));
    for k = 1:kk
        fprintf(fileId, ' \r\n');
        for j = 1:k
            fprintf(fileId, ' %-10s%-10s%12s%12s%12s%12s\r\n', ...
                output.species.item(k).formula, ...
                output.species.item(j).formula, ...
                form(output.fits.cofD(1, k, j)), ...
                form(output.fits.cofD(2, k, j)), ...
                form(output.fits.cofD(3, k, j)), ...
                form(output.fits.cofD(4, k, j)));
        end
    end
    fprintf(fileId, '\r\n\r\n\r\n');
    %output thermal diffusion ratios
    fprintf(fileId, ['COEFFICIENTS FOR THERMAL DIFFUSION RATIOS'...
        '\r\n\r\n MAXIMUM FITTING ERROR = %12s\r\n'],...
        form2(output.fits.emaxTD));
    for k = 1:output.fits.nTD
        fprintf(fileId, ' \r\n');
        kind = output.fits.indTD(k);
        for j = 1:kk
            fprintf(fileId, ' %-10s%-10s%12s%12s%12s%12s\r\n', ...
                output.species.item(kind).formula, ...
                output.species.item(j).formula, ...
                form(output.fits.cofTD(1, j, k)), ...
                form(output.fits.cofTD(2, j, k)), ...
                form(output.fits.cofTD(3, j, k)), ...
                form(output.fits.cofTD(4, j, k)));
        end
    end
    fprintf(fileId, 'Stop - Program terminated.\r\n\r\n');
    fclose(fileId);
    %compare transport output file with original output
    text1 = fileread(outf);
    text2 = fileread(origf);
    if strcmp(text1, text2)
        fprintf('\nTransport Files Match\n\n');
    else
        fprintf('\nTransport Files Do Not Match\n\n');
    end
end
```

ΠΑΡΑΡΤΗΜΑ Ε

Δεδομένα και Αποτελέσματα Προβλήματος 1

Προαναμεμιγμένη Φλόγα Μεθανίου-Αέρα

Μηχανισμός Αντίδρασης

Πίνακας 18 – Μηχανισμός Αντίδρασης Φλόγας CH_4-Αέρα

```
ELEMENTS H O C N END
SPECIES CH4 CH3 CH2 CH CH2O HCO CO2 CO H2 H O2 O OH HO2 H2O2 H2O N2
END
REACTIONS
CH3+H+M=CH4+M 8.0E26 -3. 0.

CH4+O2=CH3+HO2 7.9E13 0. 56000.

CH4+H=CH3+H2 2.2E4 3. 8750.

CH4+O=CH3+OH 1.6E6 2.36 7400.

CH4+OH=CH3+H2O 1.6E6 2.1 2460.

CH3+O=CH2O+H 6.8E13 0. 0.

CH3+OH=CH2O+H2 1.0E12 0. 0.

CH3+OH=CH2+H2O 1.5E13 0. 5000.

CH3+H=CH2+H2 9.0E13 0. 15100.

CH2+H=CH+H2 1.4E19 -2. 0.

CH2+OH=CH2O+H 2.5E13 0. 0.

CH2+OH=CH+H2O 4.5E13 0. 3000.

CH+O2=HCO+O 3.3E13 0. 0.

CH+O=CO+H 5.7E13 0. 0.

CH+OH=HCO+H 3.0E13 0. 0.
CH+CO2=HCO+CO 3.4E12 0. 690.

CH2+CO2=CH2O+CO 1.1E11 0. 1000.

CH2+O=CO+H+H 3.0E13 0. 0.

CH2+O=CO+H2 5.0E13 0. 0.

CH2+O2=CO2+H+H 1.6E12 0. 1000.

CH2+O2=CH2O+O 5.0E13 0. 9000.

CH2+O2=CO2+H2 6.9E11 0. 500.
```

```
CH2+O2=CO+H2O 1.9E10 0. -1000.

CH2+O2=CO+OH+H 8.6E10 0. -500.

CH2+O2=HCO+OH 4.3E10 0. -500.

CH2O+OH=HCO+H2O 3.43E9 1.18 -447.

CH2O+H=HCO+H2 2.19E8 1.77 3000.

CH2O+M=HCO+H+M 3.31E16 0. 81000.

CH2O+O=HCO+OH 1.81E13 0. 3082.

HCO+OH=CO+H2O 5.0E12 0. 0.

HCO+M=H+CO+M 1.6E14 0. 14700.

HCO+H=CO+H2 4.0E13 0. 0.

HCO+O=CO2+H 1.0E13 0. 0.

HCO+O2=HO2+CO 3.3E13 -0.4 0.

CO+O+M=CO2+M 3.2E13 0. -4200.
CO+OH=CO2+H 1.51E7 1.3 -758.
CO+O2=CO2+O 1.6E13 0. 41000.

HO2+CO=CO2+OH 5.8E13 0. 22934.

H2+O2=2OH 1.7E13 0. 47780.

OH+H2=H2O+H 1.17E9 1.3 3626.

H+O2=OH+O 5.13E16 -0.816 16507.

O+H2=OH+H 1.8E10 1.0 8826.

H+O2+M=HO2+M 3.61E17 -0.72 0.

H2O/18.6/ CO2/4.2/ H2/2.86/ CO/2.11/ N2/1.26/

OH+HO2=H2O+O2 7.5E12 0. 0.

H+HO2=2OH 1.4E14 0. 1073.

O+HO2=O2+OH 1.4E13 0. 1073.

2OH=O+H2O 6.0E8 1.3 0.

H+H+M=H2+M 1.0E18 -1.0 0.

H+H+H2=H2+H2 9.2E16 -0.6 0.

H+H+H2O=H2+H2O 6.0E19 -1.25 0.

H+H+CO2=H2+CO2 5.49E20 -2.0 0.

H+OH+M=H2O+M 1.6E22 -2.0 0.

H2O/5/

H+O+M=OH+M 6.2E16 -0.6 0.
```

```
H2O/5/

H+HO2=H2+O2 1.25E13 0. 0.

HO2+HO2=H2O2+O2 2.0E12 0. 0.

H2O2+M=OH+OH+M 1.3E17 0. 45500.

H2O2+H=HO2+H2 1.6E12 0. 3800.

H2O2+OH=H2O+HO2 1.0E13 0. 1800.
END
```

Διαδοχικά Αρχεία Εισόδου

Πίνακας 19 – Αρχεία Εισόδου Προβλήματος Καύσης Φλόγας CH4-Αέρα

Αρχείο 1	NSTART=6 NEWTON_ATOL=1E-10 NEWTON_RTOL=1E-4 TSTEP_ATOL=1E-5 TSTEP_RTOL=1E-5 TSTEP_DT=5.0E-7 MDOT0=4.0E-2 DIST=10.0 ADIAB=false FORCEADIAB=true ENRG=false CENDIF=false KEEPFLUX=true GRADVAN=true AREA1=true GSIGMA=0.10 INIT_XSTR=0.0 INIT_XEND=0.3 INIT_XCEN=0.1 INIT_WMIX=1.0 TFIXT=400 REFINE_GRAD=0.9 REFINE_CURV=0.9 XBOUND_0=[0.0950 0 0 0 0 0 0 0 0 0 0.19 0 0 0 0 0 0.715] XBOUND_INF=[0 0 0 0 0 0 0.095 0 0 0 0 0 0 0 0 0 0.190 0.715] INIT_INTM=[0 0.0005 0.0001 0.00001 0.001 0.00001 0 0.08 0.01 0.02 0 0.0001 0.001 0.0001 0.0001 0 0] T_PROFILE=[0.0 298; 0.03 300; 0.05 400; 0.06 766; 0.07 1512; 0.08 1892; 0.09 2000; 0.1 2030; 0.2 2111; 0.35 2190; 10.0 2190]
Αρχείο 2	RESTART=true ADIAB=true ENRG=true REFINE=true TSTEP_DT=1E-6 TSTEP_MAXSTEPS=200
Αρχείο 3	INIT_XSTR=-0.5 INIT_XEND=8.0 REFINE_GRAD=0.45 REFINE_CURV=0.7
Αρχείο 4	INIT_XSTR=-2.0 INIT_XEND=10.0

Τελική Λύσης της Κατάστασης Ισορροπίας

Η τελική λύση της κατάστασης ισορροπίας αποτελείται από 27 κόμβους και είναι η παρακάτω:

Πίνακας 20 – Τελική Λύση σε Κατάσταση Ισορροπίας Προβλήματος Καύσης Φλόγας CH4-Αέρα

r	T	CH4	CH3	CH2	CH	CH2O	HCO	CO2	CO
-2,00000	298	9,500E-02	1,873E-09	1,673E-27	1,272E-29	1,158E-10	1,092E-25	1,973E-09	1,830E-08
-0,50000	298	9,496E-02	5,359E-07	4,680E-25	2,348E-27	4,792E-08	4,482E-23	8,051E-07	5,673E-06
0,00000	318	9,343E-02	3,359E-05	9,785E-19	1,178E-29	4,301E-06	4,540E-17	7,145E-05	3,850E-04
0,05000	400	8,821E-02	1,862E-04	2,251E-14	3,770E-23	3,167E-05	3,662E-13	5,263E-04	2,277E-03
0,05500	439	8,592E-02	2,644E-04	1,286E-12	7,741E-20	4,929E-05	1,483E-11	8,225E-04	3,277E-03
0,06000	500	8,241E-02	3,923E-04	2,829E-11	3,560E-17	8,153E-05	2,258E-10	1,371E-03	4,935E-03
0,06750	641	7,430E-02	7,257E-04	1,709E-09	3,774E-14	1,702E-04	1,063E-08	2,912E-03	9,090E-03
0,07125	739	6,851E-02	1,041E-03	1,970E-08	2,193E-12	2,430E-04	1,160E-07	4,189E-03	1,222E-02
0,07500	856	6,128E-02	1,562E-03	1,491E-07	5,254E-11	3,449E-04	8,234E-07	5,942E-03	1,625E-02
0,08250	1134	4,222E-02	3,460E-03	2,806E-06	4,303E-09	6,553E-04	1,144E-05	1,104E-02	2,707E-02
0,08625	1282	3,101E-02	4,491E-03	1,064E-05	3,727E-08	8,268E-04	2,902E-05	1,448E-02	3,366E-02
0,09000	1428	1,980E-02	4,968E-03	2,915E-05	2,224E-07	9,292E-04	5,454E-05	1,856E-02	4,058E-02
0,09375	1558	1,045E-02	4,256E-03	5,338E-05	8,625E-07	8,760E-04	7,857E-05	2,315E-02	4,672E-02
0,09563	1612	7,006E-03	3,527E-03	5,972E-05	1,356E-06	7,903E-04	8,474E-05	2,557E-02	4,891E-02
0,09750	1658	4,424E-03	2,684E-03	5,884E-05	1,821E-06	6,701E-04	8,422E-05	2,805E-02	5,035E-02
0,10125	1723	1,551E-03	1,170E-03	3,739E-05	1,980E-06	3,741E-04	6,116E-05	3,309E-02	5,069E-02
0,10500	1762	4,616E-04	3,926E-04	1,575E-05	1,221E-06	1,538E-04	3,012E-05	3,796E-02	4,852E-02
0,10875	1787	1,224E-04	1,118E-04	5,135E-06	5,146E-07	4,955E-05	1,095E-05	4,243E-02	4,528E-02
0,11250	1807	2,643E-05	2,562E-05	1,289E-06	1,542E-07	1,214E-05	2,986E-06	4,635E-02	4,201E-02
0,12000	1837	2,033E-06	2,156E-06	1,187E-07	1,662E-08	1,040E-06	3,970E-07	5,249E-02	3,660E-02
0,13500	1883	6,872E-08	8,406E-08	5,050E-09	7,853E-10	3,899E-08	1,134E-07	5,958E-02	3,028E-02
0,15000	1919	2,019E-09	2,833E-09	1,811E-10	2,965E-11	1,405E-09	7,905E-08	6,346E-02	2,683E-02
0,18000	1967	2,656E-11	4,313E-11	2,906E-12	4,857E-13	1,388E-10	5,252E-08	6,682E-02	2,386E-02
0,24000	2025	2,104E-13	3,899E-13	2,722E-14	4,577E-15	9,298E-11	3,027E-08	6,980E-02	2,125E-02
0,30000	2063	1,650E-15	3,378E-15	2,723E-16	8,977E-17	7,332E-11	1,899E-08	7,204E-02	1,929E-02
8,00000	2207	2,152E-17	5,756E-17	1,141E-17	5,276E-18	1,603E-11	1,315E-09	8,264E-02	9,704E-03
10,00000	2207	2,152E-17	5,756E-17	1,141E-17	5,276E-18	1,603E-11	1,315E-09	8,264E-02	9,704E-03
r	H2	H	O2	O	OH	HO2	H2O2	H2O	N2
-2,00000	3,353E-06	2,709E-18	1,900E-01	1,351E-22	2,012E-19	2,188E-13	4,356E-11	1,163E-07	7,150E-01
-0,50000	2,747E-04	1,447E-16	1,899E-01	2,669E-20	4,053E-17	6,769E-11	1,357E-08	3,335E-05	7,148E-01
0,00000	5,120E-03	3,099E-13	1,881E-01	7,565E-17	9,596E-13	9,370E-09	9,252E-07	2,081E-03	7,108E-01
0,05000	1,183E-02	9,921E-11	1,820E-01	2,635E-13	8,719E-10	1,062E-06	5,487E-06	1,116E-02	7,038E-01
0,05500	1,321E-02	4,126E-09	1,792E-01	2,566E-11	1,989E-08	5,208E-06	7,874E-06	1,538E-02	7,019E-01
0,06000	1,476E-02	1,049E-07	1,747E-01	5,421E-10	1,401E-07	1,540E-05	1,175E-05	2,182E-02	6,995E-01
0,06750	1,730E-02	3,190E-06	1,640E-01	2,006E-08	1,729E-06	5,476E-05	2,050E-05	3,628E-02	6,952E-01
0,07125	1,863E-02	1,990E-05	1,560E-01	2,446E-07	8,503E-06	8,948E-05	2,426E-05	4,609E-02	6,929E-01

0,07500	1,998E-02	7,313E-05	1,460E-01	1,701E-06	2,700E-05	1,188E-04	2,575E-05	5,771E-02	6,906E-01
0,08250	2,267E-02	4,085E-04	1,199E-01	2,133E-05	1,306E-04	1,272E-04	1,911E-05	8,558E-02	6,866E-01
0,08625	2,379E-02	8,989E-04	1,043E-01	6,476E-05	3,059E-04	1,109E-04	1,308E-05	1,009E-01	6,851E-01
0,09000	2,445E-02	1,732E-03	8,752E-02	1,656E-04	6,655E-04	8,551E-05	7,030E-06	1,164E-01	6,841E-01
0,09375	2,417E-02	2,988E-03	7,106E-02	3,911E-04	1,349E-03	5,944E-05	3,234E-06	1,308E-01	6,836E-01
0,09563	2,357E-02	3,729E-03	6,358E-02	5,870E-04	1,845E-03	4,830E-05	2,356E-06	1,370E-01	6,837E-01
0,09750	2,269E-02	4,504E-03	5,680E-02	8,591E-04	2,442E-03	3,875E-05	2,008E-06	1,425E-01	6,838E-01
0,10125	2,028E-02	6,010E-03	4,563E-02	1,654E-03	3,868E-03	2,457E-05	2,275E-06	1,511E-01	6,845E-01
0,10500	1,783E-02	7,179E-03	3,769E-02	2,554E-03	5,244E-03	1,667E-05	2,876E-06	1,565E-01	6,854E-01
0,10875	1,593E-02	7,962E-03	3,235E-02	3,263E-03	6,299E-03	1,278E-05	3,362E-06	1,598E-01	6,864E-01
0,11250	1,468E-02	8,429E-03	2,876E-02	3,704E-03	7,014E-03	1,078E-05	3,597E-06	1,616E-01	6,874E-01
0,12000	1,364E-02	8,704E-03	2,482E-02	3,990E-03	7,713E-03	8,932E-06	3,483E-06	1,629E-01	6,891E-01
0,13500	1,328E-02	8,225E-03	2,144E-02	3,810E-03	8,062E-03	7,327E-06	2,756E-06	1,638E-01	6,915E-01
0,15000	1,303E-02	7,512E-03	1,958E-02	3,503E-03	8,103E-03	6,422E-06	2,193E-06	1,648E-01	6,932E-01
0,18000	1,226E-02	6,179E-03	1,752E-02	2,951E-03	7,922E-03	5,407E-06	1,525E-06	1,672E-01	6,953E-01
0,24000	1,072E-02	4,435E-03	1,530E-02	2,224E-03	7,391E-03	4,331E-06	9,294E-07	1,710E-01	6,979E-01
0,30000	9,366E-03	3,282E-03	1,374E-02	1,721E-03	6,814E-03	3,593E-06	6,272E-07	1,738E-01	7,000E-01
8,00000	4,064E-03	5,801E-04	7,049E-03	3,675E-04	3,700E-03	9,099E-07	8,353E-08	1,842E-01	7,077E-01
10,00000	4,064E-03	5,801E-04	7,049E-03	3,675E-04	3,700E-03	9,099E-07	8,353E-08	1,842E-01	7,077E-01

ΠΑΡΑΡΤΗΜΑ ΣΤ

Αντιπροσωπευτικές Ρουτίνες του Matlab

Ρουτίνες Πολυωνυμικής Προσέγγισης Κινητικών Ιδιοτήτων

Πίνακας 21 – Κώδικας CalculateBinaryDiffusionCoefficientFits.m

```
%FORTRAN DIFFIT parameters
%  KK: number of species
%      --> kk = length(input.species.item)
%  NT: 50, temperature discretization
%      --> nT = 50
%  NO: 4, maximum degree of polynomial fit
%      --> no = 4
%  KDIM: 99, maximum number of species allowed
%      --> not used
%  LOUT: 6, device to send output, 6 is the screen
%      --> not used
%  WT: array (1xKK) of molecular weights of species
%      --> input.species.item.weight
%  SIG: array (1xKK) of Lennard-Jones collision diameters σ of species
%      --> input.species.item.coll_dia
%  EPS: array (1xKK) of Lennard-Jones potential well depths ε/kB of species
%      --> input.species.item.well_dep
%  DIP: array (1xKK) of dipole moments μ of species
%      --> input.species.item.dip_mom
%  POL: array (1xKK) of polarizabilities α of species
%      --> input.species.item.polariz
%  P: pressure
%      --> P
%  TLOW: low temperature from thermodynamic data
%      --> Tlow = input.thermo.base.temperatures.low
%  DT: (THIGH-TLOW)/(NT-1), temperature step for discretization
%      --> dT = (Thigh-Tlow)/(nT-1)
%  ALOGT: output array (1xNT) holding the points of the LOG of temperature
%      --> alogT(1, nT)
%  FITRES: output array (1xNT) of values of the fit
%      --> fitres
%  FITWT: output array of fit weights
%      --> fitwt(1, nT)
%  FITWRK: working array of at least 165 items
%      --> fitwrk
%  XD: output array (1xNT) of calculated binary diffusion coefficients
%      --> xD
%  COFD: output array (NOxKDIMxKDIM) holding Taylor series form
%    coefficients calculated from the polynomial ones
%      --> cofD
%  EMAXD: input and output of the RMS error of the polynomial fit,
%    initially is set to zero
%      --> emaxd = 0.0
function [cofD emaxD status] = ...
    CalculateBinaryDiffusionCoefficientFits(consts, input, P)
    %initialize constants
    status = true;
    epsil = 0.0;  %flag for fit routine
    %below dipmin dipole momentum is considered zero, so the molecule is
    %not polar
    dipmin = consts.dipmin;
    %number of steps of temperature discretization for fit algorithm
    nT = consts.nT;
    %maximum degree of polynomial fit
    no = consts.no;
    %RMS error of the polynomial fit
    emaxD = 0.0;
    %convertor to convert Debyes cubic Angstroms to CGS
    fcon = 0.5*1.0E-18^2*1.0E8^3/consts.kB;

    %get low and high temperatures
```

```
    temperatures = [input.species.item.temperatures];
    Tlow = max([temperatures.low]);
    Thigh = min([temperatures.hi]);
    dT = (Thigh-Tlow)/(nT-1);
    %calculate
    alogT = zeros(1, nT);
    for n = 1:nT
        T = Tlow+(n-1)*dT;
        alogT(n) = log(T);
    end
    spec = input.species.item;
    kk = length(spec);
    xD = zeros(1, nT);
    fitwt = zeros(1, nT);
    cofD = zeros(no, kk, kk);
    for j = 1:kk
        for k = 1:j
            sigm = 0.5*(spec(j).coll_dia+spec(k).coll_dia);
            epsm = sqrt(spec(j).well_dep*spec(k).well_dep);
            if spec(j).dip_mom < dipmin && spec(k).dip_mom > dipmin
                dst = 0;
                xi = 1.0+spec(j).polariz*fcon*spec(k).dip_mom^2*...
                    sqrt(spec(k).well_dep/spec(j).well_dep)/...
                    (2.0*spec(j).coll_dia^3*spec(k).well_dep*...
                    spec(k).coll_dia^3);
            elseif spec(j).dip_mom > dipmin && spec(k).dip_mom < dipmin
                dst = 0;
                xi = 1.0+spec(k).polariz*fcon*spec(j).dip_mom^2*...
                    sqrt(spec(j).well_dep/spec(k).well_dep)/...
                    (2.0*spec(k).coll_dia^3*spec(j).well_dep*...
                    spec(j).coll_dia^3);
            else
                dst = fcon*spec(j).dip_mom*spec(k).dip_mom/(epsm*sigm^3);
                xi = 1.0;
            end
            sigm = sigm*xi^(-1.0/6.0);
            epsm = epsm*xi^2;
            help1 = (spec(j).weight+spec(k).weight)/...
                (2*spec(j).weight*spec(k).weight);
            for n = 1:nT
                T = Tlow+(n-1)*dT;
                Tr = T/epsm;
                help2 = 0.002628*sqrt(help1*T^3);
                xD(n) = help2*consts.Patm/(CalculateOmega12(1,Tr,dst)*...
                    sigm^2*P);

                xD(n) = xD(n)*CalculateBinaryDiffusionCoefficientD12(...

                    spec(j).weight, spec(k).weight, T, ...
                    spec(j).well_dep, spec(k).well_dep, ...
                    spec(j).coll_dia, spec(k).coll_dia, dst);
                xD(n) = log(xD(n));
            end
            fitwt(1) = -1.0;
            epsl = epsil;
            [ndeg epsl fitres iErr fitwrk] = PolynomialsFit(nT, alogT, ...
                xD, fitwt, no-1, epsl);
            emaxD = max(emaxD, epsl);
            if iErr ~= 1, status = false; return; end;
            ccc = 0.0;
            [tc fitwrk] = CalculateTaylorCoefficients(ndeg, ...
                ccc, fitwrk);
            for n = 1:length(tc), cofD(n, k, j) = tc(n); end
        end
    end
    %fill out the lower triangle
    for k = 1:kk-1
        for j = k+1:kk
            for n = 1:no
                cofD(n, j, k) = cofD(n, k, j);
            end
        end
    end
end
```

Πίνακας 22 – Κώδικας CalculateThermalDiffusionRatioFits.m

```
%FORTRAN THMFIT parameters
%  KK: number of species
%      --> kk = length(input.species.item)
%  NT: 50, temperature discretization
%      --> nT = 50
%  NO: 4, maximum degree of polynomial fit
%      --> no = 4
%  KDIM: 99, maximum number of species allowed
%      --> not used
%  LOUT: 6, device to send output, 6 is the screen
%      --> not used
%  WT: array (1xKK) of molecular weights of species
%      --> input.species.item.weight
%  SIG: array (1xKK) of Lennard-Jones collision diameters σ of species
%      --> input.species.item.coll_dia
%  EPS: array (1xKK) of Lennard-Jones potential well depths ε/kB of species
%      --> input.species.item.well_dep
%  DIP: array (1xKK) of dipole moments μ of species
%      --> input.species.item.dip_mom
%  POL: array (1xKK) of polarizabilities α of species
%      --> input.species.item.polariz
%  TLOW: low temperature from thermodynamic data
%      --> Tlow = input.thermo.base.temperatures.low
%  DT: (THIGH-TLOW)/(NT-1), temperature step for discretization
%      --> dT = (Thigh-Tlow)/(nT-1)
%  ALOGT: output array (1xNT) holding the points of the LOG of temperature
%      --> alogT(1, nT)
%  FITRES: output array (1xNT) of values of the fit
%      --> fitres
%  FITWT: output array of fit weights
%      --> fitwt(1, nT)
%  FITWRK: working array of at least 165 items
%      --> fitwrk
%  XD: output array (1xNT) of calculated thermal diffusion rations
%      --> xTD
%  NLITE: output number of species having thermal diffusion ratio value
%      --> nTD
%  KTDIF: output array(1xNLITE) holding indices of species having thermal
%    diffusion ratio value
%      --> indTD
%  COFTD: output array (NOxKDIMxKDIM) holding Taylor series form thermal
%    diffusion rations calculated from the polynomial ones
%      --> cofTD
%  EMAXTD: input and output of the RMS error of the polynomial fit of
%    thermal diffusion rations initially is set to zero
%      --> emaxTD = 0.0
function [cofTD nTD indTD emaxTD status] = ...
    CalculateThermalDiffusionRatioFits(consts, input)
    %initialize constants
    status = true;
    fitast = [.1106910525E+01 -.7065517161E-02 -.1671975393E-01 ...

        .1188708609E-01 .7569367323E-03 -.1313998345E-02 .1720853282E-03];

    fitbst = [.1199673577E+01 -.1140928763E+00 -.2147636665E-02 ...

        .2512965407E-01 -.3030372973E-02 -.1445009039E-02 .2492954809E-03];

    fitcst = [.8386993788E+00 .4748325276E-01 .3250097527E-01 ...

        -.1625859588E-01 -.2260153363E-02 .1844922811E-02 ...

        -.2115417788E-03];
    epsil = 0.0;  %flag for fit routine
    %number of steps of temperature discretization for fit algorithm
    nT = consts.nT;
    %number of species with atomic weight below 5 that have thermal
    %diffusion ratios
    nTD = 0;
    %maximum degree of polynomial fit
    no = consts.no;
    %RMS error of the polynomial fit
    emaxTD = 0.0;
```

```
    %get low and high temperatures
    temperatures = [input.species.item.temperatures];
    Tlow = max([temperatures.low]);
    Thigh = min([temperatures.hi]);
    dT = (Thigh-Tlow)/(nT-1);
    %calculate
    spec = input.species.item;
    kk = length(spec);
    xTD = zeros(1, nT);
    fitwt = zeros(1, nT);
    cofTD = zeros(no, kk);
    aT = zeros(1, nT);
    for n = 1:nT
        T = Tlow+(n-1)*dT;
        aT(n) = T;
    end
    for j = 1:kk
        if spec(j).weight <= consts.Wlight
            nTD = nTD + 1;
            indTD(nTD) = j;
            epsj = spec(j).well_dep*consts.kB;
            sigj = spec(j).coll_dia*1.0E-8;

            polj = spec(j).polariz*1.0E-24;

            poljst = polj/sigj^3;
            for k = 1:kk
                epsk = spec(k).well_dep*consts.kB;
                sigk = spec(k).coll_dia*1.0E-8;

                dipk = spec(k).dip_mom*1.0E-18;
                dipkst = dipk/sqrt(epsk*sigk^3);
                ekoej = epsk/epsj;
                tse = 1.0+0.25*poljst*dipkst^2*sqrt(ekoej);
                eok = tse^2*sqrt(spec(j).well_dep*spec(k).well_dep);
                wtkj = (spec(k).weight-spec(j).weight)/...
                    (spec(k).weight+spec(j).weight);
                for n = 1:nT
                    tslog = log(aT(n)/eok);
                    astar = Horner(7, fitast, tslog);
                    bstar = Horner(7, fitbst, tslog);
                    cstar = Horner(7, fitcst, tslog);
                    xTD(n) = 7.5*wtkj*(2.0*astar+5.0)*(6.0*cstar-5.0)/...
                        (astar*(16.0*astar-12.0*bstar+55.0));
                end
                fitwt(1) = -1.0;
                epsl = epsil;
                [ndeg epsl fitres iErr fitwrk] = PolynomialsFit(nT, aT, ...
                    xTD, fitwt, no-1, epsl);
                emaxTD = max(emaxTD, epsl);
                if iErr ~= 1, status = false; return; end;
                ccc = 0.0;
                [tc fitwrk] = CalculateTaylorCoefficients(ndeg, ...
                    ccc, fitwrk);
                for n = 1:length(tc), cofTD(n, k, nTD) = tc(n); end
            end
        end
    end
end
```

Πίνακας 23 – Κώδικας CalculateThermalConductivityandViscosityFits.m

```
%FORTRAN LAMFIT parameters
%  KK: number of species
%      --> kk = length(input.species.item)
%  NT: 50, temperature discretization
%      --> nT = 50
%  NO: 4, maximum degree of polynomial fit
%      --> no = 4
%  LOUT: 6, device to send output, 6 is the screen
%      --> not used
%  WT: array (1xKK) of molecular weights of species
%      --> input.species.item.weight
```

```
%  SIG: array (1xKK) of Lennard-Jones collision diameters σ of species
%      --> input.species.item.coll_dia
%  EPS: array (1xKK) of Lennard-Jones potential well depths ε/kB of species
%      --> input.species.item.well_dep
%  DIP: array (1xKK) of dipole moments μ of species
%      --> input.species.item.dip_mom
%  ZROT: array (1xKK) of rotational relaxation collision numbers of species
%      --> input.species.item.zrot
%  NLIN: array (1xKK) of geometrical configuration of species
%      --> nlin = 99000
%  P: pressure
%      --> P
%  TLOW: low temperature from thermodynamic data
%      --> Tlow = input.thermo.base.temperatures.low
%  DT: (THIGH-TLOW)/(NT-1), temperature step for discretization
%      --> dT = (Thigh-Tlow)/(nT-1)
%  ALOGT: output array (1xNT) holding the points of the LOG of temperature
%      --> alogT(1, nT)
%  FITRES: output array (1xNT) of values of the fit
%      --> fitres
%  FITWT: output array of fit weights
%      --> fitwt(1, nT)
%  FITWRK: working array of at least 165 items
%      --> fitwrk
%  XLA: output array (1xNT) of calculated thermal conductivities λk
%      --> xL
%  XETA: output array (1xNT) of calculated viscosities ηk
%      --> xH
%  CV: array(1xKK) of specific heats
%      --> cv calculated
%  ICKWRK: input and output (1x10000) working integer array
%      --> ickwrk
%  RCKWRK: input and output (1x10000) working real array
%      --> rckwrk
%  COFLAM: output array (NOxKDIM) holding Taylor series form thermal
%    conductivity coefficients calculated from the polynomial ones
%      --> cofL
%  COFETA: output array (NOxKDIM) holding Taylor series form viscosity
%    coefficients calculated from the polynomial ones
%      --> cofH
%  EMAXL: input and output of the RMS error of the polynomial fit of
%    thermal conductivities initially is set to zero
%      --> emaxL = 0.0
%  EMAXE: input and output of the RMS error of the polynomial fit of
%    viscosities initially is set to zero
%      --> emaxH = 0.0
function [cofL cofH emaxL emaxH status] = ...
    CalculateThermalConductivityandViscosityFits(consts, input, P)
    %initialize constants
    status = true;
    Ru = consts.R;
    epsil = 0.0;  %flag for fit routine
    %number of steps of temperature discretization for fit algorithm
    nT = consts.nT;
    %maximum degree of polynomial fit
    no = consts.no;
    %RMS error of the polynomial fit
    emaxL = 0.0;
    emaxH = 0.0;
    %convertor to convert Debyes cubic Angstroms to CGS
    fcon = 0.5*1.0E-18^2*1.0E8^3/consts.kB;

    %calculate factor of pressure
    pfac = consts.Patm/P;
    %get low and high temperatures
    temperatures = [input.species.item.temperatures];
    Tlow = max([temperatures.low]);
    Thigh = min([temperatures.hi]);
    dT = (Thigh-Tlow)/(nT-1);
    %calculate
    spec = input.species.item;
    kk = length(spec);
    xL = zeros(1, nT);
    xH = zeros(1, nT);
    fitwt = zeros(1, nT);
    cofL = zeros(no, kk);
```

```
    cofH = zeros(no, kk);
    alogT = zeros(1, nT);
    for k = 1:kk
        dst = fcon*spec(k).dip_mom^2/(spec(k).well_dep*spec(k).coll_dia^3);
        helpe = 2.6693E-5*sqrt(spec(k).weight)/spec(k).coll_dia^2;
        helpd = 2.6280E-3*pfac/(sqrt(spec(k).weight)*spec(k).coll_dia^2);
        for n = 1:nT
            T = Tlow+(n-1)*dT;
            prut = P/(Ru*T);
            Tr = T/spec(k).well_dep;
            cv = SpecificHeatCapacity(consts, input, T);
            alogT(n) = log(T);
            dii = sqrt(T^3)*helpd/CalculateOmega12(1, Tr, dst);
            xH(n) = sqrt(T)*helpe/CalculateOmega12(2, Tr, dst);
            rodet = dii*spec(k).weight*prut/xH(n);
            aa = 2.5-rodet;
            park = CalculateRotationalCollisionNumbers(T,spec(k).well_dep);
            if spec(k).geo_conf == 2
                bb = spec(k).zrot*park+2.0*(5.0/2.0+rodet)/pi;
                croct = 1.0;
            else
                bb = spec(k).zrot*park+2.0*(5.0/3.0+rodet)/pi;
                croct = 2.0/3.0;
            end
            ftra = 2.5*(1.0-2.0*croct*aa/(bb*pi));
            frot = rodet*(1.0+2.0*aa/(bb*pi));
            fvib = rodet;
            if spec(k).geo_conf == 0
                faktor = 5.0/2.0*1.5*Ru;
            elseif spec(k).geo_conf == 1
                faktor = (ftra*1.5+frot)*Ru+fvib*(cv(k)-2.5*Ru);
            elseif spec(k).geo_conf == 2
                faktor = (ftra+frot)*1.5*Ru+fvib*(cv(k)-3.0*Ru);
            end
            xL(n) = log(xH(n)/spec(k).weight*faktor);
            xH(n) = log(xH(n));
        end
        %fit conductivity
        fitwt(1) = -1;
        epsl = epsil;
        [ndeg epsl fitres iErr fitwrk] = PolynomialsFit(nT, alogT, ...
                xL, fitwt, no-1, epsl);
        emaxL = max(emaxL, epsl);
        if iErr ~= 1, status = false; return; end;
        ccc = 0;
        [tc fitwrk] = CalculateTaylorCoefficients(ndeg, ccc, fitwrk);
        for n = 1:length(tc), cofL(n, k) = tc(n); end
        %fit viscosity
        fitwt(1) = -1;
        epsl = epsil;
        [ndeg epsl fitres iErr fitwrk] = PolynomialsFit(nT, alogT, ...
                xH, fitwt, no-1, epsl);
        emaxH = max(emaxH, epsl);
        if iErr ~= 1, status = false; return; end;
        ccc = 0;
        [tc fitwrk] = CalculateTaylorCoefficients(ndeg, ccc, fitwrk);
        for n = 1:length(tc), cofH(n, k) = tc(n); end
    end
end
```

Πίνακας 24 – Κώδικας CalculateBinaryDiffusionCoefficientD12.m

```
function [D12] = CalculateBinaryDiffusionCoefficientD12(w1, w2, T, eps1,...
    eps2, sig1, sig2, dst)
    %calculate constants
    sumw = w1+w2;
    sig12 = 0.5*(sig1+sig2);
    Tr11 = T/eps1;
    Tr22 = T/eps2;
    Tr12 = T/sqrt(eps1*eps2);
    %calculate reduced collision integrals
    om2f11 = CalculateOmegaXX(2, Tr11, dst);
    om2f22 = CalculateOmegaXX(2, Tr22, dst);
```

```
    om1f12 = CalculateOmegaXX(1, Tr12, dst);
    om2f12 = CalculateOmegaXX(2, Tr12, dst);
    bst12 = CalculateOmegaXX(3, Tr12, dst);
    cst12 = CalculateOmegaXX(4, Tr12, dst);
    ast12 = om2f12/om1f12;
    %calculate D constants
    H = (sig1/sig2)^2*sqrt(2.0*w2/sumw)*2.0*w1^2/(sumw*w2);
    P1 = H*om2f11/om1f12;
    H = (sig2/sig1)^2*sqrt(2.0*w1/sumw)*2.0*w2^2/(sumw*w1);
    P2 = H*om2f22/om1f12;
    P12 = 15.0*((w1-w2)/sumw)^2+8.0*w1*w2*ast12/sumw^2;
    H = 2.0/(w2*sumw)*sqrt(2.0*w2/sumw)*om2f11/om1f12*(sig1/sig12)^2;
    Q1 = ((2.5-1.2*bst12)*w1^2+3.0*w2^2+1.6*w1*w2*ast12)*H;
    H = 2.0/(w1*sumw)*sqrt(2.0*w1/sumw)*om2f22/om1f12*(sig2/sig12)^2;
    Q2 = ((2.5-1.2*bst12)*w2^2+3.0*w1^2+1.6*w1*w2*ast12)*H;
    H = ((w1-w2)/sumw)^2*(2.5-1.2*bst12)*15.0+4.0*w1*w2*ast12/sumw^2*...
        (11.0-2.4*bst12);
    Q12 = H+1.6*sumw*om2f11*om2f22/(sqrt(w1*w2)*om1f12^2)*...
        sig1^2*sig2^2/sig12^4;
    %calculate binary diffusion coefficient
    D12 = ((6.0*cst12-5.0)^2/10.0)*(P1+P2+P12)/(Q1+Q2+Q12)+1;
end
```

Πίνακας 25 – Κώδικας CalculateOmega12.m

```
function [output] = CalculateOmega12(N, Tr, Dr)
    %initialize tables and arrays
    vert = zeros(1, 3);
    arg = zeros(1, 3);
    val = zeros(1, 3);
    tstern = [.1 .2 .3 .4 .5 .6 .7 .8 .9 1. 1.2 1.4 1.6 1.8 2. 2.5 3. ...

        3.5 4. 5. 6. 7. 8. 9. 10. 12. 14. 16. 18. 20. 25. 30. 35. 40. ...

        50. 75. 100.];

    delta = [0. .25 .5 .75 1. 1.5 2. 2.5];

    O = [4.008 4.002 4.655 5.52 6.454 8.214 9.824 11.31;

        3.130 3.164 3.355 3.721 4.198 5.23 6.225 7.160;

        2.649 2.657 2.77 3.002 3.319 4.054 4.785 5.483;

        2.314 2.32 2.402 2.572 2.812 3.386 3.972 4.539;

        2.066 2.073 2.14 2.278 2.472 2.946 3.437 3.918;

        1.877 1.885 1.944 2.06 2.225 2.628 3.054 3.747;

        1.729 1.738 1.79 1.893 2.036 2.388 2.763 3.137;

        1.6122 1.622 1.67 1.76 1.886 2.198 2.535 2.872;

        1.517 1.527 1.572 1.653 1.765 2.044 2.35 2.657;

        1.44 1.45 1.49 1.564 1.665 1.917 2.196 2.4780;

        1.3204 1.33 1.364 1.425 1.51 1.72 1.956 2.199;

        1.234 1.24 1.272 1.324 1.394 1.573 1.777 1.99;

        1.168 1.176 1.202 1.246 1.306 1.46 1.64 1.827;

        1.1166 1.124 1.146 1.185 1.237 1.372 1.53 1.7;

        1.075 1.082 1.102 1.135 1.181 1.3 1.441 1.592;

        1.0006 1.005 1.02 1.046 1.08 1.17 1.278 1.397;
```

```
        .95 .9538 .9656 .9852 1.012 1.082 1.168 1.265;
        .9131 .9162 .9256 .9413 .9626 1.019 1.09 1.17;
        .8845 .8871 .8948 .9076 .9252 .972 1.03 1.098;
        .8428 .8446 .850 .859 .8716 .9053 .9483 .9984;
        .813 .8142 .8183 .825 .8344 .8598 .8927 .9316;
        .7898 .791 .794 .7993 .8066 .8265 .8526 .8836;
        .7711 .772 .7745 .7788 .7846 .8007 .822 .8474;
        .7555 .7562 .7584 .7619 .7667 .78 .7976 .8189;
        .7422 .743 .7446 .7475 .7515 .7627 .7776 .796;
        .72022 .7206 .722 .7241 .7271 .7354 .7464 .76;
        .7025 .703 .704 .7055 .7078 .7142 .7228 .7334;
        .68776 .688 .6888 .6901 .6919 .697 .704 .7125;
        .6751 .6753 .676 .677 .6785 .6827 .6884 .6955;
        .664 .6642 .6648 .6657 .6669 .6704 .6752 .681;
        .6414 .6415 .6418 .6425 .6433 .6457 .649 .653;
        .6235 .6236 .6239 .6243 .6249 .6267 .629 .632;
        .60882 .6089 .6091 .6094 .61 .6112 .613 .6154;
        .5964 .5964 .5966 .597 .5972 .5983 .600 .6017;
        .5763 .5763 .5764 .5766 .5768 .5775 .5785 .58;
        .5415 .5415 .5416 .5416 .5418 .542 .5424 .543;
        .518 .518 .5182 .5184 .5184 .5185 .5186 .5187];
    P = [4.1 4.266 4.833 5.742 6.729 8.624 10.34 11.890;
        3.263 3.305 3.516 3.914 4.433 5.57 6.637 7.618;
        2.84 2.836 2.936 3.168 3.511 4.329 5.126 5.874;
        2.531 2.522 2.586 2.749 3.004 3.64 4.282 4.895;
        2.284 2.277 2.329 2.46 2.665 3.187 3.727 4.249;
        2.084 2.081 2.13 2.243 2.417 2.862 3.329 3.786;
        1.922 1.924 1.97 2.072 2.225 2.641 3.028 3.435;
        1.7902 1.795 1.84 1.934 2.07 2.417 2.788 3.156;
        1.682 1.689 1.733 1.82 1.944 2.258 2.596 2.933;
        1.593 1.60 1.644 1.725 1.84 2.124 2.435 2.746;
        1.455 1.465 1.504 1.574 1.67 1.913 2.181 2.45;
        1.355 1.365 1.4 1.461 1.544 1.754 1.989 2.228;
        1.28 1.289 1.321 1.374 1.447 1.63 1.838 2.053;
```

```
        1.222 1.231 1.26 1.306 1.37 1.532 1.718 1.912;

        1.176 1.184 1.209 1.25 1.307 1.45 1.618 1.795;

        1.0933 1.1 1.119 1.15 1.193 1.304 1.435 1.578;

        1.039 1.044 1.06 1.083 1.117 1.204 1.31 1.428;

        .9996 1.004 1.016 1.035 1.062 1.133 1.22 1.32;

        .9699 .9732 .983 .9991 1.021 1.08 1.153 1.236;

        .9268 .9291 .936 .9473 .9628 1.005 1.058 1.12;

        .8962 .8979 .903 .9114 .923 .9545 .9955 1.044;

        .8727 .8741 .878 .8845 .8935 .918 .9505 .9893;

        .8538 .8549 .858 .8632 .8703 .890 .9164 .9482;

        .8379 .8388 .8414 .8456 .8515 .868 .8895 .916;

        .8243 .8251 .8273 .8308 .8356 .8493 .8676 .89;

        .8018 .8024 .8039 .8065 .810 .820 .8337 .8504;

        .7836 .784 .7852 .7872 .7899 .7976 .808 .8212;

        .7683 .7687 .7696 .771 .7733 .7794 .788 .7983;

        .7552 .7554 .7562 .7575 .7592 .764 .771 .7797;

        .7436 .7438 .7445 .7455 .747 .7512 .757 .7642;

        .71982 .72 .7204 .7211 .7221 .725 .7289 .7339;

        .701 .7011 .7014 .702 .7026 .7047 .7076 .7112;

        .68545 .6855 .686 .686 .6867 .6883 .6905 .693;

        .6723 .6724 .6726 .673 .6733 .6745 .676 .6784;

        .651 .651 .6512 .6513 .6516 .6524 .6534 .6546;

        .614 .614 .6143 .6145 .6147 .6148 .6148 .6147;

        .5887 .5889 .5894 .59 .5903 .5901 .5895 .5885];

    %check for invalid reduced temperature Tr or reduced dipole moment

    if Dr < -.00001 || Dr > 2.5 || Tr < .09 || Tr > 500 || ...

            (abs(Dr) > 1.0E-5 && Tr > 75.0)
        output = NaN;
        return;
    end
    if Tr > 75.0
        if N == 1
            output = 0.623-0.136E-2*Tr+0.346E-5*Tr*Tr-0.343E-8*Tr*Tr*Tr;

        elseif N == 2

            output = 0.703-0.146E-2*Tr+0.357E-5*Tr*Tr-0.343E-8*Tr*Tr*Tr;

        end
        return;
    end
    if Tr <= 0.2
        ii = 2;
```

```
    else
        ii = 37;
        for i = 2:36
            if tstern(i-1) < Tr && tstern(i) >= Tr, ii = i; end
        end
    end
    if abs(Dr) >= 1.0E-5
        if Dr <= 0.25
            kk = 2;
        else
            kk = 7;
            for k = 2:7
                if delta(k-1) < Dr && delta(k) >= Dr, kk = k; end
            end
        end
        for i = 1:3
            for k = 1:3
                arg(k) = delta(kk-2+k);

                if N == 1
                    val(k) = O(ii-2+i, kk-2+k);
                elseif N == 2
                    val(k) = P(ii-2+i, kk-2+k);
                end
            end
            vert(i) = QuadraticInterpolation(Dr, arg, val);
        end
        for i = 1:3
            arg(i) = tstern(ii-2+i);
        end
        output = QuadraticInterpolation(Tr, arg, vert);
    else
        for i = 1:3
            arg(i) = tstern(ii-2+i);
            if N == 1

                val(i) = O(ii-2+i, 1);

            else
                val(i) = P(ii-2+i, 1);
            end
        end
        output = QuadraticInterpolation(Tr, arg, val);
    end
end
```

Πίνακας 26 – Κώδικας CalculateOmegaXX.m

```
function [output] = CalculateOmegaXX(N, Tr, Dr)
    if N == 1 || N == 2
        output = CalculateOmega12(N, Tr, Dr);
    elseif N == 3
        if Tr <= 5.0
            output = 1.36-0.223*Tr+0.0613*Tr*Tr-0.00554*Tr*Tr*Tr;
        else
            output = 1.095;
        end
    elseif N == 4
        if Tr <= 5.0
            output = 0.823+0.0128*Tr+0.0112*Tr*Tr-0.00193*Tr*Tr*Tr;
        else
            output = .9483;
        end
    else
        output = NaN;
    end
end
```

Πίνακας 27 – Κώδικας CalculateRotationalCollisionNumbers.m

```
%this is function PARKER of CHEMKIN TRANSPORT module
function [p] = CalculateRotationalCollisionNumbers(T, eps)
    %definition of constants
    PI32O2 = 2.7842;
    P2O4P2 = 4.4674;
    PI32 = 5.5683;
    d = eps/T;
    dr = eps/298.0;
    p = (1.0+PI32O2*sqrt(dr)+P2O4P2*dr+PI32*dr^1.5)/...

        (1.0+PI32O2*sqrt(d)+P2O4P2*d+PI32*d^1.5);

end
```

Ρουτίνες Υπολογισμού Θερμοδυναμικών και Κινητικών Ιδιοτήτων

Πίνακας 28 – Κώδικας GetKfi.m

```
function [Kfi] = GetKfi(consts, input, chem, singleYT, ii)
    if singleYT
        istr = ii;
        iend = ii;
    else
        istr = 1;
        iend = chem.N;
    end
    iwid = iend-istr+1;
    %define constants
    Rc = consts.Rc;
    %do calculation
    Kfi = zeros(iwid, chem.II);
    for i = 1:chem.II
        Arr = input.reactions.item(i).Arrhenius;
        if input.reactions.item(i).is_simple
            %if reaction is not pressure dependent and does not require
            %any special formula, then calculate Kfis from the simple
            %Arrhenius formula
            for n = istr:iend
                Kfi(n-istr+1, i) = Arr.A*chem.TT(n)^Arr.B*...

                    exp(-Arr.E/Rc/chem.TT(n));
            end
        %check if reaction is pressure dependent
        elseif input.reactions.item(i).pressure_dependent
            Arr = input.reactions.item(i).press;
            for n = istr:iend
                K0 = Arr.A0*chem.TT(n)^Arr.B0*exp(-Arr.E0/Rc/chem.TT(n));
                Kinf = Arr.Ainf*chem.TT(n)^Arr.Binf*...
                    exp(-Arr.Einf/Rc/chem.TT(n));
                %calculate reduced pressure Pr = K0/Kinf*[M]
                Pr = K0*chem.EM(n,i)/Kinf;
                %calculate F based on Lindemann, Troe or SRI formulas
                if input.reactions.item(i).press.is_lind
                    F = 1;
                elseif input.reactions.item(i).press.is_troe
                    %calculation of factor F based on Troe formula
                    params = input.reactions.item(i).press.troe;
                    Fcent = (1-params.a)*exp(-chem.TT(n)/params.T3)+...
                        params.a*exp(-chem.TT(n)/params.T1);
                    if ~isnan(params.T2)
                        Fcent = Fcent + exp(-params.T2/chem.TT(n));
                    end
                    c = -0.4-0.67*log10(Fcent);
                    nn = 0.75-1.27*log10(Fcent);
                    d = 0.14;
                    F = 10^(log10(Fcent)/(1+((log10(Pr)+c)/(nn-d*...
                        (log10(Pr)+c)))^2));
                elseif input.reactions.item(i).press.is_sri
                    %calculation of factor F based on SRI formula
                    params = input.reactions.item(i).press.sri;
```

```
                    X = 1/(1+log10(Pr)^2);
                    F = params.d*(params.a*exp(-params.b/chem.TT(n))+...
                        exp(-chem.TT(n)/params.c))^X*chem.TT(n)^params.e;
                else
                    F = 1;
                end
                if input.reactions.item(i).press.is_falloff
                    Kfi(n-istr+1, i) = Kinf*(Pr/(1+Pr))*F;
                else
                    Kfi(n-istr+1, i) = K0*(1/(1+Pr))*F;
                end
            end
        elseif input.reactions.item(i).nopress.is_lt
            %use Landau-Teller formulation
            params = input.reactions.item(i).nopress.lt;
            for n = istr:iend
                Kfi(n-istr+1, i) = Arr.A*chem.TT(n)^Arr.B*...
                    exp(-Arr.E/Rc/chem.TT(n)+...
                    params.B/chem.TT(n)^(1/3)+params.C/chem.TT(n)^(2/3));
            end
        elseif input.reactions.item(i).nopress.is_jan
            %use JAN formulation
            params = input.reactions.item(i).nopress.jan;
            for n = istr:iend
                fit = 0;
                for j = 1:9, fit = fit+params(j)*log(chem.TT(n))^j; end
                Kfi(n-istr+1, i) = Arr.A*chem.TT(n)^Arr.B*...

                    exp(Arr.E/chem.TT(n)+fit);
            end
        elseif input.reactions.item(i).nopress.is_fit1
            %use FIT1 formulation
            params = input.reactions.item(i).nopress.fit1;
            for n = istr:iend
                fit = 0;
                for j = 1:4, fit = fit+params(j)/chem.TT(n)^j; end
                Kfi(n-istr+1, i) = Arr.A*chem.TT(n)^Arr.B*exp(fit);
            end
        else
            %if reaction is not pressure dependent and does not require
            %any special formula, then calculate Kfis from the simple
            %Arrhenius formula
            for n = istr:iend
                Kfi(n-istr+1, i) = Arr.A*chem.TT(n)^Arr.B*...

                    exp(-Arr.E/Rc/chem.TT(n));
            end
        end
    end
end
```

Πίνακας 29 – Κώδικας GetKri.m

```
function [Kri] = GetKri(consts, input, chem, singleYT, ii)
    if singleYT
        istr = ii;
        iend = ii;
    else
        istr = 1;
        iend = chem.N;
    end
    iwid = iend-istr+1;
    %define constants
    Rc = consts.Rc;
    %do calculation
    Kri = zeros(iwid, chem.II);
    for i = 1:chem.II
        %detect if reverse rate parameters are defined
        if input.reactions.item(i).has_rev
            Arr = input.reactions.item(i).rev;
            if input.reactions.item(i).nopress.is_lt
                %calculate Kri according to the Landau-Teller formula
                params = input.reactions.item(i).nopress.lt;
```

```
                for n = istr:iend
                    Kri(n-istr+1, i) = Arr.A*chem.TT(n)^Arr.B*exp(-Arr.E...
                        /Rc/chem.TT(n)+params.Brev/chem.TT(n)^(1/3)+...
                        params.Crev/chem.TT(n)^(2/3));
                end
            else
                %calculate Kri according to the simple Arrhenius formula
                for n = istr:iend
                    Kri(n-istr+1, i) = Arr.A*chem.TT(n)^Arr.B*...
                        exp(-Arr.E/Rc/chem.TT(n));
                end
            end
        else
            %create array of all reaction species and their stoichiometric
            %coefficients
            ind = [[input.reactions.item(i).reactants.index] ...
                [input.reactions.item(i).products.index]];
            stoic = [-[input.reactions.item(i).reactants.stoich_coef] ...
                input.reactions.item(i).products.stoich_coef];
            stoic_sum = sum(stoic);
            for n = istr:iend
                %calculate DSi/R and DHi/RT
                DSi_R = 0;
                DHi_RT = 0;
                for j = 1:length(ind)
                    k = ind(j);
                    DSi_R = DSi_R+chem.SR(n,k)*stoic(j);
                    DHi_RT = DHi_RT+chem.HRT(n,k)*stoic(j);
                end
                Kpi = exp(DSi_R - DHi_RT);
                Kci = Kpi*(consts.Patm/consts.R/chem.TT(n))^stoic_sum;
                Kri(n-istr+1, i) = chem.Kf(n,i)/Kci;

            end
        end
    end
end
```

Πίνακας 30 – Κώδικας Getqri.m

```
%radiative heat loss, it is an array (N)
function [qri] = Getqri(consts, input, params, chem, singleYT, ii)
    if singleYT
        istr = ii;
        iend = ii;
    else
        istr = 1;
        iend = chem.N;
    end
    iwid = iend-istr+1;
    qri = zeros(iwid, 1);
    K2 = length(input.species.radiative);
    A = 2*298*pi*consts.h*consts.c^2/consts.sigma/consts.Patm;
    B = consts.h*consts.c/consts.kB;
    C = 4*chem.P*consts.sigma;
    D = params.Tinf^4;
    for n = istr:iend
        for kk = 1:K2
            k = input.species.radiative(kk);
            rad = input.species.item(k).radiation;
            kp = 0;
            for j = 1:length(rad.bands)
                kp = kp+A*rad.intensities(j)*rad.bcenters(j)^3/...
                    (exp(B*rad.bcenters(j)/chem.TT(n))-1)/chem.TT(n)^5;
            end
            qri(n-istr+1) = qri(n-istr+1)+chem.X(n,k)*kp;
        end
        qri(n-istr+1) = qri(n-istr+1)*C*(chem.TT(n)^4-D);
    end
end
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

Belytschko, T., Liu, W. K. & Moran, B., 2000. *Nonlinear Finite Elements for Continua and Structures.* West Sussex: John Wiley & Sons.

Bird, R. B., Stewart, W. E. & Lightfoot, E. N., 1960. *Transport Phenomena.* New York: John Wiley & Sons.

Chao, B. H. & Law, C. K., 1993. Asymptotic Theory of Flame Extinction with Surface Radiation. *Combustion and Flame,* 92, pp. 1-24.

Chao, B. H., Law, C. K. & Tien, J. S., 1993. Structure and extinction of diffusion flames with flame radiation. *Symposium (International) on Combustion*, 23(1), pp. 523-531.

Christiansen, E. W., Law, C. K. & Sung, C. J., 2001. Steady and Pulsating Propagation and Extinction of Rich Hydrogen/Air Flames at Elevated Pressures. *Combustion and Flame,* 124(1-2), pp. 35-49.

Christiansen, E. W., Tse, S. D. & Law, C. K., 2003. A computational study of oscillatory extinction of spherical diffusion flames. *Combustion and Flame,* 134(4), pp. 327–337.

Dhooge, A., Govaerts, W. & Kuznetsof, Y. A., 2003. MATCONT: A MATLAB package for numerical bifurcation analysis of ODEs. *Transactions on Mathematical Software,* 29(2), pp. 141-164.

Dietrich, D. L. et al, 1997. *NASA Conference Publication No. 10194.* Cleveland, OH, NASA.

El-Mahallawy, F. & E-Din Habik, S., 2002. *Fundamentals and Technology of Combustion.* Kidlington: Elsevier.

Frenklach, M., et al., 1995. *GRI-Mech.* [Ηλεκτρονικό] Available at: http://www.me.berkeley.edu/gri_mech [Πρόσβαση 23 September 2014].

Furi, M., Papas, P. & Monkewitz, P. A., 2000. Non-premixed Jet Flame Pulsations Near Extinction. *Proc. Combust.Inst.,* 28, pp. 831-838.

Kee, R. J., et al., 1998. *PREMIX: A FORTRAN Program for Modeling Steady Laminar One-Dimensional Premixed Flames.* San Diego, CA: Reaction Design.

Kee, R. J. et al, 2000. *CHEMKIN Collection.* San Diego, C: Reaction Design, Inc.

Kirkby, L. L. & Schmitz, R. A., 1966. An Analytical Study of the Stability of a Laminar Diffusion Flame. *Combustion and Flame,* 10(3), p. 205–220.

Kuznetzov, Y. A., 1998. *Elements of Applied Bifurcation Theory, Second Edition.* New York: Springer-Verlag.

Marrero, T. R. & Mason, E. A., 1972. Gaseous Diffusion Coefficients. *Journal of Physical and Chemical Reference Data*, 1, pp. 3-118.

McBride, B. J., Gordon, S. & Reno, M. A., 1993. Coefficients for Calculating Thermodynamic and Transport Properties of Individual Species. *NASA Report TM-4513,* October.

Mills, K. & Matalon, M., 1998. Extinction of Spherical Diffusion Flames in the Presence of Radiant Loss, *Proceedings of the Combustion Institute*, 27, pp. 2535–2541.

Monchick, L. & Mason, E. A., 1961. Transport properties of polar gases. *Journal of Chemical Physics,* 35, pp. 1676-1697.

Mueller, M. A., Kim, T. J., Yetter, R. A. & Dryer, F. L., 1999. Flow reactor studies and kinetic modeling of the H2/O2 reaction. *International Journal of Chemical Kinetics,* 31(2), pp. 113-125.

Nayagam, V. & Williams, F. A., 1998. *Seventh International Conference on Numerical Combustion*, York, England

Rogg, B., 1998. RUN-1DL: The Cambridge Universal Laminar Flamelet Computer Code.In: *Reduced Kinetic Mechanisms for Application in Combustion Systems.* Berlin: Springer-Verlag.

Siegel, R. & Howell, J. R., 1981. *Thermal Radiation Heat Transfer.* New York: Hemisphere Publishing Corporation.

Strogatz, S. H., 1994. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering.* Reading MA: Addison-Wesley Publishing Company.

Tien, C. L., 1967. Thermal Radiation Properties of Gases, in T.F.Irvine, Jr. and J.P. Hartnett (eds.), "Advances in Heat Transfer", vol 5, pp 253-324, Academin Press, New York, 1998.

Williams, F. A., 1985. *Combustion Theory: The Fundamental Theory of Chemically Reacting Flow Systems, Second Edition.* Menlo Park CA: The Benjamin/Cummings Publishing Company.

Wriggers, P., 2008. *Nonlinear Finite Element Methods.* Hannover: Springer.