

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

Μεταπτυχιακή Εργασία

**ΑΝΑΠΤΥΞΗ ΚΑΙ ΕΦΑΡΜΟΓΗ ΛΟΓΙΣΜΙΚΟΥ ΣΕ
ΠΕΡΙΒΑΛΛΟΝ ΜΑΤΛΑΒ ΠΡΟΣΟΜΟΙΩΣΗΣ ΗΛΙΑΚΩΝ
ΘΕΡΜΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΜΕ ΧΡΗΣΗ ΠΡΑΓΜΑΤΙΚΩΝ
ΚΛΙΜΑΤΙΚΩΝ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΕΦΑΡΜΟΓΕΣ ΘΕΡΜΑΝΣΗΣ
ΚΑΙ ΖΕΣΤΟΥ ΝΕΡΟΥ ΧΡΗΣΗΣ**

υπό

ΝΤΑΛΙΔΗ ΠΟΛΥΧΡΟΝΗ

Διπλωματούχου Μηχανικού Παραγωγής και Διοίκησης
Δ.Π.Θ. 2015

Υπεβλήθη για την εκπλήρωση μέρους των

απαιτήσεων για την απόκτηση του

Μεταπτυχιακού Διπλώματος Ειδίκευσης

2018

© 2018 Νταλίδης Πολυχρόνης

Η έγκριση της μεταπτυχιακής εργασίας από το Τμήμα Μηχανολόγων Μηχανικών της Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας δεν υποδηλώνει αποδοχή των απόψεων του συγγραφέα (Ν. 5343/32 αρ. 202 παρ. 2).

Εγκρίθηκε από τα Μέλη της Τριμελούς Εξεταστικής Επιτροπής:

Πρώτος Εξεταστής Δρ. Δημήτριος Βαλουγεώργης
(Επιβλέπων) Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών,
Πανεπιστήμιο Θεσσαλίας

Δεύτερος Εξεταστής Δρ. Κοζανίδης Γεώργιος
Επίκουρος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών,
Πανεπιστήμιο Θεσσαλίας

Τρίτος Εξεταστής Δρ. Σταπουντζής Ερρίκος
Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών,
Πανεπιστήμιο Θεσσαλίας

Ευχαριστίες

Στο πλαίσιο της παρούσας διπλωματικής εργασίας, οφείλω να ευχαριστήσω θερμά ορισμένους ανθρώπους για τη βοήθεια που μου προσέφεραν στην ολοκλήρωση της εκπόνησης της εργασίας και στην ολοκλήρωση των σπουδών μου για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης.

Πρώτα από όλους, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα της μεταπτυχιακής εργασίας μου, Δρ. Δημήτριο Βαλουγεώργη, Καθηγητή του Τμήματος Μηχανολόγων Μηχανικών του Πανεπιστημίου Θεσσαλίας. Η ανάθεση της παρούσας διπλωματικής με έκανε γνώστη του τομέα των ΗΘΣ, της προσομοίωσης θερμικών συστημάτων και διατάξεων αλλά και του προγραμματισμού. Η ενθάρρυνσή του, η υπομονή του, το ενδιαφέρον του, η συνεχής συνεργασία μαζί του και η μεταδοτικότητά του, παρά τις δυσκολίες που αντιμετώπισα λόγω της εργασίας μου και της μεγάλης απόστασης από το Τμήμα Μηχανολόγων Μηχανικών του Πανεπιστημίου Θεσσαλίας, αποτέλεσαν καταλυτικούς παράγοντες για την εκπόνηση της παρούσας εργασίας. Στη συνέχεια, θα ήθελα να ευχαριστήσω όλους τους διδάσκοντες του Τμήματος Μηχανολόγων Μηχανικών, της Πολυτεχνικής Σχολής του Π.Θ. και ιδιαίτερα την εξεταστική μου επιτροπή Δρ. Κοζανίδη Γεώργιο και Δρ. Σταπουντζή Ερρίκο για τις γνώσεις που μου μετέδωσαν και το υψηλό επίπεδο του μαθήματος που παρείχαν, με ιδιαίτερο ζήλο, κατά τη διάρκεια των διαλέξεων τους.

Επίσης, οφείλω να ευχαριστήσω θερμά τα στελέχη και τους συνεργάτες μου στην εταιρία «SABO S.A.», για τη συνεργασία τους και την κατανόηση που έδειξαν κατά τη διάρκεια των σπουδών μου για την απόκτηση του ΜΔΕ. Ακόμη, θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτωρ Αλέξανδρο Τσιμπούκη για την πολύτιμη βοήθεια που μου παρείχε κατά την εκπόνηση της μεταπτυχιακής μου εργασίας αλλά και την άμεση ανταπόκρισή του στα ζητήματα που αντιμετώπισα.

Τέλος, δεν θα μπορούσα να μην εκφράσω τις ευχαριστίες μου προς την οικογένειά μου η οποία με στήριξε και συνεχίζει να με στηρίζει με κάθε τρόπο, ηθικό και υλικό, όλα αυτά τα χρόνια.

Νταλίδης Πολυχρόνης

**ΑΝΑΠΤΥΞΗ ΚΑΙ ΕΦΑΡΜΟΓΗ ΛΟΓΙΣΜΙΚΟΥ ΣΕ ΠΕΡΙΒΑΛΛΟΝ MATLAB
ΠΡΟΣΟΜΟΙΩΣΗΣ ΗΛΙΑΚΩΝ ΘΕΡΜΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΜΕ ΧΡΗΣΗ
ΠΡΑΓΜΑΤΙΚΩΝ ΚΛΙΜΑΤΙΚΩΝ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΕΦΑΡΜΟΓΕΣ
ΘΕΡΜΑΝΣΗΣ ΚΑΙ ΖΕΣΤΟΥ ΝΕΡΟΥ ΧΡΗΣΗΣ**

ΝΤΑΛΙΔΗΣ ΠΟΛΥΧΡΟΝΗΣ

Πανεπιστήμιο Θεσσαλίας, Τμήμα Μηχανολόγων Μηχανικών, 2018

Επιβλέπων Καθηγητής: Δρ. Δημήτριος Βαλουγεώργης

Περίληψη

Τα ηλιοθερμικά συστήματα χρησιμοποιούνται ευρέως για την κάλυψη των αναγκών ζεστού νερού χρήσης και σε ορισμένες περιπτώσεις και για την κάλυψη αναγκών θέρμανσης σε ποικίλες διατάξεις. Σε αυτή τη μεταπτυχιακή εργασία μελετήθηκε το αντικείμενο των ηλιοθερμικών συστημάτων, αναλύθηκε η λειτουργία τους, ενώ ιδιαίτερη έμφαση δόθηκε στην προσομοίωση τους σε περιβάλλον Matlab.

Αρχικά πραγματοποιήθηκε περιγραφή της πηγής ενέργειας των ΗΘΣ, του ηλίου, και πώς οι σχετικές κινήσεις του με τη γη επηρεάζουν τη διαθέσιμη ενέργεια των συστημάτων. Ακολούθησε μαθηματική περιγραφή των ενεργειακών ισοζυγίων διάφορων διατάξεων που μελετήθηκαν και του τρόπου λειτουργίας τους. Στη συνέχεια αναπτύχθηκε πηγαίος κώδικας σε Matlab που περιλαμβάνει γραφικό περιβάλλον, ο οποίος προσομοίωσε τη λειτουργία των συστημάτων χρησιμοποιώντας τόσο πραγματικά δεδομένα από στοιχεία του ΤΕΕ, όσο και δεδομένα από τον χρήστη, χρησιμοποιώντας εξισώσεις αφενός του μαθήματος και αφετέρου αντίστοιχης βιβλιογραφίας. Τέλος, πραγματοποιήθηκε μία λεπτομερής και αναλυτική σύγκριση του αλγορίθμου προσομοίωσης στο Matlab με αποτελέσματα της βιβλιογραφίας, άλλων λογισμικών και διαφόρων πηγών από το διαδίκτυο ώστε να εξασφαλισθεί η ορθότητα του.

Τα συμπεράσματα τα οποία εξήχθησαν από τα αποτελέσματα ήταν ότι η ανάπτυξη λογισμικού προσομοίωσης ΗΘΣ αποτελεί μία πολύπλοκη διαδικασία με πολλούς εξωτερικούς παράγοντες οι οποίοι επηρεάζουν άμεσα και σημαντικά τις ποσότητες που τα διέπουν. Βασικότερο στοιχείο των διαφορών που παρουσιάστηκαν κατά τις συγκρίσεις αποτέλεσε η ακτινοβολία, ο τρόπος υπολογισμού της και η απόκλιση των διαφόρων μεθόδων από την πραγματικότητα, η οποία γίνεται όλο και πιο ασαφής μέσω της κλιματικής αλλαγής που χαρακτηρίζει την εποχή μας. Δευτερεύοντες αλλά εξίσου σημαντικοί παράγοντες αποτέλεσαν η κλίση και επιφάνεια του συλλέκτη, η ύπαρξη διαστρωμάτωσης ή εναλλάκτη, οι συνθήκες αντανάκλασης του περιβάλλοντος, ο μήνας και η τοποθεσία της προσομοίωσης.

Το λογισμικό που αναπτύχθηκε είναι εύκολα προσαρμοζόμενο στα χαρακτηριστικά ποικίλων εφαρμογών θέρμανσης και ζεστού νερού χρήσης και ιδιαίτερα φιλικό προς τον χρήστη. Δύναται να χρησιμοποιηθεί ευρέως για εκπαιδευτικούς σκοπούς, όπως επίσης και για την συστηματική μελέτη, σχεδιασμό και βελτιστοποίηση.

Πίνακας Περιεχομένων

Κεφάλαιο 1. Εισαγωγή	1
1.1 Γενικά.....	1
1.2 Αντικείμενο Μεταπτυχιακής Εργασίας	5
1.3 Δομή Μεταπτυχιακής Εργασίας	6
Κεφάλαιο 2. Σχεδιασμός και Προσομοίωση Ηλιοθερμικών Συστημάτων.....	9
2.1 Ηλιακή μηχανική και ΗΘΣ	9
2.1.1 Μεταβλητές που επηρεάζουν τη θέση του ηλίου	10
2.1.2 Γεωγραφικές συντεταγμένες.....	11
2.1.3 Ακτινοβολία λόγω διόρθωσης τροχιάς.....	12
2.1.4 Προσανατολισμός ΗΘΣ.....	13
2.1.5 Ηλιακή ακτινοβολία.....	17
2.1.6 Ωρολογιακός και ηλιακός χρόνος	21
2.1.7 Υπολογισμός θερμοκρασίας νερού και εδάφους	23
2.1.8 Βαθμοήμερες θέρμανσης και υπολογισμός φορτίων θέρμανσης ...	24
2.1.9 Στοιχεία από TEE	25
2.2 Λειτουργία των Ηλιακών Θερμικών Συστημάτων	25
2.2.1 Τυπική διάταξη ΗΘΣ θέρμανσης.....	25
2.2.2 Ανάλυση ηλιακού συλλέκτη	26
2.2.3 Μοντέλο λειτουργίας ηλιακού θερμικού συστήματος για θέρμανση	29
2.2.4 Εναλλάκτης και κύκλωμα Ζεστού Νερού Χρήσης.....	31

2.2.5 Δεξαμενή αποθήκευσης	34
2.2.6 Δυναμικό και στατικό φορτίο κατανάλωσης	38
Κεφάλαιο 3. Ανάλυση λειτουργίας αλγορίθμου	39
3.1 Λογισμικό Matlab	39
3.2 Δομή του αλγορίθμου	39
3.3 Μεταβλητές του αλγορίθμου	40
3.4 Γραφικό περιβάλλον και λειτουργίες.....	46
3.5 Ορισμός αλγοριθμικού χρόνου λειτουργίας	49
3.7 Τροφοδοσία και ενημέρωση πραγματικών δεδομένων	52
3.8 Διάγραμμα ροής και βηματική εκτέλεση αλγορίθμου.....	55
3.9 Παρουσίαση πηγαιού κώδικα	63
3.9.1 Διακόπτες.....	64
3.9.2 Χρονικό βήμα εκτέλεσης.....	64
3.9.3 Ενημέρωση με πραγματικά δεδομένα.....	65
3.9.4 Ηλιακή και πραγματική ώρα.....	66
3.9.5 Υπολογισμός ακτινοβολίας και ηλιακών γωνιών	67
3.9.6 Υπολογισμοί διαστάσεων δεξαμενής.....	70
3.9.7 Στοιχεία για τους θερμικούς υπολογισμούς.....	71
3.9.8 Βαθμομημέρες.....	72
3.9.9 Θερμοκρασία νερού δικτύου ύδρευσης	72
3.9.10 Ζεστό νερό χρήσης	73

3.9.11 Εναλλάκτης.....	74
3.9.12 Δυναμικό φορτίο.....	74
3.9.13 Υπολογισμοί με ή χωρίς διαστρωμάτωση δοχείου αποθήκευσης	75
3.9.14 Αυτοματισμοί.....	77
3.9.15 Αποτελέσματα Excel και γραφικού περιβάλλοντος	78
Κεφάλαιο 4. Αποτελέσματα και πιστοποίηση πηγαίου κώδικα	81
4.1 Πιστοποίηση κώδικα θέρμανσης και στατικού φορτίου	81
4.2 Σύγκριση μεθόδων υπολογισμού ακτινοβολιών.....	94
4.3 Πιστοποίηση κώδικα ZNX και Δυναμικού Φορτίου.....	102
4.4 Παράδειγμα λειτουργίας κώδικα στην ολοκληρωμένη του μορφή	104
Κεφάλαιο 5 Συμπεράσματα	111
Αναφορές	114
Παραρτήματα.....	116
Α. Πηγαίος κώδικας ΗΘΣ για θέρμανση χώρων και ZNX.....	116
Β. Πηγαίος κώδικας γραφικού περιβάλλοντος ΗΘΣ.....	133
Γ. Πίνακες του ΤΕΕ	154

Κατάλογος Πινάκων

Πίνακας 2.1 Τυπικοί συντελεστές αντανάκλασης ανάλογα με το περιβάλλον.	18
Πίνακας 2.2 Συντελεστές θερμικής διάχυσης του εδάφους.....	24
Πίνακας 3.1 Μεταβλητές ελέγχου διακοπών.	40
Πίνακας 3.2 Μεταβλητές του χρόνου και τόπου της προσομοίωσης.	41
Πίνακας 3.3 Μεταβλητές κτιρίου, δεξαμενής αποθήκευσης, δικτύου ύδρευσης.	41
Πίνακας 3.4 Μεταβλητές συλλέκτη και εναλλάκτη.	42
Πίνακας 3.5 Μεταβλητές θερμικών φορτίων, ακτινοβολίας και παροχών.....	42
Πίνακας 3.6 Μεταβλητές γωνιών ηλίου – συλλέκτη.	43
Πίνακας 3.7 Μεταβλητές ηλιακού και πραγματικού χρόνου.	43
Πίνακας 3.8 Μεταβλητές θερμοκρασιών του συστήματος.....	43
Πίνακας 3.9 Μεταβλητές λειτουργίας διαστρωμάτωσης.	44
Πίνακας 3.10 Μεταβλητές διαχείρισης και εκτύπωσης δεδομένων.	44
Πίνακας 4.1 Ώρες ανατολής και δύσης του ηλίου για την περιοχή της Ελλάδας κατά τη διάρκεια του έτους.....	95
Πίνακας Γ.1 Τοπολογικά δεδομένα για διάφορες τοποθεσίες της Ελλάδας..	154
Πίνακας Γ.2 Μέση μηνιαία θερμοκρασία εικοσσιτετραώρου για διάφορες τοποθεσίες της Ελλάδας.....	155
Πίνακας Γ.3 Μέση μηνιαία ηλιακή ακτινοβολία στο οριζόντιο επίπεδο για διάφορες τοποθεσίες της Ελλάδας.	156

Πίνακας Γ.4 Μέσος μηνιαίος συντελεστής αιθριότητας για διάφορες τοποθεσίες της Ελλάδας.....	157
Πίνακας Γ.5 Μέση θερμοκρασία δικτύου παροχής νερού για διάφορες τοποθεσίες της Ελλάδας.....	158
Πίνακας Γ.6 Μέγιστη μηνιαία θερμοκρασία για διάφορες τοποθεσίες της Ελλάδας.....	159
Πίνακας Γ.7 Μέγιστη μηνιαία διακύμανση θερμοκρασίας για διάφορες τοποθεσίες της Ελλάδας.....	160
Πίνακας Γ.8 Βαθμοημέρες θέρμανσης με θερμοκρασία αναφοράς 18 βαθμούς κελσίου για διάφορες τοποθεσίες της Ελλάδας.	161

Κατάλογος Σχημάτων

Σχήμα 4.1 Θερμοκρασία εισόδου του φέροντος ρευστού στο συλλέκτη για περίοδο δύο ημερών για επιφάνειες 20,30,40 m ²	83
Σχήμα 4.2 Θερμοκρασία εξόδου του φέροντος ρευστού από το συλλεκτη για περίοδο δύο ημερών για επιφάνειες 20,30,40 m ²	84
Σχήμα 4.3 Θερμοκρασία αποθήκευσης του ρευστού για περίοδο δύο ημερών για επιφάνειες 20,30,40 m ²	84
Σχήμα 4.4 Θερμοκρασία εισόδου του φέροντος ρευστού στην κατανάλωση για περίοδο δύο ημερών για επιφάνειες 20,30,40 m ²	85
Σχήμα 4.5 Θερμοκρασία εξόδου του φέροντος ρευστού από την κατανάλωση για περίοδο δύο ημερών για επιφάνειες 20,30,40 m ²	85
Σχήμα 4.6 Αποθηκευμένη ενέργεια στη δεξαμενή αποθήκευσης για περίοδο δύο ημερών για επιφάνεια 20,30,40 m ²	86
Σχήμα 4.7 Βαθμός απόδοσης του συστήματος για περίοδο δύο ημερών για επιφάνειες 20,30,40 m ²	86
Σχήμα 4.8 Θερμοκρασία αποθήκευσης πιστοποιημένου αλγορίθμου Fortran ως προς το χρόνο για 3 ημέρες με θερμικό φορτίο 1000,2000,4000,8000 W.	87
Σχήμα 4.9 Θερμοκρασία αποθήκευσης του αλγορίθμου Matlab ως προς το χρόνο για 4 ημέρες με θερμικό φορτίο 1000,2000,4000,8000 W.	87
Σχήμα 4.10 Θερμοκρασία αποθήκευσης αλγορίθμου Fotran ως προς το χρόνο για 3 ημέρες με $A=15,30,45,60$ m ²	88
Σχήμα 4.11 Θερμοκρασία αποθήκευσης του αλγορίθμου Matlab ως προς το χρόνο για 4 ημέρες με $A=15,30,45,60$ m ²	88

Σχήμα 4.12 Σύγκριση ακτινοβολιών μεταξύ των δύο αλγορίθμων για 2 ημέρες του Ιουλίου.....	89
Σχήμα 4.13 Θερμοκρασία αποθήκευσης ως προς το χρόνο για 4 ημέρες με σταθερό θερμικό φορτίο 1000 W και φορτίο 887 W για τις βαθμοημέρες	91
Σχήμα 4.14 Θερμοκρασία αποθήκευσης ως προς το χρόνο για 4 ημέρες με ή χωρίς διαστρωμάτωση.	92
Σχήμα 4.15 Βαθμός απόδοσης για τα δύο είδη δεξαμενών για προσομοίωση 4 ημερών.	92
Σχήμα 4.16 Σύγκριση θερμοκρασιών νερού παροχής δικτύου.....	93
Σχήμα 4.17 Ωριαία ολική ηλιακή ακτινοβολία σε κεκλιμένη επιφάνεια για τον Ιανουάριο και Ιούλιο σύμφωνα με τους αλγορίθμους Matlab και Fortran.....	94
Σχήμα 4.18 Ακτινοβολία κατά τη διάρκεια της ημέρας για τα δεδομένα του παραδείγματος σύμφωνα με τον πρώτο και δεύτερο τρόπο υπολογισμού της ακτινοβολίας του αλγορίθμου.....	96
Σχήμα 4.19 Ακτινοβολία στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιανουαρίου για γωνίες $\beta=\varphi$, $\beta= \varphi+15$, $\beta=\varphi-15$	97
Σχήμα 4.20 Ακτινοβολία στην περιοχή της Αλεξανδρούπολης για την πρώτη ημέρα του Ιανουαρίου για γωνίες $\beta=\varphi$, $\beta= \varphi+15$, $\beta=\varphi-15$	97
Σχήμα 4.21 Ακτινοβολία στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιουλίου για γωνίες $\beta=\varphi$, $\beta= \varphi+15$, $\beta=\varphi-15$	98
Σχήμα 4.22 Ακτινοβολία στην περιοχή της Αλεξανδρούπολης για την πρώτη ημέρα του Ιουλίου για γωνίες $\beta=\varphi$, $\beta= \varphi+15$, $\beta=\varphi-15$	98

Σχήμα 4.23 Ακτινοβολία αλγορίθμου Matlab στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιανουαρίου για γωνία $\beta=\varphi$ σε σχέση με Trnsys και JRC.....	99
Σχήμα 4.24 Ακτινοβολία αλγορίθμου Matlab στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιανουαρίου για γωνία $\beta=\varphi-15$ σε σχέση με Trnsys και JRC.	99
Σχήμα 4.25 Ακτινοβολία αλγορίθμου Matlab στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιανουαρίου για γωνία $\beta= \varphi+15$ σε σχέση με Trnsys και JRC.	100
Σχήμα 4.26 Ακτινοβολία αλγορίθμου Matlab στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιουλίου για γωνία $\beta=\varphi$ σε σχέση με Trnsys και JRC.....	100
Σχήμα 4.27 Ακτινοβολία αλγορίθμου Matlab στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιουλίου για γωνία $\beta=\varphi-15$ σε σχέση με Trnsys και JRC.	100
Σχήμα 4.28 Ακτινοβολία αλγορίθμου Matlab στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιουλίου για γωνία $\beta= \varphi+15$ σε σχέση με Trnsys και JRC.	101
Σχήμα 4.29 Θερμοκρασία εξόδου του φέροντος ρευστού από το συλλέκτη προς τη δεξαμενή αποθήκευσης για μία ημέρα.....	103
Σχήμα 4.30 Θερμοκρασία αποθήκευσης του φέροντος ρευστού για μία ημέρα.	103
Σχήμα 4.31 Θερμοκρασία εισόδου του φέροντος ρευστού στην κατανάλωση από τη δεξαμενή αποθήκευσης για μία ημέρα.....	104
Σχήμα 4.32 Θερμοκρασία εισόδου και εξόδου του φέροντος ρευστού από το συλλέκτη προς τη δεξαμενή αποθήκευσης για την τελευταία ημέρα του Οκτωβρίου και την πρώτη ημέρα του Νοεμβρίου.	105
Σχήμα 4.33 Θερμοκρασία αποθήκευσης του φέροντος ρευστού για την τελευταία ημέρα του Οκτωβρίου και την πρώτη ημέρα του Νοεμβρίου.	105

Σχήμα 4.34 Θερμοκρασία εισόδου και εξόδου του φέροντος ρευστού από την κατανάλωση προς τη δεξαμενή αποθήκευσης για την τελευταία ημέρα του Οκτωβρίου και την πρώτη ημέρα του Νοεμβρίου.	106
Σχήμα 4.35 Αποθηκευμένη ενέργεια στη δεξαμενή αποθήκευσης για την τελευταία ημέρα του Οκτωβρίου και την πρώτη ημέρα του Νοεμβρίου.	106
Σχήμα 4.36 Βαθμός απόδοσης του συστήματος για περίοδο δύο ημερών για την τελευταία ημέρα του Οκτωβρίου και την πρώτη ημέρα του Νοεμβρίου.	107
Σχήμα 4.37 Θερμοκρασία περιβάλλοντος για την τελευταία ημέρα του Οκτωβρίου και την πρώτη ημέρα του Νοεμβρίου.	107
Σχήμα 4.38 Ακτινοβολία για την τελευταία ημέρα του Οκτωβρίου και την πρώτη ημέρα του Νοεμβρίου.	108

Κατάλογος Εικόνων

Εικόνα 1.1 Η πορεία της αγοράς ΗΘΣ από το 2000 ως το 2011.....	2
Εικόνα 1.2 Εγκατεστημένη ισχύς ΗΘΣ στην Ευρώπη	2
Εικόνα 2.1 Θεωρητική και πραγματική ακτινοβολία.	10
Εικόνα 2.2 Απεικόνιση γεωγραφικού μήκους και πλάτους γής.	12
Εικόνα 2.3 Τροχία γής σε σχέση με ήλιο.....	13
Εικόνα 2.4 Γωνία απόκλισης ανά μήνα του έτους για το βόρειο ημισφαίριο	13
Εικόνα 2.5 Ηλιακές γωνίες.	15
Εικόνα 2.6 Γωνίες συλλέκτη.....	16
Εικόνα 2.7 Τροχιά του ηλίου για τυπικό γεωγραφικό πλάτος 38°N.	17
Εικόνα 2.8 Ισοζύγιο ακτινοβολίας από και προς τη γή.	18
Εικόνα 2.9 Ηλιακό μεσημέρι στο βόριο ημισφαίριο για μία τοποθεσία.	22
Εικόνα 2.10 Τυπική διάταξη ενός ΗΘΣ.....	26
Εικόνα 2.11 Τυπικά μέρη ενός ηλιακού συλλέκτη.....	26
Εικόνα 2.12 Τυπική διάταξη και διάγραμμα ροής ΗΘΣ για θέρμανση.....	29
Εικόνα 2.13 Τυπική διάταξη και διάγραμμα ροής ΗΘΣ για ΖΝΧ.	31
Εικόνα 2.14 Δεξαμενή αποθήκευσης με διαστρωμάτωση.....	35
Εικόνα 3.1 Διάγραμμα ροής γενικής λειτουργίας του αλγορίθμου.	40
Εικόνα 3.2 Απεικόνιση του γραφικού παραθύρου διεπαφής χρήστη.	46
Εικόνα 3.3 Διάγραμμα χρονικού βήματος λειτουργίας του αλγορίθμου.....	50

Εικόνα 3.4 Διάγραμμα αλληλεπίδρασης Matlab, Excel και γραφικού περιβάλλοντος.....	51
Εικόνα 3.5 Απεικόνιση των πινάκων δεδομένων εισαγωγής	52
Εικόνα 3.6 Διάγραμμα αλληλεπίδρασης μεταβλητών προσπέλασης δεδομένων.	53
Εικόνα 3.7 Διάγραμμα ροής προσπέλασης δεδομένων	54
Εικόνα 3.8 Απεικόνιση των πινάκων δεδομένων εισαγωγής για γεωγραφικό μήκος και πλάτος.	55
Εικόνα 3.9 Διάγραμμα αλληλεπίδρασης μεταβλητών προσπέλασης δεδομένων γεωγραφικού μήκους και πλάτους.	55
Εικόνα 3.10 Διάγραμμα ροής ρύθμισης και αρχικοποίησης παραμέτρων του αλγορίθμου.....	58
Εικόνα 3.11 Διάγραμμα ροής υπολογισμού ηλιακών παραμέτρων.....	59
Εικόνα 3.12 Διάγραμμα ροής λειτουργίας εναλλάκτη.	59
Εικόνα 3.13 Διάγραμμα ροής αυτοματισμών.	61
Εικόνα 3.14 Διάγραμμα ροής υπολογισμών δεξαμενής αποθήκευσης.	62
Εικόνα 3.15 Διάγραμμα ροής εκτύπωσης αποτελεσμάτων.	63

Κεφάλαιο 1. Εισαγωγή

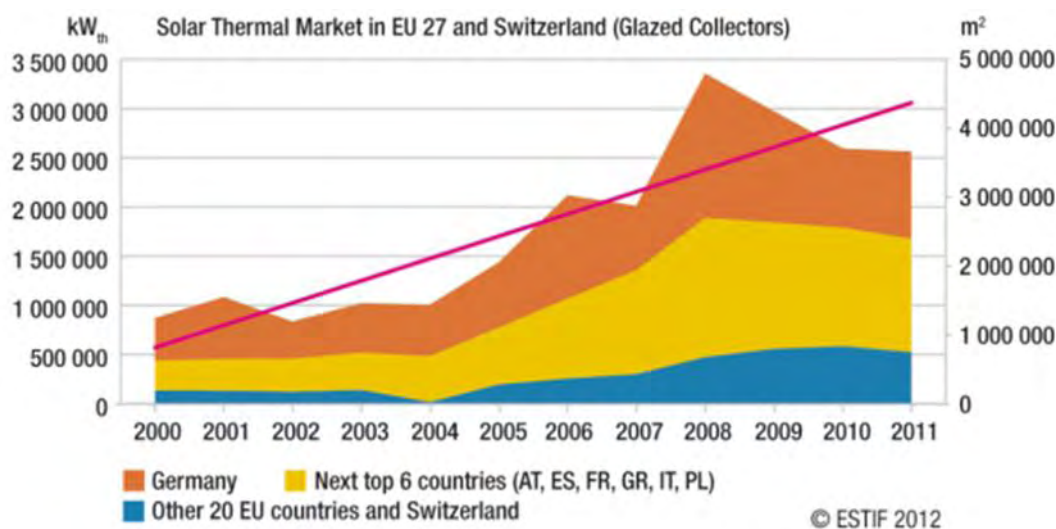
1.1 Γενικά

Είναι γεγονός πως τα τελευταία χρόνια παρατηρείται έντονη αύξηση της ενεργειακής κατανάλωσης στις σύγχρονες κοινωνίες. Χαρακτηριστικό είναι το γεγονός πως η συνολική ενεργειακή κατανάλωση ξεπερνάει τους 12000 Mton, ενώ παράλληλα τα ορυκτά καύσιμα (και κυρίως το πετρέλαιο) αποτελούν τις δημοφιλέστερες πηγές ενέργειας. Ιδιαίτερο ενδιαφέρον παρουσιάζει το γεγονός πως από την συνολική ενεργειακή κατανάλωση το 47% προέρχεται από την κάλυψη των θερμικών αναγκών ενώ η κατανάλωση για ηλεκτρισμό και μεταφορές είναι 17% και 27% αντίστοιχα [1]. Επίσης, ο οικιακός τομέας είναι ένας από τους περισσότερο ενεργοβόρους τομείς, καθώς του αντιστοιχεί το 41% της συνολικής κατανάλωσης ενέργειας θέρμανσης [2]. Αποτέλεσμα των παραπάνω είναι η υποβάθμιση και μόλυνση του περιβάλλοντος λόγω της εκπομπής ρύπων, όπως του CO₂, που αποτελεί τον βασικότερο ρύπο για το φαινόμενο του θερμοκηπίου. Με στόχο τον περιορισμό και την καταπολέμηση της κλιματικής αλλαγής είναι φανερό πως απαιτείται η υιοθέτηση και αξιοποίηση εναλλακτικών μορφών ενέργειας όπως οι Ανανεώσιμες Πηγές Ενέργειας (ΑΠΕ). Συγκεκριμένα η Ευρωπαϊκή Ένωση (ΕΕ) έχει θέσει τρεις στόχους για το 2020. Επιδιώκεται μείωση των εκπομπών αερίων του θερμοκηπίου κατά 20% σε σχέση με το 1990, το 20% της κατανάλωσης ενέργειας να προέρχεται από ΑΠΕ και 20% εξοικονόμηση ενέργειας σε σχέση με τις προβλέψεις [3].

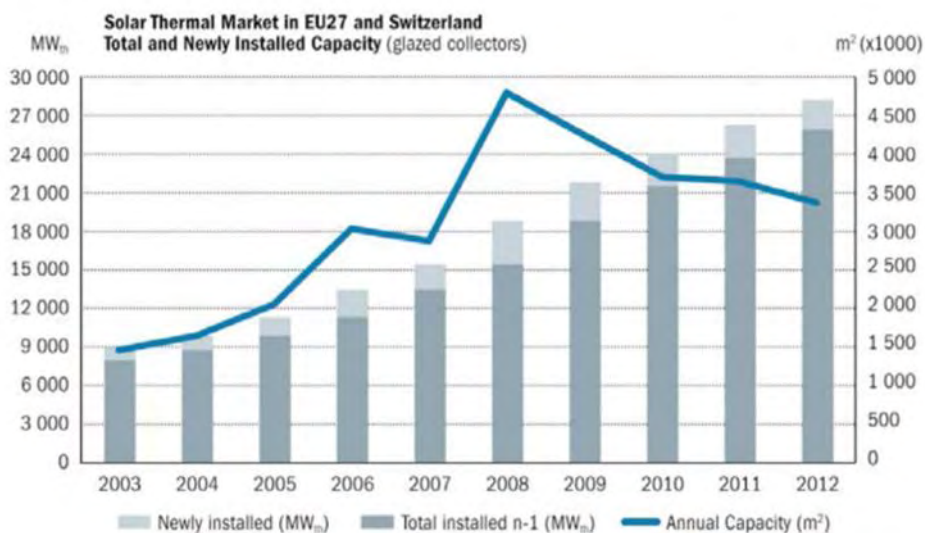
Ιδιαίτερο ενδιαφέρον παρουσιάζουν τα ηλιακά συστήματα τα οποία μπορούν να χρησιμοποιηθούν για την κάλυψη αναγκών Ζεστού Νερού Χρήσης (ZNX), θέρμανσης και ψύξης. Η αξιοποίηση του ηλιακού δυναμικού έχει αποτελέσει αντικείμενο εκτεταμένης έρευνας τα τελευταία χρόνια. Το μεγαλύτερο πλεονέκτημα της ηλιακής ενέργειας σε σχέση με άλλες μορφές έγκειται στο γεγονός πως μπορεί να παρέχεται χωρίς να απαιτείται κάποια προ επεξεργασία, χωρίς να δημιουργείται απειλή για το περιβάλλον και είναι ανεξάντλητη [4]. Τα βασικά στοιχεία ενός ηλιακού συστήματος είναι ο ηλιακός συλλέκτης και η δεξαμενή αποθήκευσης ενώ το εργαζόμενο μέσο συνήθως είναι νερό ή αέρας.

Ηλιακά συστήματα τα οποία χρησιμοποιούνται για παραγωγή ZNX είναι ήδη διαδεδομένα σε πολλές περιοχές ανά τον κόσμο. Ιδιαίτερα σε περιοχές όπως η Νότια Ευρώπη, οι οποίες χαρακτηρίζονται από μεσογειακό κλίμα, υπάρχουν θερμοσιφωνικά

συστήματα στην πλειονότητα των νοικοκυριών. Ωστόσο, αυτές οι εγκαταστάσεις είναι μικρής κλίμακας καθώς στοχεύουν στην κάλυψη αναγκών λίγων ατόμων ανά οικογένεια. Στην Κεντρική και Βόρεια Ευρώπη λόγω έλλειψης έντονης ηλιοφάνειας χρησιμοποιούνται συστήματα εξαναγκασμένης κυκλοφορίας με σκοπό την κάλυψη αναγκών σε ΖΝΧ μεγαλύτερου αριθμού ατόμων (π.χ. τετραμελών οικογενειών). Τα συστήματα αυτά παρουσιάζουν μεγάλο ποσοστό κάλυψης (60%) και οι ηλιακοί συλλέκτες ανέρχονται στα 4-6 m² [5].



Εικόνα 1.1 Η πορεία της αγοράς ΗΘΣ από το 2000 ως το 2011



Εικόνα 1.2 Εγκατεστημένη ισχύς ΗΘΣ στην Ευρώπη

Για την ανάπτυξη και προώθηση των ΗΘΣ είναι απαραίτητο να βρεθούν τρόποι μείωσης του κόστους των συστημάτων, βελτίωσης των τεχνικών χαρακτηριστικών

τους και απλοποίησης της διαδικασίας εγκατάστασης, ώστε να γίνουν δελεαστικά προς τους καταναλωτές και τις βιομηχανίες [5]. Αυτό αποτελεί στόχο του προγράμματος SHIP roadmap όπου γίνεται μία προσπάθεια να βρεθούν τυποποιημένες και οικονομικά βέλτιστες λύσεις για όλες τις βιομηχανικές διεργασίες. Η δημιουργία νέων μοντέλων συλλεκτών είναι απαραίτητη ώστε να είναι δυνατή η χρήση τους σε βιομηχανικές εγκαταστάσεις καθώς και η ανάπτυξη καινοτόμων εργαλείων σχεδιασμού [6].

Η μείωση του κόστους στα ΗΘΣ μπορεί να επιτευχθεί επίσης με την αύξηση του βαθμού απόδοσης των συστημάτων, καθώς κάτι τέτοιο θα οδηγήσει και σε μικρότερους χρόνους απόσβεσης του αρχικού κεφαλαίου, με μείωση του κόστους των επιμέρους εξαρτημάτων καθώς και με την απλοποίηση της διαδικασίας εγκατάστασης του συστήματος και του συλλέκτη. Επιδιώκεται αύξηση της εκμετάλλευσης της ηλιακής ενέργειας για ΖΝΧ και θέρμανση των χώρων (από 25 σε 60%) χωρίς όμως την αύξηση του κόστους θέρμανσης. Το παραπάνω μπορεί να επιτευχθεί μέσω του σωστού σχεδιασμού του συστήματος. Τα συστήματα ελέγχου, οι ηλιακοί συλλέκτες και η δεξαμενή αποθήκευσης αποτελούν τμήματα μείζονος σημασίας για την βελτίωση και τεχνολογική εξέλιξη των ΗΘΣ.

Ένα καλό σύστημα ελέγχου μπορεί να οδηγήσει σε σημαντική αύξηση του βαθμού απόδοσης ενός ηλιοθερμικού συστήματος. Έχουν πραγματοποιηθεί πολλές έρευνες για τη βελτιστοποίηση του ελέγχου στα μοντέλα των ΗΘΣ. Βελτιώσεις μπορούν να επιτευχθούν με ρύθμιση της παροχής της μάζας του εργαζόμενου μέσου καθώς και με ρύθμιση των ηλιακών συλλεκτών ώστε να αξιοποιούν την μέγιστη δυνατή προσπίπτουσα ηλιακή ακτινοβολία [7]. Το 2014 στην Κίνα πραγματοποιήθηκε έρευνα από τους Z.S.Lu και Wang για το πώς επηρεάζεται η απόδοση των ηλιοθερμικών συστημάτων από τον τύπο του συλλέκτη. Από τα αποτελέσματα της έρευνας διαπιστώνεται πως οι συλλέκτες οι οποίοι προσφέρουν τα καλύτερα αποτελέσματα είναι οι συλλέκτες κενού [8]. Πολλές μελέτες έχουν γίνει και έχουν αναπτυχθεί μοντέλα για την περιγραφή της κατανομής της θερμοκρασίας μέσα στη δεξαμενή η οποία επηρεάζει σε σημαντικό βαθμό την συμπεριφορά και αυξάνει την απόδοση του συστήματος [9] [10].

Όλα τα παραπάνω έχουν δημιουργήσει την ανάγκη για την εκπόνηση μελετών με μεγάλη ακρίβεια και λεπτομέρεια, η οποία επιτυγχάνεται μέσα από λογισμικά προσομοίωσης ηλιακών θερμικών συστημάτων. Τα λογισμικά αυτά έχουν ως στόχο την πρόβλεψη και ακριβή μελέτη του συστήματος, με στόχο τη βελτιστοποίηση των

παραμέτρων που το διέπουν. Τα πιο διαδεδομένα λογισμικά μελετών είναι το Trnsys, το Polysun και το 4M.

Το Trnsys (TRaNsient SYstems Simulation program), το οποίο είναι ανοικτό λογισμικό και αναπτύχθηκε από το Solar lab του πανεπιστημίου του Wisconsin αποτελεί ένα πλήρες και ανοιχτό προς το κοινό υπολογιστικό περιβάλλον για την προσομοίωση χρονικά μεταβαλλόμενων συστημάτων, συμπεριλαμβανομένων και πολυζωνικών κτιρίων [11]. Η σουίτα 4M της ομώνυμης εταιρίας αποτελεί ένα εμπορικό υπολογιστικό εργαλείο για τη δημιουργία H/M μελετών εξαιρετικά διαδεδομένο στην Ελλάδα το οποίο εμπεριέχει μεταξύ άλλων πακέτο ηλιακών υπολογισμών [12]. Το Polysun της εταιρίας Velasolaris είναι ένα εμπορικό λογισμικό το οποίο έχει δημιουργηθεί αποκλειστικά για την προσομοίωση ποικίλων διατάξεων ΗΘΣ [13].

Η χρήση τους γίνεται για την αξιολόγηση νέων ενεργειακών διατάξεων, από απλά συστήματα θέρμανσης νερού οικιακής χρήσης έως το σχεδιασμό και τη προσομοίωση κτιρίων και του εξοπλισμού τους, συμπεριλαμβανομένων των συστημάτων ελέγχου. Η σουίτα της 4M χρησιμοποιεί ως βάση την πιστοποιημένη μέθοδο των καμπυλών f , η οποία αναπτύχθηκε από τους Duffie και Beckman το 1976 [14]. Οι καμπύλες f στηρίζονται στη λογική ότι για συστήματα θέρμανσης και ζεστού νερού χρήσης το κλάσμα του ολικού θερμικού φορτίου εξαρτάται από δύο παραμέτρους, τις X και Y , οι οποίες διορθώνονται με συντελεστές ανάλογα με τη χρήση του ΗΘΣ.

Η πρωτοτυπία και διαφορά του αλγορίθμου που θα παρουσιαστεί στα επόμενα κεφάλαια είναι ότι δεν βασίζεται σε εμπειρικές μεθόδους και τύπους με διορθωτικούς συντελεστές, αλλά χρησιμοποιεί τους αναλυτικούς τύπους που διέπουν τα ΗΘΣ. Οι υπολογισμοί γίνονται δυναμικά με πραγματικό χρονικό βήμα για κάθε δευτερόλεπτο της εκτέλεσής της σύμφωνα με τη διάταξη που έχει ορίσει ο χρήστης, όπως συμβαίνει και με τα λογισμικά Trnsys και Polysun. Επίσης, τα αποτελέσματα δεν εκφράζονται από δύο μόνο μεταβλητές, αλλά από τα αναλυτικά δεδομένα εισαγωγής του προγράμματος.

Ειδικότερα, σε αντίθεση με το μοντέλο που περιγράφεται, οι καμπύλες f δεν μπορούν να χρησιμοποιηθούν για εφαρμογές εκτός από αυτές που σχεδιάστηκαν, όπως για παράδειγμα αυτή της ηλιακής ψύξης, ή κατά την περίπτωση όπου στο σύστημα δεν

υπάρχει δεξαμενή αποθήκευσης. Επιπρόσθετα δεν μπορούν να χρησιμοποιηθούν για τον υπολογισμό του βαθμού απόδοσης του συστήματος και δεν συνυπολογίζουν απώλειες όπως αυτές από αντλίες, ελεγκτές, ανεμιστήρες και άλλες μορφές παρασιτικής ενέργειας, η οποία εκτιμάται σε ένα ποσοστό της τάξεως του 5% [15]. Σύμφωνα με μελέτες των Fanney και Klein [16] οι καμπύλες f παρουσιάζουν υπερεκτίμηση των υπολογισμών σε σύγκριση με πραγματικές δοκιμές από 1% έως 5%. Τέλος, με τη μέθοδο αυτή δεν είναι δυνατό να λάβει κανείς υπόψη περιβαλλοντολογικούς παράγοντες όπως ομίχλη ή υψόμετρο [17].

Όπως είναι λογικό, ο αλγόριθμος που θα παρουσιαστεί παρακάτω δεν θα καταφέρει να έχει την ακρίβεια και ευελιξία των Polysun και Trnsys τα οποία αναπτύσσονται εδώ και δεκαετίες, αλλά θα είναι πιο ευέλικτος από μεγάλο πλήθος εφαρμογών των καμπυλών f .

1.2 Αντικείμενο Μεταπτυχιακής Εργασίας

Στη συγκεκριμένη μεταπτυχιακή εργασία αναπτύσσεται μία προσομοίωση των ηλιακών θερμικών συστημάτων για την παραγωγή ζεστού νερού χρήσης και θέρμανσης χώρων σε περιβάλλον Matlab. Στα πλαίσια της εργασίας χρησιμοποιήθηκε ως βάση το πηγαίο λογισμικό μελέτης ΗΘΣ σε περιβάλλον Fortran του Εργαστηρίου Φυσικών και Χημικών Διεργασιών του Πανεπιστημίου Θεσσαλίας.

Το λογισμικό αυτό μεταγλωττίστηκε από Fortran σε Matlab ενώ έγινε μία προσπάθεια σύμπτυξης των διάφορων αποσπασμάτων. Τα αποσπάσματα που συμπύχθηκαν σε ένα ενιαίο αλγόριθμο αφορούσαν τα ηλιακά θερμικά συστήματα θέρμανσης και ζεστού νερού χρήσης με έμφαση στη διαστρωμάτωση, τον εναλλάκτη, και το δυναμικά χρονομεταβαλλόμενο φορτίο.

Στη συνέχεια ο αλγόριθμος εμπλουτίστηκε με πραγματικά δεδομένα από το TEE ώστε να επιτευχθεί ένα πιο ρεαλιστικό αποτέλεσμα. Έπειτα, μέσω των δεδομένων αυτών έγιναν προσπάθειες βελτίωσης του αλγορίθμου υλοποιώντας κώδικα για τον υπολογισμό της ηλιακής ακτινοβολίας μέσω εξισώσεων προσομοίωσης. Προτάθηκε το πολύ μικρό βήμα εκτέλεσης του ενός δευτερολέπτου για μεγαλύτερη ακρίβεια, προστέθηκε η λειτουργία των βαθμομερών και μία μέθοδος υπολογισμού της θερμοκρασίας του νερού από την παροχή του δικτύου. Τέλος αναπτύχθηκε ένα γραφικό παράθυρο διεπαφής χρήστη για την εύκολη διαχείριση του κώδικα.

Στα επόμενα κεφάλαια θα παρουσιασθεί η μοντελοποίηση για την ανάπτυξη του λογισμικού. Ιδιαίτερη έμφαση δόθηκε στην ακρίβεια του αλγορίθμου, την επεξεργασία πραγματικών δεδομένων και το συνδυασμό τους με τις εξισώσεις που διέπουν τα ΗΘΣ. Βασικό μέλημα της εργασίας αποτέλεσαν το φιλικό προς το χρήστη γραφικό περιβάλλον και η λειτουργικότητα του αλγορίθμου για τις διάφορες πόλεις της Ελλάδας.

1.3 Δομή Μεταπτυχιακής Εργασίας

Στο πρώτο κεφάλαιο δίδεται μια εισαγωγή στα ΗΘΣ και παρουσιάζονται το αντικείμενο και η δομή της μεταπτυχιακής εργασίας.

Στο δεύτερο κεφάλαιο πραγματοποιείται βιβλιογραφική ανασκόπηση για τις σχετικές κινήσεις μεταξύ ηλίου και γης και των εξισώσεων που τις περιγράφουν πάντα με γνώμονα την ηλιακή ακτινοβολία. Στη συνέχεια αναλύονται τα ηλιακά συστήματα θέρμανσης και ζεστού νερού χρήσης, το μαθηματικό υπόβαθρο που τα διέπει, οι παραδοχές που έγιναν, καθώς και έννοιες όπως οι γεωγραφικές συντεταγμένες, οι βαθμομέρες και μετεωρολογικά δεδομένα όλα απαραίτητα στοιχεία για την ανάπτυξη του λογισμικού προσομοίωσης.

Στο τρίτο κεφάλαιο παρουσιάζεται το λογισμικό που χρησιμοποιήθηκε για την ανάπτυξη του αλγορίθμου. Περιγράφεται λεκτικά και σχηματικά η γενική λειτουργία του και οι μεταβλητές του. Στη συνέχεια παρουσιάζεται το παράθυρο διεπαφής χρήστη και οι δυνατότητές του. Αναλύεται ο τρόπος αλληλεπίδρασης και συνεργασίας του γραφικού περιβάλλοντος, του λογιστικού φύλλου Excel και του Matlab. Ακολουθεί ο ορισμός του χρόνου προσομοίωσης και αναλυτική επεξήγηση του τρόπου λειτουργίας της προσομοίωσης βήμα – βήμα, πάντα σύμφωνα με όσα αναφέρθηκαν στα προηγούμενα κεφάλαια.

Το τέταρτο κεφάλαιο περιλαμβάνει την πιστοποίηση του πηγαίου λογισμικού το οποίο αναπτύχθηκε και περιγράφηκε στο τρίτο κεφάλαιο. Παρουσιάζεται η επίλυση πρότυπων προβλημάτων και ακολουθεί σύγκριση των αποτελεσμάτων που προκύπτουν από την εφαρμογή του κώδικα Matlab με δεδομένα βιβλιογραφίας ή με αποτελέσματα εμπορικών πακέτων, ώστε να πιστοποιηθεί η σωστή λειτουργία του.

Το πέμπτο κεφάλαιο περιλαμβάνει συμπεράσματα από την εκτέλεση του αλγορίθμου, εξηγεί τις αποκλίσεις που προκύπτουν, παραθέτει προτάσεις για τη βελτίωση του και εκθέτει τα προβλήματα και τις ατέλειες του.

Τέλος, στα παραρτήματα παρατίθενται το λογισμικό προσομοίωσης, ο κώδικας γραφικού περιβάλλοντος και πίνακες μετεωρολογικών δεδομένων του ΤΕΕ.

Κεφάλαιο 2. Σχεδιασμός και Προσομοίωση Ηλιοθερμικών Συστημάτων

Το υλικό του Κεφαλαίου 2 έχει ως βάση και κυρίως αναφέρεται στις Σημειώσεις του μαθήματος Ηλιακά Θερμικά Συστήματα του Προγράμματος Μεταπτυχιακών Σπουδών του Τμήματος Μηχανολόγων Μηχανικών του Πανεπιστημίου Θεσσαλίας.

2.1 Ηλιακή μηχανική και ΗΘΣ

Ο ήλιος αποτελεί ένα γιγαντιαίο θερμοπυρηνικό αντιδραστήρα σύντηξης με χημική σύσταση 74% υδρογόνο, 25% ήλιο και 1% άλλα στοιχεία. Μέσω αντιδράσεων μετατρέπει αδιάλειπτα υδρογόνο σε ήλιο ακτινοβολώντας τεράστιες ποσότητες ενέργειας. Στο κέντρο του η θερμοκρασία του υπολογίζεται στους 14.000.000 βαθμούς Kelvin, ενώ η θερμοκρασία της επιφάνειας του στους 5800 βαθμούς Kelvin. Εξαιτίας αυτών των θερμοκρασιών τα άτομα των στοιχείων που αποτελούν τη σύσταση του αστέρα έχουν χάσει τα ηλεκτρόνια τους και βρίσκονται σε μία κατάσταση νέφους θετικών και αρνητικών φορτίων που καλείται κατάσταση πλάσματος. Οι ταχύτατα κινούμενοι πυρήνες υδρογόνου συσσωματώνονται, υπερνικώντας τις μεταξύ τους απωστικές ηλεκτρομαγνητικές δυνάμεις και δημιουργούν πυρήνες του ηλίου (He). Η πυρηνική αυτή αντίδραση είναι εξώθερμη και χαρακτηρίζεται από την έκλυση τεράστιων ποσοτήτων ενέργειας, που ακτινοβολείται προς όλες τις κατευθύνσεις στο διάστημα.

Ο κύριος τρόπος μεταφοράς της ενέργειας αυτής είναι η ηλεκτρομαγνητική ακτινοβολία. Η ένταση της ηλεκτρομαγνητικής ακτινοβολίας που αντιστοιχεί στη μέση απόσταση ηλίου-πλανήτη ονομάζεται ηλιακή σταθερά. Για τον πλανήτη μας η ηλιακή σταθερά είναι: $I_{sc}=1367 \text{ W/m}^2$ και αντιστοιχεί στη μέση απόσταση γης – ηλίου που είναι $149,5 \times 10^6 \text{ km}$.

Η ενέργεια της ηλεκτρομαγνητικής ακτινοβολίας μεταδίδεται με τη μορφή φωτονίων και ισούται με :

$$E = h \nu = h (c/\lambda) \quad (2.1)$$

όπου $h = 6.626 \times 10^{-34}$ η σταθερά του Planck, $c = 2.9979 \times 10^8$ η ταχύτητα φωτονίων στο κενό, ν η συχνότητα της ηλιακής ακτινοβολίας και λ το μήκος κύματος των φωτονίων.

Το φάσμα της ηλιακής ακτινοβολίας είναι από 150nm έως 4000nm (Εικόνα 2.1). Το 9% της ηλιακής ακτινοβολίας έξω από την ατμόσφαιρα της γης ανήκει στο υπεριώδες φάσμα, το 44% στο ορατό φάσμα, ενώ το υπόλοιπο 47% στο υπέρυθρο.



Εικόνα 2.1 Θεωρητική και πραγματική ακτινοβολία.

Η ηλιακή ενέργεια στο σύνολό της είναι πρακτικά ανεξάντλητη, αφού προέρχεται από τον ήλιο, και ως εκ τούτου δεν υπάρχουν περιορισμοί για την εκμετάλλευσή της. Όσον αφορά την εκμετάλλευση της ηλιακής ενέργειας, θα μπορούσαμε να πούμε ότι χωρίζεται σε τρεις κατηγορίες εφαρμογών: τα παθητικά ηλιακά συστήματα, τα ενεργητικά ηλιακά συστήματα ή ηλιοθερμικά συστήματα, και τα φωτοβολταϊκά συστήματα. Τα παθητικά και τα ενεργητικά ηλιακά συστήματα εκμεταλλεύονται τη θερμότητα που εκπέμπεται μέσω της ηλιακής ακτινοβολίας και τη μετατρέπουν σε θερμική, ενώ τα φωτοβολταϊκά συστήματα στηρίζονται στη μετατροπή της ηλιακής ακτινοβολίας σε ηλεκτρικό ρεύμα μέσω του φωτοβολταϊκού φαινομένου. Στα πλαίσια αυτής της εργασίας θα ασχοληθούμε αποκλειστικά με την θερμική ενέργεια του ηλίου μέσω ακτινοβολίας.

2.1.1 Μεταβλητές που επηρεάζουν τη θέση του ηλίου

Όπως είναι γνωστό ο ήλιος και η γη βρίσκονται σε συνεχή κίνηση μεταξύ τους, πράγμα που καθιστά τη διαθέσιμη ηλιακή ενέργεια περιοδική, για τον λόγο αυτό θα πρέπει πρώτα να αναφέρουμε τις εξισώσεις που περιγράφουν την μεταξύ τους τροχιά. Σύμφωνα με τα παραπάνω η θέση του ηλίου στον ουρανό είναι συνάρτηση

- της τοποθεσίας της περιοχής που μελετάται επάνω στη γη,
- της χρονικής στιγμής στη διάρκεια του έτους και

- της ώρας στη διάρκεια της ημέρας.

Η τοποθεσία προσδιορίζεται από το γεωγραφικό πλάτος φ και γεωγραφικό μήκος λ . Η χρονική στιγμή στη διάρκεια του έτους προσδιορίζεται από την απόκλιση δ . Η ώρα κατά την διάρκεια της ημέρας προσδιορίζεται από την ωριαία ηλιακή γωνία ω . Οι τρεις αυτές μεταβλητές επηρεάζουν σχεδόν ολοκληρωτικά την διαθέσιμη ηλιακή ενέργεια που φτάνει στη γη και επομένως όλα τα είδη των ΗΘΣ. Για το λόγο αυτό κρίνεται απαραίτητο να αναλυθούν διεξοδικά πριν προχωρήσουμε στην προσομοίωση οποιοδήποτε τύπου ΗΘΣ.

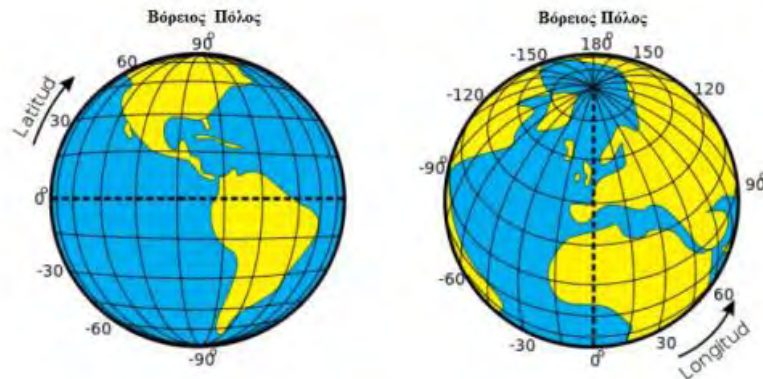
2.1.2 Γεωγραφικές συντεταγμένες

Οι Γεωγραφικές συντεταγμένες είναι ένα σύστημα συντεταγμένων με δύο μεγέθη από τα οποία προσδιορίζεται η θέση των διαφόρων τόπων στην επιφάνεια της γης. Ως βάση των γεωγραφικών συντεταγμένων λαμβάνεται ο ισημερινός και ο πρώτος μεσημβρινός. Οι γεωγραφικές συντεταγμένες αποτελούνται από το γεωγραφικό πλάτος και το γεωγραφικό μήκος.

Γεωγραφικό πλάτος (latitude) (φ) ενός σημείου που βρίσκεται στην επιφάνεια της γης είναι η γωνία που σχηματίζει η κατακόρυφος του τόπου με το επίπεδο του ισημερινού. Το γεωγραφικό πλάτος χαρακτηρίζεται Βόρειο Β (North N) ή Νότιο Ν (South S) ανάλογα σε ποιο ημισφαίριο βρίσκεται το σημείο. Σημεία του ίδιου ημισφαιρίου έχουν ομώνυμα πλάτη σε αντίθεση με σημείο διαφορετικού ημισφαιρίου που έχουν ετερόνυμα πλάτη. Το γεωγραφικό πλάτος μετράται σε μοίρες και υποδιαιρέσεις αυτών, πρώτα και δεύτερα. Οι τιμές που λαμβάνει κυμαίνονται από 0° - 90° Β ή 0° - 90° Ν (αρχής γενομένης της μέτρησης από τον ισημερινό του οποίου το γεωγραφικό πλάτος είναι 0°).

Γεωγραφικό μήκος (longitude) (λ) ενός σημείου στην επιφάνεια της γης είναι η γωνία που σχηματίζεται από το επίπεδο του μεσημβρινού που διέρχεται από το εν λόγω σημείο με το επίπεδο του πρώτου μεσημβρινού. Το γεωγραφικό μήκος χαρακτηρίζεται Ανατολικό Α (East E) ή Δυτικό Δ (West W) ανάλογα σε ποιο ημισφαίριο βρίσκεται το σημείο. Σημεία του ίδιου ημισφαιρίου έχουν ομώνυμα μήκη σε αντίθεση με τόπους διαφορετικού ημισφαιρίου που έχουν ετερόνυμα μήκη. Το γεωγραφικό μήκος μετράται σε μοίρες και υποδιαιρέσεις αυτών, πρώτα και δεύτερα. Οι τιμές που λαμβάνει κυμαίνονται από 0° - 180° Α ή 0° - 180° Δ (αρχής γενομένης της μέτρησης από τον πρώτο μεσημβρινό με γεωγραφικό μήκος 0°).

Οι έννοιες του γεωγραφικού μήκους και πλάτους απεικονίζονται στην Εικόνα 2.2:



Εικόνα 2.2 Απεικόνιση γεωγραφικού μήκους και πλάτους γης.

2.1.3 Ακτινοβολία λόγω διόρθωσης τροχιάς

Ηλιακή Σταθερά

Ως ηλιακή σταθερά ορίζεται η ροή της ηλιακής ακτινοβολίας που προσπίπτει σε μία μοναδιαία επιφάνεια κάθετη στις ακτίνες του ήλιου στο όριο της ατμόσφαιρας: $I_{sc}=1367 \text{ W/m}^2$. Οι τιμές στην βιβλιογραφία κυμαίνονται από 1353 W/m^2 έως 1395 W/m^2 . Για την παρούσα εργασία η ηλιακή σταθερά θα θεωρηθεί 1367 W/m^2 . Η ηλιακή σταθερά διορθώνεται λόγω της ελλειπτικής τροχιάς της γης γύρω από τον ήλιο και της εκκεντρότητάς της, επομένως λόγω της μεταβαλλόμενης απόστασης ηλίου-γης περιγράφεται από τη σχέση

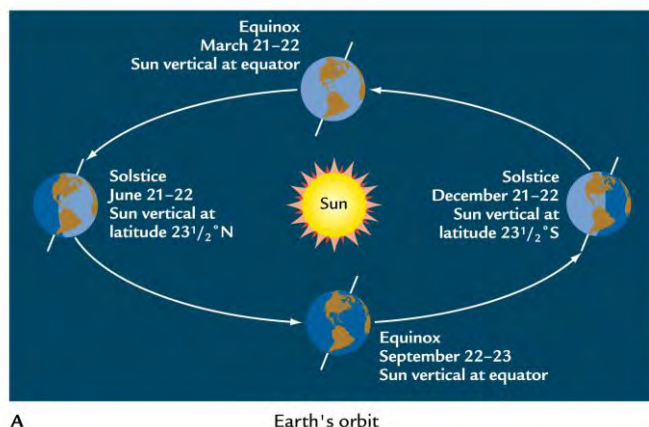
$$I_o(n) = I_{sc} \left[1 + 0.034 \cos \left(\frac{360 n}{365} \right) \right] \quad (2.2)$$

όπου $I_{sc} = 1367 \text{ W/m}^2$ η ηλιακή σταθερά και n ο αριθμός των ημερών μετρώντας από την 1η Ιανουαρίου. Η μεταβολή της ακτινοβολίας είναι περίπου $\pm 3\%$ με μέγιστη τιμή τις 31 Δεκεμβρίου και ελάχιστη τιμή την πρώτη Ιουλίου.

Ηλιακή απόκλιση

Ηλιακή απόκλιση $-23.45^\circ \leq \delta \leq 23.45^\circ$ είναι η γωνία που σχηματίζεται ανάμεσα στην ευθεία που ενώνει τα κέντρα γης - ήλιου στο επίπεδο του ισημερινού. Η γωνία αυτή αλλάζει κατά τη διάρκεια του έτους, με ακραίες τιμές στις $+23.45$ μοίρες στις 21 Ιουνίου και τις -23.45 μοίρες στις 21 Δεκεμβρίου (θερινό και χειμερινό

ηλιοστάσιο), ενώ παίρνει την τιμή μηδέν κατά τις ημέρες που έχουμε ισημερία (Εικόνα 2.3).

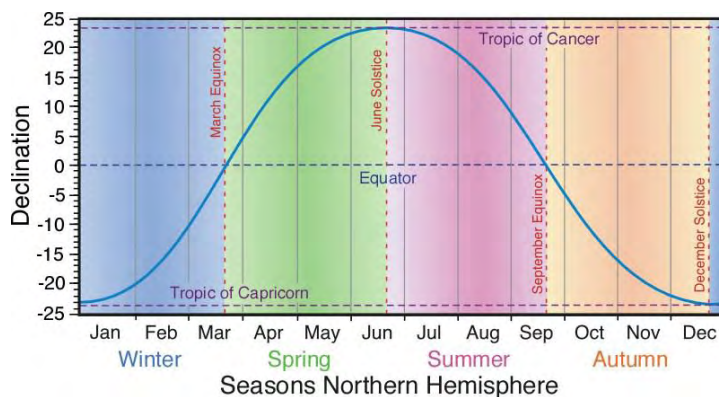


Εικόνα 2.3 Τροχία γής σε σχέση με ήλιο.

Η ηλιακή απόκλιση δίνεται από τη σχέση

$$\delta = 23.45 \sin \left[360 \left(\frac{284 + n}{365} \right) \right] \quad (2.3)$$

όπου n ο αριθμός των ημερών μετρώντας από την πρώτη Ιανουαρίου. Η γωνία αυτή είναι ιδιαίτερης σημασίας αφού πρακτικά υποδηλώνει τις εποχές της γης, όπως φαίνεται στην Εικόνα 2.4.



Εικόνα 2.4 Γωνία απόκλισης ανά μήνα του έτους για το βόρειο ημισφαίριο

2.1.4 Προσανατολισμός ΗΘΣ

Για την καλύτερη περιγραφή των ηλιακών συστημάτων πρέπει να καθορίσουμε τον προσανατολισμό του συλλέκτη, για αυτό ορίζουμε τις παρακάτω γωνίες:

Κλίση συλλέκτη β

Κλίση συλλέκτη β είναι η γωνία μεταξύ της επιφάνειας του συλλέκτη και του οριζόντιου επιπέδου. Παίρνει τιμές $0^\circ \leq \beta \leq 180^\circ$, για οριζόντιο επίπεδο ισχύει $\beta = 0^\circ$ ενώ για κατακόρυφο $\beta = 90^\circ$. Η κλίση του συλλέκτη β αποτελεί αντικείμενο ιδιαίτερης μελέτης, αφού για να μεγιστοποιηθεί η ετήσια προσπίπτουσα ακτινοβολία θα πρέπει η κλίση β να επιλεγεί ώστε η ηλιακή ακτινοβολία να πέφτει κάθετα στην επιφάνεια του συλλέκτη. Αυτό είναι αρκετά δύσκολο και δαπανηρό ώστε να εξασφαλισθεί, λόγω της συνεχής κίνησης του ηλίου. Για το λόγο αυτό σε ορισμένες εφαρμογές χρησιμοποιούνται διατάξεις και μηχανισμοί κατευθυνόμενοι είτε από φωτοκύτταρα είτε από υπολογιστές ώστε ο συλλέκτης να ακολουθεί την τροχιά του ηλίου σε όλη τη διάρκεια της ημέρας, καθ' όλη τη διάρκεια του έτους.

Αζιμούθια γωνία συλλέκτη γ

Αζιμούθια γωνία συλλέκτη γ είναι η γωνία μεταξύ του τοπικού μεσημβρινού (νότος) και της προβολής της καθέτου της επιφάνειας πάνω στον ορίζοντα.

Επιφάνεια που «βλέπει» νότο: $\gamma = 0$.

Επιφάνεια που «βλέπει» δυτικά: $\gamma > 0$.

Επιφάνεια που «βλέπει» ανατολικά: $\gamma < 0$.

Οριζόντιος συλλέκτης: $\gamma = 0$.

Γωνία πρόσπτωσης θ

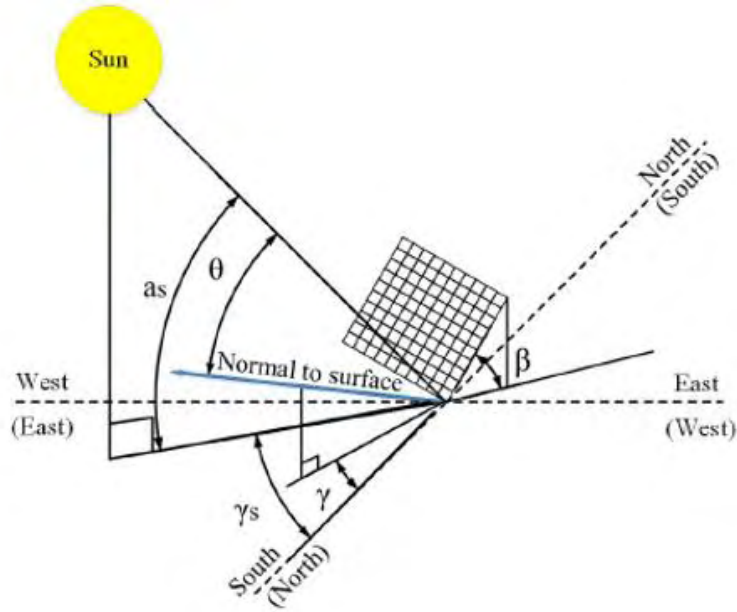
Γωνία πρόσπτωσης θ είναι η γωνία μεταξύ της ηλιακής ακτινοβολίας και της καθέτου της επιφάνειας. Για επίπεδη επιφάνεια με κλίση β και προσανατολισμό γ η γωνία δίνεται από τον τύπο:

$$\begin{aligned}
 \cos(\theta) &= \sin(\delta) \sin(\varphi) \cos(\beta) - \sin(\delta) \cos(\varphi) \sin(\beta) \cos(\gamma) \\
 &\quad + \cos(\delta) \cos(\varphi) \cos(\beta) \cos(\omega) + \cos(\delta) \sin(\varphi) \cos(\gamma) \cos(\omega) \\
 &\quad + \cos(\delta) \sin(\beta) \sin(\gamma) \sin(\omega) \\
 &= \cos(\theta_z) \cos(\beta) + \sin(\theta_z) \sin(\beta) \cos(\gamma_s - \gamma) \\
 &= \sin(a_s) \cos(\beta) + \cos(a_s) \sin(\beta) \cos(\gamma_s - \gamma) \quad (2.4)
 \end{aligned}$$

Στην περίπτωση που η επιφάνεια πρόσπτωσης είναι προσανατολισμένη Νότια, όπως συμβαίνει με τις περισσότερες εγκαταστάσεις στην Ελλάδα, η παραπάνω εξίσωση απλοποιείται ως εξής:

$$\cos(\theta) = \cos(\varphi - \beta) \cos(\delta) \cos(\omega) + \sin(\varphi - \beta) \sin(\delta) \quad (2.5)$$

Οι έννοιες αυτές εμφανίζονται και σχηματικά στην Εικόνα 2.5:



Εικόνα 2.5 Ηλιακές γωνίες.

Γωνία γ_s

Ηλιακή αζιμούθια γωνία γ_s είναι η γωνία μεταξύ του τοπικού μεσημβρινού και της προβολής της ηλιακής ακτινοβολίας στο οριζόντιο επίπεδο και υπολογίζεται μέσω της εξίσωσης:

$$\sin(\gamma_s) = \frac{\cos(\delta) \sin(\omega)}{\sin(\theta_z)} \quad (2.6)$$

Ανατολικά του μεσημβρινού $\gamma_s < 0$ και δυτικά του μεσημβρινού: $\gamma_s > 0$.

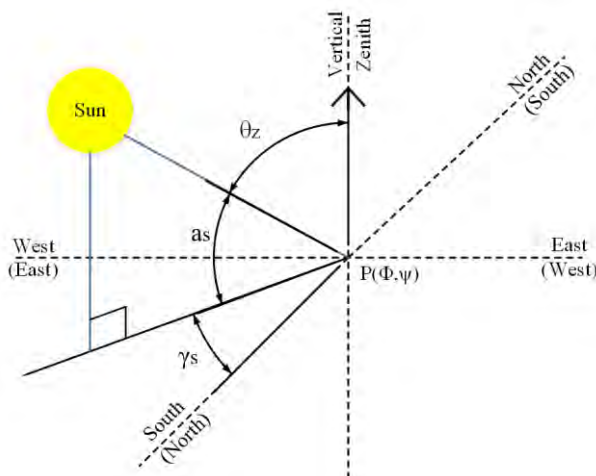
Γωνία θ_z

Ηλιακή ζενιθιακή γωνία θ_z είναι η γωνία μεταξύ του τοπικού ζενίθ και της ηλιακής ακτινοβολίας, δηλαδή η κλίση της ακτινοβολίας ως προς τον ορίζοντα. Υπολογίζεται μέσω της σχέσης:

$$\cos(\theta_z) = \cos(\delta) \cos(\varphi) \cos(\omega) + \sin(\varphi) \sin(\delta) \quad (2.7)$$

Ηλιακό ύψος a_s

Ηλιακό ύψος a_s είναι η γωνία μεταξύ του οριζοντα και της κατεύθυνσης της ηλιακής ακτινοβολίας, δηλαδή $a_s = 90^\circ - \theta_z$. Για την ανατολή του ηλίου: $a_s = 0$, ενώ για το ηλιακό μεσημέρι: $a_s = \text{μέγιστο}$. Η γωνία αυτή στο βόρειο ημισφαίριο είναι μέγιστη κατά το καλοκαιρινό και ελάχιστη κατά το χειμερινό ηλιοστάσιο. Όλες οι παραπάνω γωνίες εμφανίζονται συγκεντρωτικά στην Εικόνα 2.6:



Εικόνα 2.6 Γωνίες συλλέκτη.

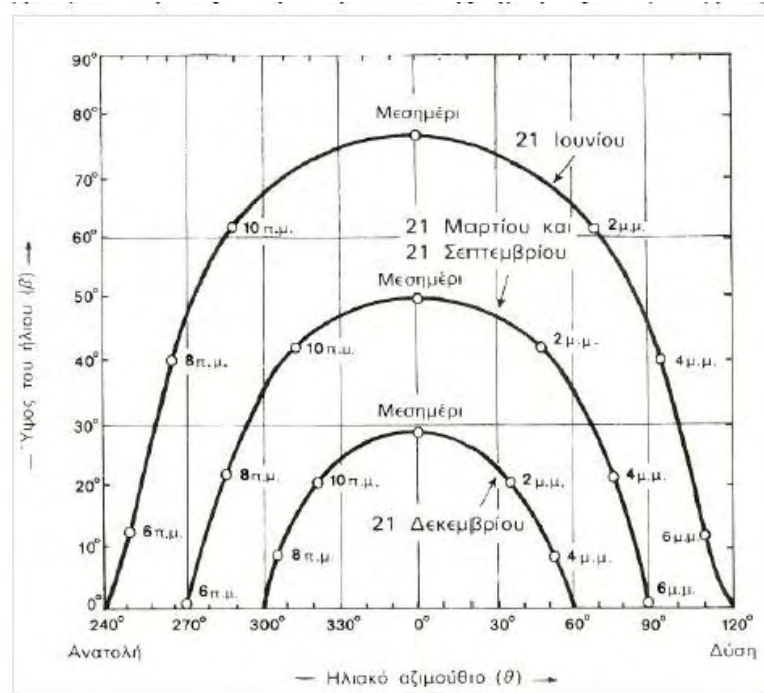
Προσδιορισμός της θέσης του ηλίου

Με βάση τις ποσότητες φ , δ και ω προκύπτουν από τις παρακάτω σχέσεις οι γωνίες γ_s και $a_s = 90 - \theta_z$, που προσδιορίζουν τη θέση του ήλιου. Για τις γωνίες αυτές ισχύει:

$$\sin(a_s) = \sin(\delta) \sin(\varphi) + \cos(\delta) \cos(\varphi) \cos(\omega) \quad (2.8)$$

$$\sin(\gamma_s) = \frac{\cos(\delta) \sin(\omega)}{\sin(\theta_z)} = \frac{\cos(\delta) \sin(\omega)}{\cos(a_s)} \quad (2.9)$$

Συνδυάζοντας όλα όσα αναφέραμε νωρίτερα, στην Εικόνα 2.7 φαίνεται ένα παράδειγμα των τροχιών που διαγράφει ο ήλιος στη διάρκεια της ημέρας σε γεωγραφικό πλάτος βόρειο 38° .



Εικόνα 2.7 Τροχιά του ηλίου για τυπικό γεωγραφικό πλάτος 38°N.

2.1.5 Ηλιακή ακτινοβολία

Ηλιακή ακτινοβολία στο όριο της ατμόσφαιρας σε οριζόντιο επίπεδο

Η ηλιακή ακτινοβολία σε οριζόντιο επίπεδο συμβολίζεται με H_o και δίδεται από τη σχέση:

$$H_o = I_o(n) \sin(a) = I_{sc} \left[1 + 0.034 \cos\left(\frac{360n}{365}\right) \right] \sin(a) \quad (2.10)$$

Ολοκληρώνοντας τη σχέση ως προς το χρόνο από την ανατολή έως τη δύση του ηλίου προκύπτει η μέση εξωγήινη ημερήσια ακτινοβολία σε οριζόντιο επίπεδο.

Άμεση ηλιακή ακτινοβολία

Η άμεση ηλιακή ακτινοβολία που δέχεται μία επιφάνεια προέρχεται απευθείας από τον ήλιο, χωρίς να μεσολαβήσει διασκορπισμός μέσα στην ατμόσφαιρα.

Διάχυτη ηλιακή ακτινοβολία

Διάχυτη ηλιακή ακτινοβολία ονομάζεται η ακτινοβολία που έχει υποστεί σκέδαση και αλλαγή κατεύθυνσης μέσα στην ατμόσφαιρα.

Ανακλώμενη ηλιακή ακτινοβολία

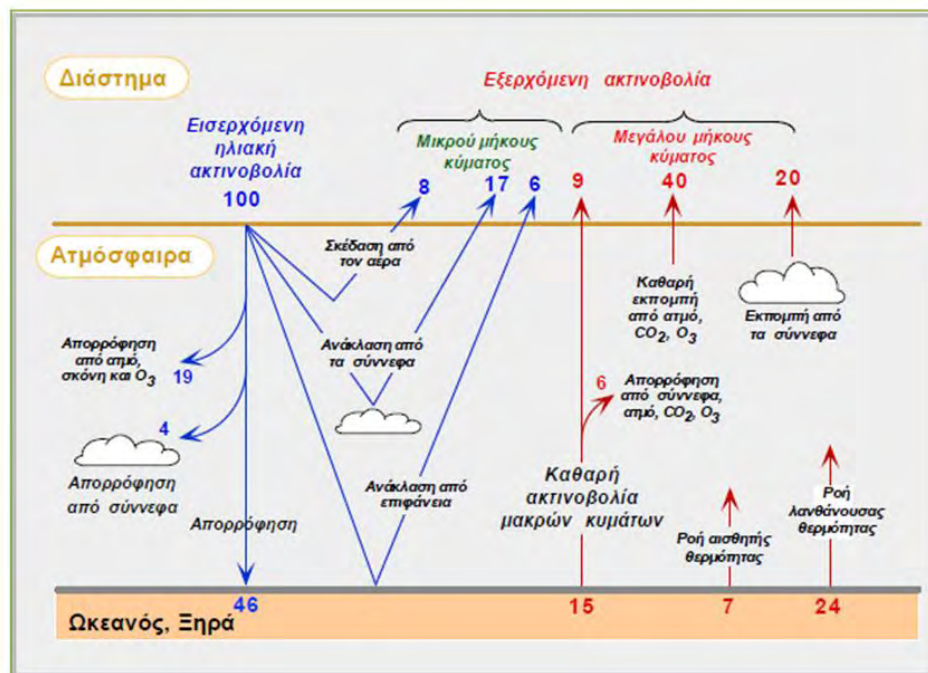
Η ανακλώμενη ηλιακή ακτινοβολία προκύπτει από την ανάκλαση της ακτινοβολίας που προσπίπτει στο έδαφος και χαρακτηρίζεται από τον συντελεστή αντανάκλασης, όπως φαίνονται στον Πίνακα 2.1.

Πίνακας 2.1 Τυπικοί συντελεστές αντανάκλασης ανάλογα με το περιβάλλον.

Επιφάνεια	Συντελεστής ανάκλασης (τ)
Καθαρό (φρέσκο) χιόνι	0,80 - 0,95
Βρώμικο παλιό χιόνι	0,40 - 0,70
Πάγος	0,20 - 0,40
Ήρεμη θάλασσα	0,03 - 0,30
Άμμος	0,20 - 0,45
Γρασίδι	0,15 - 0,25
Δάσος	0,15 - 0,20
Τσιμέντο	0,10 - 0,35
Άσφαλτος	0,05 - 0,20

Η ανακλώμενη ακτινοβολία εξαρτάται από τη μορφολογία, το χρώμα του εδάφους και την πυκνότητα των νεφών, ενώ η άμεση και διάχυτη ηλιακή ακτινοβολίες επηρεάζονται σημαντικά από τη σύσταση της ατμόσφαιρας τη δεδομένη χρονική στιγμή (π.χ. συννεφιά, υγρασία) την ημέρα κατά τη διάρκεια του έτους και τη γωνία πρόσπτωσης των ηλιακών ακτινών στην επιφάνεια του συλλέκτη.

Μία σχηματική απεικόνιση του τρόπου αλληλεπίδρασης της ανακλώμενης ακτινοβολίας με το περιβάλλον ακολουθεί στην Εικόνα 2.8.



Εικόνα 2.8 Ισοζύγιο ακτινοβολίας από και προς τη γή.

Προσπίπτουσα ηλιακή ακτινοβολία

Ο υπολογισμός της μέσης προσπίπτουσας ακτινοβολίας ανά ώρα, μέρα, μήνα ή έτος σε επιφάνειες (συλλέκτες, παράθυρα, κτίρια, κ.τ.λ.) είναι ιδιαίτερα δύσκολος για τους εξής δύο λόγους:

- Έλλειψη δεδομένων (με εξαίρεση συγκεκριμένες τοποθεσίες).
- Ο προσδιορισμός των «μέσων ακτινοβολιών» είναι επίπονος και το σημαντικότερο πολλές φορές όχι ιδιαίτερα χρήσιμος ή και παραπλανητικός στον λεπτομερή σχεδιασμό.

Αξίζει να αναφερθεί ότι η χρήσιμη ηλιακή-θερμική ενέργεια εντός δύο-ημερών εκ των οποίων η μία είναι χωρίς σύννεφα και η δεύτερη με πολλά σύννεφα είναι μεγαλύτερη από μία άλλη περίοδο 2-ημερών όπου και οι δύο μέρες είναι μερικώς συννεφιασμένες ακόμη και αν η συνολική προσπίπτουσα ακτινοβολία εντός των δύο διημέρων είναι η ίδια. Γενικά οι υπολογισμοί εξαρτώνται σε πολύ μεγάλο βαθμό από την διαθεσιμότητα δεδομένων.

Όργανα μέτρησης ηλιακής ακτινοβολίας

Η ηλιακή ακτινοβολία μετράται με πυρηλιόμετρα και πυρανόμετρα:

- Πυρηλιόμετρο: Όργανο που μετρά την άμεση ηλιακή ακτινοβολία κάθετα στη κατεύθυνση της ακτινοβολίας.
- Πυρανόμετρο: Όργανο που μετρά την ημισφαιρική ολική ηλιακή ακτινοβολία (άμεση και διάχυτη ακτινοβολία) σε οριζόντιο επίπεδο.

Συνήθως τα δεδομένα περιλαμβάνουν τη τοπική ημισφαιρική ολική ηλιακή ακτινοβολία (πυρανόμετρο) ή/και την άμεση ακτινοβολία (πυρηλιόμετρο). Χωρίς τη πληροφορία αυτή ακριβείς υπολογισμοί δεν είναι εφικτοί.

Διαδικασία υπολογισμού ηλιακής ακτινοβολίας

- Προσδιορισμός τοπικής ημισφαιρικής ολικής ηλιακής ακτινοβολίας.
- Προσδιορισμός άμεσης και διάχυτης ακτινοβολίας (ισότροπη διάχυτη ακτινοβολία).
- Προσδιορισμός γωνίας μεταξύ άμεσης ακτινοβολίας και κεκλιμένης επιφάνειας.
- Προσδιορισμός άμεσης, διάχυτης και ανακλώμενης ακτινοβολίας πάνω στην κεκλιμένη επιφάνεια.

Κατά την περίπτωση όπου η τοπική ημισφαιρική ολική ηλιακή ακτινοβολία H και άμεση ηλιακή ακτινοβολία I_b είναι γνωστές (ανά ώρα):

Η διάχυτη ακτινοβολία είναι :

$$H_d = H - H_b = H - I_b \sin(a) \quad (2.11)$$

ενώ η ηλιακή ακτινοβολία σε κεκλιμένη επιφάνεια υπολογίζεται από τη σχέση:

$$q_s = H_b R_b + H_d R_d + H \rho R_r = H \times R \quad (2.12)$$

όπου

$$R_b = \frac{\cos(\theta)}{\sin(a)}$$

$$R_d = \cos^2\left(\frac{\beta}{2}\right)$$

$$R_r = \sin^2\left(\frac{\beta}{2}\right)$$

$$R = \left(1 - \frac{H_o}{H}\right) R_b + \frac{H_d}{H} R_d + \rho R$$

Σε περίπτωση όπου μόνο η τοπική ημισφαιρική ολική ηλιακή ακτινοβολία H είναι γνωστή, διαθέτουμε δύο τρόπους υπολογισμού, οι οποίοι βασίζονται στη σχέση του συντελεστή αιθριότητας:

$$K_\tau = H/H_o \quad (2.13)$$

Πρώτος τρόπος

Αν ο λόγος του συντελεστή αιθριότητας ικανοποιεί τα όρια

$$0.3 \leq K_\tau \leq 0.85$$

τότε η ηλιακή ακτινοβολία είναι

$$I_b = -520 + 1800 K_\tau \quad (2.14)$$

διαφορετικά, αν ισχύει $K_\tau < 0.3$ τότε η ηλιακή ακτινοβολία μηδενίζεται, δηλαδή $I_b = 0$.

Δεύτερος τρόπος

Αν ο λόγος του συντελεστή αιθριότητας ικανοποιεί τα παρακάτω όρια:

$$K_\tau < 0.35$$

τότε η ηλιακή ακτινοβολία είναι

$$H_d/H = 1 - 0.249 K_\tau \quad (2.15)$$

Διαφορετικά, αν ισχύει $0.3 \leq K_\tau \leq 0.75$ τότε έχουμε

$$H_d/H = 1.557 - 1.84 K_\tau \quad (2.16)$$

αλλιώς $0.75 < K_\tau$ και υπολογίζουμε

$$H_d/H = 0.177 \quad (2.17)$$

Στην παρούσα εργασία θα χρησιμοποιηθεί ο πρώτος τρόπος, ενώ θα γίνει μία σύντομη σύγκριση αποτελεσμάτων μεταξύ αυτού και της απλής ημιτονοειδούς κατανομής η οποία χρησιμοποιείται σε πληθώρα εφαρμογών.

2.1.6 Ωρολογιακός και ηλιακός χρόνος

Ο ηλιακός χρόνος είναι αυτός όπου ο προσδιορισμός του βασίζεται στη γωνιακή μετακίνηση του ηλίου σε σχέση με τον τοπικό μεσημβρινό, επομένως δεν συμπίπτει απόλυτα με τον τοπικό ωρολογιακό χρόνο. Ηλιακό μεσημέρι θεωρείται η χρονική στιγμή που ο ήλιος μεσουρανή και τότε είναι η δωδέκατη ώρα, πρακτικά η στιγμή που ένας κατακόρυφος στύλος ρίχνει τη μικρότερη σκιά στον οριζόντιο επίπεδο. Οι δύο χρόνοι συνδέονται με την παρακάτω σχέση, όπου μεταβάλλεται κατά τη διάρκεια του έτους:

$$t_{solar} = t_{local} \pm 4(\text{min}/\mu\text{οίρα})(L_{st} - L_{local}) + E \quad (2.18)$$

όπου

L_{st} : Το γεωγραφικό μήκος ατράκτου.

L_{local} : Το γεωγραφικό μήκος της τοποθεσίας που μελετάται.

$$E = 0.0172 + 0.398 \cos(B) - 7.3456 \sin(B) - 3.3468 \cos^2(B) - 9.3544 \sin^2(B) \quad (2.19)$$

$$B = 360(n - 1)/365 \quad (2.20)$$

n : Ο αριθμός των ημερών μετρώντας από την πρώτη Ιανουαρίου.

Ωριαία γωνία

Ωριαία ηλιακή γωνία ω είναι η γωνία που έχει περιστραφεί η γη από το ηλιακό μεσημέρι, ανατολικά ή δυτικά με ρυθμό 15 μοίρες ανά ώρα. Σχετίζεται με τον ηλιακό χρόνο με την εξίσωση 2.21.

$$\omega = 15^\circ hr^{-1}(t_{solar} - 12) \quad (2.21)$$

Γωνία ώρας ανατολής και δύσης

Γωνία ώρας ανατολής και δύσης ω_{ss} είναι η γωνία για την οποία η ζενιθιακή γωνία ισούται με 90 μοίρες, $\theta_z = 90^\circ$, ή το ηλιακό ύψος $a_s = 0$ ισούται με μηδέν.

Για οριζόντιο επίπεδο υπολογίζεται θέτοντας:

$$a_s = 0 \rightarrow \sin(a_s) = 0 = \sin(\delta) \sin(\varphi) + \cos(\delta) \cos(\varphi) \cos(\omega_{ss})$$

$$\omega_{ss} = \cos^{-1}(-\tan(\varphi) \tan(\delta)) \quad (2.22)$$

Για κεκλιμένο επίπεδο ισχύει αν δ και φ έχουν το ίδιο πρόσημο (καλοκαίρι είτε βόριο είτε νότιο ημισφαίριο):

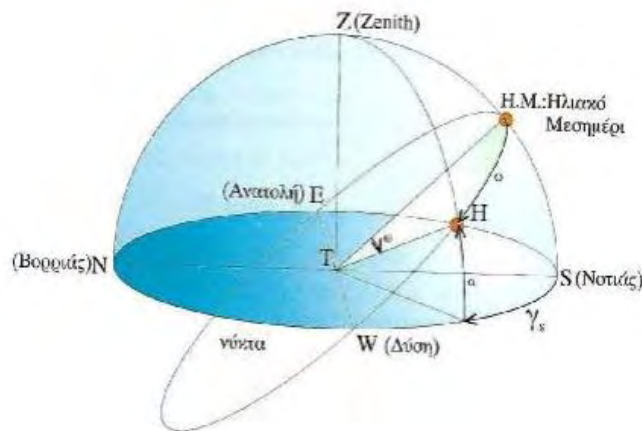
$$\omega_{ss} = \cos^{-1}(-\tan(\varphi - \beta) \tan(\delta)) \quad (2.23)$$

Ενώ, αν δ και φ έχουν το αντίθετο πρόσημο (χειμώνας είτε βόριο είτε νότιο ημισφαίριο):

$$\omega_{ss} = \cos^{-1}(-\tan(\varphi) \tan(\delta)) \quad (2.24)$$

Επομένως η διάρκεια σε ώρες από την ανατολή έως τη δύση δίνεται από τον τύπο:

$$t_d = 2/15^\circ \omega_{ss} = 2/15^\circ hr \cos^{-1}(-\tan(\varphi) \tan(\delta)) \quad (2.25)$$



Εικόνα 2.9 Ηλιακό μεσημέρι στο βόριο ημισφαίριο για μία τοποθεσία.

2.1.7 Υπολογισμός θερμοκρασίας νερού και εδάφους

Καταλυτικός παράγοντας για την μελέτη και προσομοίωση των ΗΘΣ για θέρμανση ζεστού νερού χρήσης αποτελεί η θερμοκρασία του νερού του δικτύου της εκάστοτε περιοχής, αφού αποτελεί καθοριστικό παράγοντα για τους υπολογισμούς. Παρόλα αυτά στην Ελλάδα η θερμοκρασία αυτή δεν έχει ερευνηθεί διεξοδικά ώστε να υπάρχουν σαφή και ακριβή στοιχεία. Το πρόβλημα αυτό αντιμετωπίζεται στην παρούσα εργασία, χρησιμοποιώντας την οδηγία του ΤΕΕ, με δύο τρόπους. Αφενός χρησιμοποιώντας πίνακες του ΕΛΟΤ 1291 που παρατίθεται στο παράρτημα, σύμφωνα δεδομένα του ΤΕΕ, και αφετέρου χρησιμοποιώντας τον υπολογισμό της θερμοκρασίας του εδάφους.

Η θερμοκρασία του εδάφους, θεωρώντας ομογενές έδαφος με σταθερό συντελεστή θερμικής διάχυσης, χωρίς πηγές ή καταβόθρες θερμότητας δίνεται από τη σχέση:

$$T(z) = T_m - A_s e^{-z\left(\frac{\pi}{365\alpha}\right)^{\frac{1}{2}}} \cos\left[\frac{2\pi}{365}\left(n - \nu - \frac{z}{2} \frac{365^{1/2}}{\pi\alpha}\right)\right] \quad (2.26)$$

όπου

n : Ο αριθμός των ημερών μετρώντας από την πρώτη Ιανουαρίου.

ν : Η ημέρα του έτους κατά την οποία η θερμοκρασία της επιφάνειας του εδάφους λαμβάνει τη χαμηλότερή της τιμή, για την Ελλάδα η τιμή αυτή θεωρείται 30 Ιανουαρίου.

T_m : Η μέση ετήσια θερμοκρασία επιφάνειας εδάφους.

A_s : Το πλάτος της ετήσιας διακύμανσης της θερμοκρασίας της επιφάνειας του εδάφους.

α : Ο συντελεστής θερμικής διάχυσης του εδάφους για τον οποίο ακολουθεί σύντομη αναφορά.

Ο συντελεστής θερμικής διάχυσης του εδάφους, ορίζεται ως ο λόγος του συντελεστή θερμικής αγωγιμότητας του k προς το γινόμενο της πυκνότητας του ρ , επί το συντελεστή ειδικής θερμότητας υπό σταθερή πίεση C_p και δίνεται από τον τύπο:

$$\alpha = \frac{k}{\rho C_p} \quad (2.27)$$

Στα πλαίσια της εργασίας θα χρησιμοποιηθούν τιμές από τη βιβλιογραφία σύμφωνες με τον Πίνακα 2.2.

Πίνακας 2.2 Συντελεστές θερμικής διάχυσης του εδάφους.

Τύπος Εδάφους	α ($\times 10^{-8} \cdot \text{m}^2 \cdot \text{s}^{-1}$)
Γρανίτης	0,49 - 0,70
Ασβεστόλιθος	0,80 - 1,83
Μάρμαρο	1,39
Ψαμμίτης	1,06 - 1,26
Αργίλος	0,139

Η θερμοκρασία του νερού δικτύου θεωρείται ίση με τη θερμοκρασία εδάφους σε βάθος 0.5 μέτρα που υπολογίζεται με βάση της μέσης θερμοκρασίας αέρα της εκάστοτε περιοχής, σύμφωνα με τις εξισώσεις 2.26 - 2.27 και τον τρόπο που αναλύθηκε παραπάνω.

2.1.8 Βαθμομέρες θέρμανσης και υπολογισμός φορτίων θέρμανσης

Η πιο απλή και σύντομη μέθοδος υπολογισμού των απαιτούμενων φορτίων θέρμανσης ενός κτιρίου, είναι οι βαθμομέρες θέρμανσης. Η παράμετρος αυτή προκύπτει από την μέση μηνιαία θερμοκρασία εικοσιτετραώρου για τους χειμερινούς μήνες σε μια περιοχή και μια θερμοκρασία αναφοράς η οποία συνήθως λαμβάνεται ίση με 18 °C. Η εξίσωση υπολογισμού των βαθμομερών θέρμανσης DD είναι:

$$DD = \sum [n (T_{ref} - T_a)^+] \quad (2.28)$$

όπου T_{ref} η θερμοκρασία αναφοράς, T_a η μηνιαία θερμοκρασία αέρα περιβάλλοντος, n ο αριθμός ημερών για κάθε μήνα.

Για τον υπολογισμό των απαιτούμενων φορτίων ενός κτηρίου χρησιμοποιούμε την εξίσωση:

$$Q = U_m A DD \quad (2.29)$$

όπου Q οι θερμικές απώλειες από το κτίριο, U_m ο μέσος συντελεστής θερμοπερατότητας του κτιρίου, A η επιφάνεια του κτηρίου και DD οι βαθμομέρες θέρμανσης.

2.1.9 Στοιχεία από ΤΕΕ

Στα πλαίσια της προσομοίωσης χρησιμοποιήθηκαν στοιχεία από το ΤΕΕ [18] ώστε τα αποτελέσματα της εργασίας να ανταποκρίνονται σε ρεαλιστικά δεδομένα και οι εκτιμήσεις της να είναι όσο το δυνατόν πιο κοντά στην πραγματικότητα. Σε αυτή την παράγραφο αναφέρονται οι πίνακες των δεδομένων που διαχειρίζεται η προσομοίωση, αυτά αφορούν:

- την τοποθεσία της πόλης υπό μελέτη
- τη μέση μηνιαία θερμοκρασία εικοσιτετράωρου
- τη μέση μηνιαία ολική ηλιακή ακτινοβολία στο οριζόντιο επίπεδο
- το μέσο μηνιαίος συντελεστή αιθριότητας
- τη μέση θερμοκρασία δικτύου νερού σύμφωνα με ΕΛΟΤ 1291
- τη μέγιστη θερμοκρασία εικοσιτετράωρου
- τη μέγιστη διακύμανση θερμοκρασίας εικοσιτετράωρου
- τις βαθμοημέρες θέρμανσης με θερμοκρασία αναφοράς 18 βαθμούς κελσίου.

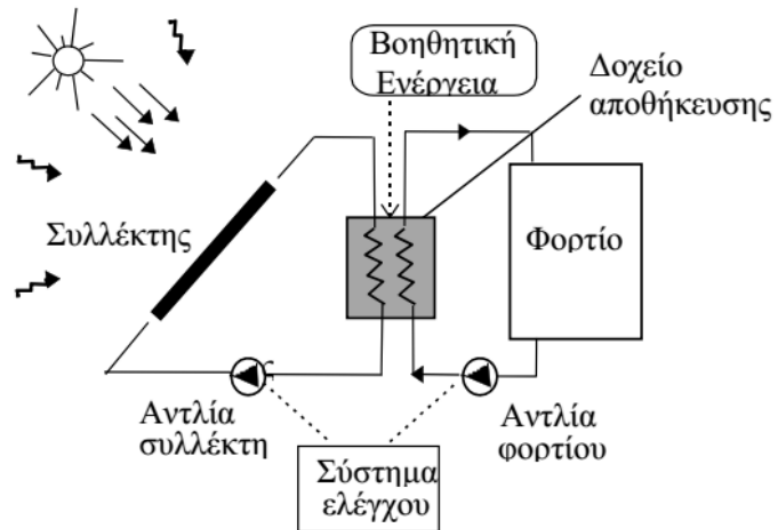
Οι αναλυτικοί πίνακες βρίσκονται στο παράρτημα Γ της εργασίας.

2.2 Λειτουργία των Ηλιακών Θερμικών Συστημάτων

2.2.1 Τυπική διάταξη ΗΘΣ θέρμανσης

Στην Εικόνα 2.10 παρουσιάζεται ένα τυπικό ΗΘΣ. Τα βασικά μέρη του συστήματος είναι :

- Ο ηλιακός συλλέκτης όπου η προσπίπτουσα ηλιακή ενέργεια μετατρέπεται σε θερμική ενέργεια του φέροντος ή θερμοαπαγωγού ρευστού.
- Το δοχείο αποθήκευσης όπου αποθηκεύεται η θερμική ενέργεια.
- Οι κυκλοφορητές που υποστηρίζουν την κίνηση των ρευστών.
- Η βοηθητική πηγή ενέργειας η οποία παρέχει την απαιτούμενη ενέργεια όταν η ηλιακή ενέργεια δεν επαρκεί για την κάλυψη των αναγκών.
- Το σύστημα ελέγχου.

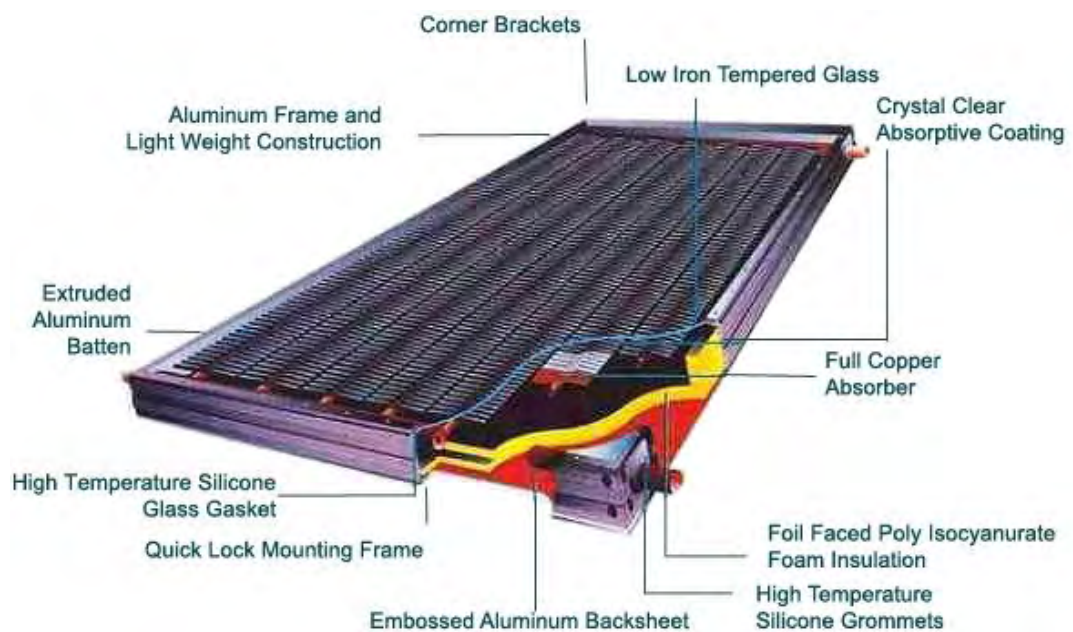


Εικόνα 2.10 Τυπική διάταξη ενός ΗΘΣ.

Στη συνέχεια θα παρουσιαστεί η μαθηματική περιγραφή των βασικών στοιχείων των ΗΘΣ που χρησιμοποιήθηκαν στο πηγαίο κώδικα.

2.2.2 Ανάλυση ηλιακού συλλέκτη

Από τα βασικότερα στοιχεία ενός ΗΘΣ συστήματος είναι ο επίπεδος ηλιακός συλλέκτης (Εικόνα 2.11).



Εικόνα 2.11 Τυπικά μέρη ενός ηλιακού συλλέκτη.

Η χρήσιμη ενέργεια ενός ηλιακού συλλέκτη δίνεται από τον τύπο :

$$\begin{aligned}\dot{Q}_u &= A_{\sigma\upsilon\lambda} q_s(t) = \\ &= (\tau\alpha)_{eff} q_s(t) A_{\sigma\upsilon\lambda} - \bar{U}A_{\pi\lambda}(\bar{T}_{\pi\lambda} - T_\alpha) - \varepsilon_{eff}A_{\pi\lambda}\sigma(\bar{T}_{\pi\lambda}^4 - T_\alpha^4) \quad (2.30)\end{aligned}$$

όπου

$A_{\sigma\upsilon\lambda}$: Η επιφάνεια του συλλέκτη.

$q_s(t)$: Η προσπίπτουσα ακτινοβολία στον συλλέκτη.

$(\tau\alpha)_{eff}$: Συνολικό κλάσμα απορρόφησης του συλλέκτη.

\bar{U} : Ο συνολικός μέσος συντελεστής θερμικών απωλειών.

$\bar{T}_{\pi\lambda}$: Μέση θερμοκρασία πλάκας απορρόφησης.

T_α : Θερμοκρασία περιβάλλοντος.

ε_{eff} : Σύνθετος συντελεστής εκπομπής.

σ : Σταθερά Stephan-Boltzman ($5.67 \times 10^{-8} \text{ W/m}^2\text{K}^4$).

Ο βαθμός απόδοσης του ηλιακού συλλέκτη είναι ο λόγος της χρήσιμης θερμικής ενέργειας Q_u προς τη διαθέσιμη προσπίπτουσα ηλιακή ενέργεια και δίνεται από τον τύπο

$$n = \frac{\dot{Q}_u}{A_{\sigma\upsilon\lambda}q_s} \approx (\tau\alpha)_{eff} - \frac{\bar{U}A_{\pi\lambda}(\bar{T}_{\pi\lambda} - T_\alpha)}{A_{\sigma\upsilon\lambda}q_s} \quad (2.31)$$

Ορίζοντας τις αδιάστατες μεταβλητές:

$$b = \frac{\bar{U}A_{\pi\lambda}T_\alpha}{A_{\sigma\upsilon\lambda}q_{s,ref}} \quad (2.32)$$

$$\bar{\theta}_{\pi\lambda} = \frac{\bar{T}_{\pi\lambda}}{T_\alpha} \quad (2.33)$$

$$\theta_\alpha = \frac{T_\alpha}{T_\alpha} \quad (2.34)$$

$$\psi(t) = \frac{q_s(t)}{q_{s,ref}} \quad (2.35)$$

η εξίσωση 2.31 μπορεί να γραφεί στη μορφή

$$n = \frac{Q_u}{A_{\sigma\upsilon\lambda} q_s} \approx (\tau\alpha)_{eff} - \frac{b (\Theta_{\pi\lambda} - \Theta_\alpha)}{\psi(t)} \quad (2.36)$$

όπου T_a και $q_{s,ref}$ η μέση ημερήσια θερμοκρασία και η προσπίπτουσα ακτινοβολία αναφοράς αντίστοιχα.

Επίσης, είναι επιθυμητό να γνωρίζουμε τον βαθμό απόδοσης σε σχέση με τη θερμοκρασία εισόδου του φέροντος ρευστού στον συλλέκτη. Ξαναγράφοντας τις εξισώσεις 2.31 και 2.36 έχουμε:

$$\begin{aligned} n &= (\tau\alpha)_{eff} - \frac{\bar{U}A_{\pi\lambda}(\bar{T}_{\pi\lambda} - T_\alpha)}{A_{\sigma\upsilon\lambda}q_s} = (\tau\alpha)_{eff} - \frac{b (\bar{\Theta}_{\pi\lambda} - \Theta_\alpha)}{\psi(t)} = \\ &= F_R \left[(\tau\alpha)_{eff} - \frac{\bar{U}A_{\pi\lambda}(T_{fin} - T_\alpha)}{q_s A_{\sigma\upsilon\lambda}} \right] = F_R \left[(\tau\alpha)_{eff} - \frac{b (\Theta_{fin} - \Theta_\alpha)}{\psi(t)} \right] \end{aligned} \quad (2.37)$$

με τον συντελεστή F_R να εκφράζει το κλάσμα της ωφέλιμης ενέργειας ως προς τη θερμική ενέργεια του συλλέκτη αν η μέση θερμοκρασία της πλάκας απορρόφησης ήταν ίση με τη θερμοκρασία εισόδου του φέροντος ρευστού στον συλλέκτη και δίνεται από τον τύπο:

$$F_R = \frac{(\tau\alpha)_{eff} - \frac{\bar{U}A_{\pi\lambda}(\bar{T}_{\pi\lambda} - T_\alpha)}{q_s A_{\sigma\upsilon\lambda}}}{(\tau\alpha)_{eff} - \frac{\bar{U}A_{\pi\lambda}(T_{fin} - T_\alpha)}{q_s A_{\sigma\upsilon\lambda}}} = \frac{(\tau\alpha)_{eff} - \frac{b (\bar{\Theta}_{\pi\lambda} - \Theta_\alpha)}{\psi(t)}}{(\tau\alpha)_{eff} - \frac{b (\Theta_{fin} - \Theta_\alpha)}{\psi(t)}} \quad (2.38)$$

Με βάση τα παραπάνω και από την εξίσωση 2.37 το χρήσιμο φορτίο του συλλέκτη γράφεται στη μορφή:

$$\begin{aligned} \dot{Q}_u(t) &= n A_{\sigma\upsilon\lambda} q_s(t) = \dot{m}_c C (T_{f,out} - T_{f,in}) \\ &= F_R \left[(\tau\alpha)_{eff} - \frac{\bar{U}A_{\pi\lambda}(T_{fin} - T_\alpha)}{q_s A_{\sigma\upsilon\lambda}} \right] A_{\sigma\upsilon\lambda} q_s(t) = \\ &= F_R \left[(\tau\alpha)_{eff} - \frac{b (\Theta_{fin} - \Theta_\alpha)}{\psi(t)} \right] A_{\sigma\upsilon\lambda} q_s(t) \end{aligned} \quad (2.39)$$

όπου

m_c : Η μαζική παροχή του φέροντος ρευστού το οποίο εισέρχεται στον συλλέκτη.

$T_{f,out}$: Η θερμοκρασία εξόδου του ρευστού από τον συλλέκτη.

$T_{f,in}$: Η θερμοκρασία εισόδου του ρευστού στον συλλέκτη.

C : Η ειδική θερμοχωρητικότητα του ρευστού.

Από την εξίσωση 2.39 επιλύοντας ως προς $T_{f,out}$ προκύπτει :

$$T_{f,out} = T_{f,in} \left(1 - \frac{F_R b}{\gamma} \right) + \frac{F_R \bar{T}_\alpha}{\gamma} [(\tau\alpha)_{eff} \psi(t) + b \theta_\alpha] \quad (2.40)$$

και

$$\gamma = \frac{m_c C \bar{T}_\alpha}{A_{\sigma\sigma\lambda} q_{s,ref}} \quad (2.41)$$

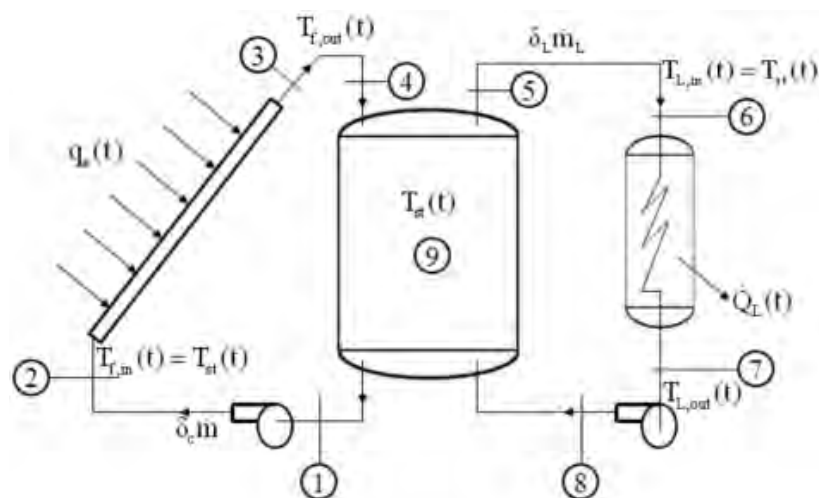
όπου το γ εκφράζει την ικανότητα του φέροντος ρευστού να απάγει ενέργεια από τον συλλέκτη προς την ηλιακή θερμότητα αναφοράς και μεταβάλλεται από το μηδέν μέχρι πολύ υψηλές τιμές.

Στην περίπτωση που δεν κυκλοφορεί ρευστό στον συλλέκτη τότε από την εξίσωση 2.37 και την εξίσωση του βαθμού απόδοσης για $n = 0$ προκύπτει η θερμοκρασία της πλάκας απορρόφησης για συνθήκες μηδενικής μαζικής παροχής.

$$T_{stag} = T_\alpha + \frac{(\tau\alpha)_{eff} q_s A_{\sigma\sigma\lambda}}{\bar{U} A_{\pi\lambda}} = T_\alpha + \frac{(\tau\alpha)_{eff} \psi(t) \bar{T}}{b} \quad (2.42)$$

2.2.3 Μοντέλο λειτουργίας ηλιακού θερμικού συστήματος για θέρμανση

Στην Εικόνα 2.12 παρουσιάζεται ένα τυπικό ΗΘΣ θέρμανσης.



Εικόνα 2.12 Τυπική διάταξη και διάγραμμα ροής ΗΘΣ για θέρμανση

Τα θερμικά ισοζύγια που το περιγράφουν δίνονται από τους παρακάτω τύπους:

$$T_{stag} = T_{\alpha} + \frac{(\tau\alpha)_{eff} \psi(t)}{b} \bar{T}_{\alpha} = T_{\alpha} + \frac{(\tau\alpha)_{eff} A_{\sigma\lambda} q_s}{\bar{U} A_{\pi\lambda}} \quad (2.43)$$

$$T_{f,out} = T_{f,in} \left(1 - \frac{F_R b}{\gamma}\right) + \frac{F_R \bar{T}_a}{\gamma} + [(\tau\alpha)_{eff} \psi(t) + b \theta_{\alpha}] \quad (2.44)$$

$$\dot{Q}_L = \delta_L m_L c (T_{Lin} - T_{Lout}) \quad (2.45)$$

$$\dot{Q}_{st}(t) = m_{st} C \frac{dT_{st}}{dt} = \delta_{\Sigma} \dot{Q}_u - \delta_L \dot{Q}_L - U_{st} A_{st} (T_{st} - T_a) \quad (2.46)$$

$$Q_u = \dot{m}_c C (T_{f,out} - T_{f,in}) = A [(\tau\alpha)_{eff} q_s - \bar{U} (\bar{T}_{\pi\lambda} - T_a)] \quad (2.47)$$

Θέση 1: Η θερμοκρασία του φέροντος ρευστού είναι ίση με τη θερμοκρασία του ρευστού εντός του αποθηκευτικού δοχείου και ίση με $T_{st}(t)$.

Θέση 2: Η θερμοκρασία του ρευστού είναι $T_{f,in}(t)$ και στο παρόν απλό μοντέλο θεωρούμε ότι είναι ίση με $T_{st}(t)$, δηλαδή δεν έχουμε θερμικές απώλειες από τη θέση 1 στη θέση 2.

Στη συνέχεια βρίσκουμε τη θερμοκρασία της πλάκας στη περίπτωση μηδενικής παροχής T_{stag} (εξίσωση 2.43).

Θέση 3: Η θερμοκρασία του ρευστού είναι $T_{f,out}(t)$ και υπολογίζεται με βάση την εξίσωση 2.44.

Θέση 4: Η θερμοκρασία του ρευστού είναι $T_{f,out}(t)$, δηλαδή δεν έχουμε θερμικές απώλειες από τη θέση 3 στη θέση 4.

Η παροχή από τη θέση 1 στη θέση 4 είναι $Q_{(3-4)} = \delta_c m_c$ όπου δ_c είναι μηδέν ή ένα αναλόγως εάν ο κυκλοφορητής είναι **ON** ή **OFF** και m_c η παροχή μάζας του φέροντος ρευστού.

Θέση 5: Η θερμοκρασία του ρευστού χρήσης είναι ίση με τη θερμοκρασία του ρευστού εντός του αποθηκευτικού δοχείου και ίση με $T_{st}(t)$.

Θέση 6: Η θερμοκρασία του ρευστού είναι $T_{l,in}(t)$, και στο παρόν απλό μοντέλο θεωρούμε ότι είναι ίση με $T_{st}(t)$, δηλαδή δεν έχουμε θερμικές απώλειες από τη θέση 5 στη θέση 6.

Θέση 7: Η θερμοκρασία του ρευστού είναι $T_{l,out}(t)$, και υπολογίζεται με βάση την εξίσωση 2.45.

Θέση 8: Η θερμοκρασία του ρευστού είναι $T_{l,out}(t)$, δηλαδή δεν έχουμε θερμικές απώλειες από τη θέση 7 στη θέση 8.

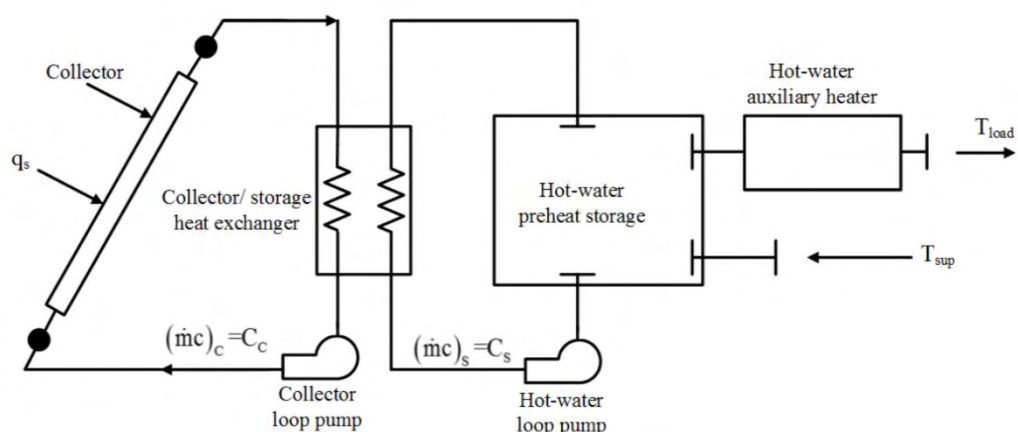
Η παροχή από τη θέση 5 στη θέση 8 είναι $Q_{(5-8)} = \delta_L m_L$ όπου δ_L είναι μηδέν ή ένα αναλόγως εάν ο κυκλοφορητής είναι **ON** ή **OFF** και m_L η παροχή μάζας του φέροντος ρευστού.

Θέση 9: Η θερμοκρασία εντός του αποθηκευτικού δοχείου $T_{st}(t)$ και υπολογίζεται με βάση τις εξισώσεις 2.45 και 2.46. Πιο συγκεκριμένα από την Εξίσωση 2.47 βρίσκουμε το Q_u και στη συνέχεια από την εξίσωση 2.45 το $Q_{st}(t)$ και το $T_{st}(t + \Delta t)$.

Με βάση τα παραπάνω και το σύστημα ελέγχου που θα δούμε σε επόμενη παράγραφο μοντελοποιείται η χρονικά μεταβαλλόμενη λειτουργία του συστήματος.

2.2.4 Εναλλάκτης και κύκλωμα Ζεστού Νερού Χρήσης

Το κύκλωμα του συστήματος για την παραγωγή ΖΝΧ παρουσιάζεται στην Εικόνα 2.13.



Εικόνα 2.13 Τυπική διάταξη και διάγραμμα ροής ΗΘΣ για ΖΝΧ.

Στην περίπτωση όπου υπάρχει εναλλάκτης στο σύστημα, η εξίσωση 2.30 πρέπει να τροποποιηθεί ώστε να η χρήσιμη ενέργεια να υπολογίζεται λαμβάνοντας υπ' όψιν τις απώλειες λόγω της παρουσίας του εναλλάκτη. Η θερμική ενέργεια που μεταφέρει το φέρον ρευστό είναι:

$$\dot{Q}_u = (\dot{m}C)_{\sigma\upsilon\lambda} (T_{f,out} - T_{f,in}) \quad (2.48)$$

Η μεταφερόμενη θερμική ενέργεια στον εναλλάκτη δίνεται από τη σχέση:

$$\dot{Q}_u = \varepsilon(\dot{m}C) (T_{f,out} - T_{e,in}) \quad (2.49)$$

όπου ο δείκτης ε συμβολίζει τον εναλλάκτη (exchanger) και συνεπώς $T_{e,in}$ η θερμοκρασία εισόδου του ρευστού (νερό) στον εναλλάκτη.

Για εναλλάκτη αντιροής ισχύει:

$$(\dot{m}C)_{min} = \min[(\dot{m}C)_{\sigma\upsilon\lambda}, (\dot{m}C)_e] \quad (2.50)$$

εάν $C_r < 1$

$$\varepsilon = \frac{1 - \exp[-NTU(1 - C_r)]}{1 - C_r \exp[-NTU(1 - C_r)]} \quad (2.51)$$

εάν $C_r = 1$

$$\varepsilon = \frac{NTU}{1 + NTU} \quad (2.52)$$

όπου

$$C_r = \frac{(\dot{m}C)_{min}}{(\dot{m}C)_{max}} \quad (2.53)$$

$$NTU = \frac{(UA)_{\varepsilon\upsilon\alpha\lambda\lambda\acute{\alpha}\kappa\tau\eta}}{(\dot{m}C)_{min}} \quad (2.54)$$

Επίσης, η θερμική ενέργεια στο ρευστό που κυκλοφορεί στον εναλλάκτη δίνεται από τη σχέση:

$$\dot{Q}_u = (\dot{m}C)_e (T_{e,out} - T_{e,in}) \quad (2.55)$$

Συνδυάζοντας τις παραπάνω σχέσεις προκύπτει:

$$T_{f,in} = T_{f,out} - \frac{\dot{Q}_u}{(\dot{m}C)_{\sigma\upsilon\lambda}} \quad (2.56)$$

$$T_{f,out} = T_{e,in} + \frac{\dot{Q}_u}{\varepsilon(\dot{m}C)_{min}} \quad (2.57)$$

$$T_{e,out} = T_{e,in} + \frac{\dot{Q}_u}{(\dot{m}C)_e} \quad (2.58)$$

Με αντικατάσταση στην εξίσωση 2.30 τελικά προκύπτει πως το χρήσιμο φορτίο λαμβάνοντας υπ' όψιν την παρουσία εναλλάκτη αντιροής είναι:

$$Q_u = \frac{F_R A [q_s - U(T_{e,in} - T_\alpha)]}{1 + \frac{F_R U A}{(\dot{m}C)_{\sigma\upsilon\lambda}} \left[\frac{(\dot{m}C)_{\sigma\upsilon\lambda}}{\varepsilon(\dot{m}C)_{min}} - 1 \right]} = F_R' A [q_s - U(T_{e,in} - T_\alpha)] \quad (2.59)$$

Η ποσότητα F_R'/F_R είναι πάντα μικρότερη της μονάδας και εκφράζει το μέτρο απόδοσης του εναλλάκτη και δίνεται από τον τύπο:

$$F_R'/F_R = \left[1 + \frac{A F_R U}{(\dot{m}C)_\Sigma} \left(\frac{(\dot{m}C)_\Sigma}{(\dot{m}C)_{min}} - 1 \right) \right]^{-1} \quad (2.60)$$

Επίσης δεν ισχύει $T_{st} = T_{L,out} = T_{f,in}$, και για τη θερμοκρασία εξόδου τροποποιούμε την εξίσωση 2.44 ως εξής:

$$T_{f,out} = T_{f,in} - T_{L,out} \left(\frac{F_R' b}{\gamma} \right) + \frac{F_R' \bar{T}_a}{\gamma} + [(\tau\alpha)_{eff} \psi(t) + b \theta_\alpha] \quad (2.61)$$

Τέλος το φορτίο κάλυψης στην περίπτωση όπου $T_{st} < T_{load}$ δίνεται από την εξίσωση:

$$f = \frac{T_{st} - T_{sup}}{T_{load} - T_{sup}} \quad (2.62)$$

Περιγραφή Φορτίου Ζήτησης για ZNX

Χρησιμοποιώντας τους παραπάνω τύπους πρέπει να γίνει περιγραφή του φορτίου που πρέπει να καλύπτεται από το ΗΘΣ για ZNX. Το φορτίο που πρέπει να εξυπηρετηθεί δίνεται από τη σχέση:

$$\dot{Q}_L(t) = \delta_L \dot{m}_L C (T_{L,in} - T_{L,out}) \quad (2.63)$$

όπου

$\dot{Q}_L(t)$: Το θερμικό φορτίο, το οποίο συνήθως δίδεται από τον χρήστη και αποτελεί μία από τις προδιαγραφές που πρέπει να λάβει υπόψη του ο μελετητής.

\dot{m}_L : Η μαζική παροχή του ρευστού χρήσης (τις περισσότερες φορές είναι νερό) και c η θερμοχωρητικότητα του.

$T_{L,in}$: Η θερμοκρασία εσόδου του ρευστού στην κατανάλωση.

$T_{L,out}$: Η θερμοκρασία εξόδου του ρευστού από την κατανάλωση (η θερμοκρασία επιστροφής).

δ_L : Διακόπτης κυκλώματος φορτίου. Λαμβάνει τις τιμές 0 και 1 και υποδηλώνει εάν ο κυκλοφορητής είναι ανοιχτός ή κλειστός βάσει διαφόρων συνθηκών ελέγχου οι οποίες ορίζονται από τον μελετητή.

2.2.5 Δεξαμενή αποθήκευσης

Το δοχείο αποθήκευσης είναι μία δεξαμενή καλά μονωμένη η οποία περιέχει νερό. Όταν υπάρχει περίσσεια ενέργειας, προκειμένου να μην απορριφθεί στο περιβάλλον, αποθηκεύεται ώστε να χρησιμοποιηθεί όταν είναι απαραίτητο. Διακρίνονται δύο περιπτώσεις για την μελέτη της δεξαμενής αποθήκευσης. Μπορεί να θεωρηθεί πως η δεξαμενή βρίσκεται στην ίδια θερμοκρασία στο σύνολό της και συνεπώς δεν υπάρχει διαστρωμάτωση. Διαφορετικά θεωρείται πως υπάρχει διαστρωμάτωση του υγρού στη δεξαμενή και συνεπώς υπάρχει θερμοκρασιακή διαφορά κατά μήκος της ροής. Η ύπαρξη διαστρωμάτωσης επηρεάζει τον βαθμό απόδοσης του συστήματος και για αυτό μοντελοποιούνται και οι δύο περιπτώσεις.

Δοχείο χωρίς διαστρωμάτωση

Η πιο απλή περίπτωση είναι να θεωρηθεί πως η δεξαμενή βρίσκεται σε μία ενιαία θερμοκρασία. Το θερμικό φορτίο που αποθηκεύεται ανά μονάδα χρόνου ισούται με τη διαφορά ανάμεσα σε αυτό που έρχεται από τον συλλέκτη μέσω του φέροντος ρευστού και αυτού που πηγαίνει στην κατανάλωση μέσω του ρευστού χρήσης μείον τις απώλειες του αποθηκευτικού δοχείου. Επίσης, σχετίζεται με την αλλαγή της θερμοκρασίας του ρευστού εντός του δοχείου σύμφωνα με τη σχέση

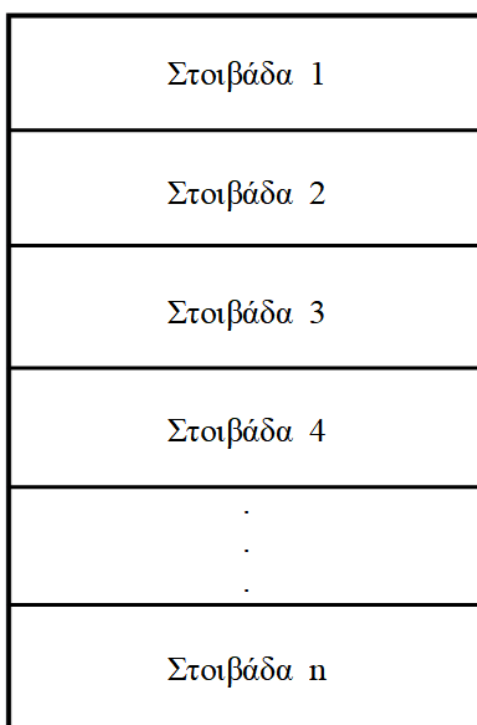
$$\dot{Q}_{st}(t) = m_{st}C \frac{dT_{st}}{dt} = \rho_{st}V_{st}C \frac{dT_{st}}{dt} = \dot{Q}_u - \dot{Q}_L - U_{st}A_{st}(T_{st} - T_a) \quad (2.64)$$

όπου $\dot{Q}_{st}(t)$ η ισχύς του αποθηκευμένου φορτίου, T_{st} η θερμοκρασία του ρευστού αποθήκευσης, m_{st} και C η μάζα και θερμοχωρητικότητα του ρευστού αποθήκευσης αντίστοιχα.

Δοχείο με διαστρωμάτωση

Στην περίπτωση που θεωρηθεί ότι υπάρχει διαστρωμάτωση της θερμοκρασίας στη δεξαμενή αποθήκευσης, τότε η μοντελοποίηση της δεξαμενής είναι πιο πολύπλοκη. Στη συνέχεια παρουσιάζεται το μοντέλο που χρησιμοποιήθηκε στην παρούσα μεταπτυχιακή εργασία.

Έστω δοχείο αποθήκευσης υγρού συνολικού ύψους L με n στοιβάδες ύψους Δx ($L = n \Delta x$) με διαφορετικές θερμοκρασίες T_m , όπου $m=1,2,\dots,n$ από την κορυφή προς τη βάση του δοχείου όπως φαίνεται στο Εικόνα 2.14.



Εικόνα 2.14 Δεξαμενή αποθήκευσης με διαστρωμάτωση

Από τον συλλέκτη εξέρχεται υγρό θερμοκρασίας $T_{f,out}$ το οποίο εισέρχεται στην κορυφή του δοχείου (στιβάδα 1). Εάν $T_{f,out} > T_1$ τότε η εσωτερική ενέργεια της στιβάδας 1 αυξάνει κατά το ποσό

$$\dot{m}_\sigma C (T_{f,out} - T_1) dt$$

Αντίθετα, στην περίπτωση όπου $T_{f,out} < T_1$, τότε το εισερχόμενο από τον συλλέκτη ρευστό θα βυθιστεί προς τις χαμηλότερες στιβάδες λόγω διαφοράς πυκνότητας μέχρι $T_{m-1} > T_{f,out} > T_m$. Τότε η εσωτερική ενέργεια της ζώνης m θα αυξηθεί κατά το ποσό

$$\dot{m}_\sigma C (T_{f,out} - T_m) dt$$

Επίσης, οι συνθήκες στις προηγούμενες στιβάδες ($1, 2, 3, \dots, m-1$) δεν επηρεάζονται.

Αντίθετα η εσωτερική ενέργεια στις κάτω στιβάδες ($m+1, m+2, \dots, n$) αλλάζει καθώς η μάζα του ρευστού που εισέρχεται στην στιβάδα m μετατοπίζει την ίδια ποσότητα μάζας θερμοκρασίας T_m στην στιβάδα $m+1$ κ.τ.λ., μέχρι τη στιβάδα n με τις εσωτερικές ενέργειες να αυξάνουν κατά το ποσό

$$\dot{m}_\sigma C(T_m - T_{m+1})dt$$

Η ίδια διαδικασία αλλά με αντίθετη φορά επαναλαμβάνεται για το ρευστό το οποίο επιστρέφει από την κατανάλωση με θερμοκρασία $T_{L,out}$.

Εάν $T_{L,out} < T_N$ τότε η εσωτερική ενέργεια της στιβάδας n θα αλλάξει κατά το ποσό

$$\dot{m}_L C(T_{L,out} - T_n)dt$$

Εάν $T_{L,out} > T_N$ τότε το εισερχόμενο ρευστό ανυψώνεται λόγω διαφοράς πυκνότητας προς τις επάνω στιβάδες μέχρι $T_{m-1} < T_{L,out} < T_m$ και τότε η εσωτερική ενέργεια της ζώνης m μειώνεται κατά το ποσό

$$\dot{m}_L C(T_{L,out} - T_m)dt$$

Οι συνθήκες στις κάτω στιβάδες ($m, m+1, \dots, n$) παραμένουν ως έχουν. Αντίθετα, η εσωτερική ενέργεια στις επάνω στιβάδες ($m-1, m-2, \dots, 1$) μεταβάλλεται καθώς η μάζα του ρευστού που εισέρχεται στη στιβάδα m μετατοπίζει την ίδια ποσότητα μάζας θερμοκρασίας T_m στη στιβάδα $m+1$ κ.τ.λ., μέχρι τη πρώτη στιβάδα με τις εσωτερικές ενέργειες να μειώνονται κατά το ποσό

$$\dot{m}_L C(T_m - T_{m+1})dt$$

Όπως είναι αναμενόμενο απαραίτητος είναι ο υπολογισμός των θερμοκρασιών κάθε στιβάδας. Αρχικά πρέπει να πραγματοποιηθεί θερμικό ισοζύγιο σε μία στιβάδα.

Για μία τυχαία στιβάδα m θα ισχύει :

Ρυθμός αλλαγής εσωτερικής ενέργειας στοιβάδας m =

= ρυθμό αλλαγής ενθαλπίας λόγω πιθανής εισόδου ρευστού από εναλλάκτη +

+ ρυθμό αλλαγής ενθαλπίας λόγω πιθανής εισόδου ρευστού από στοιβάδα ($m-1$) -

- ρυθμό αλλαγής ενθαλπίας λόγω πιθανής εξόδου ρευστού προς στοιβάδα ($m+1$) +

- + ρυθμό αλλαγής ενθαλπίας λόγω πιθανής εισόδου ρευστού από επιστροφή φορτίου +
- + ρυθμό αλλαγής ενθαλπίας λόγω πιθανής εισόδου ρευστού από στοιβάδα $(m+1)$ –
- ρυθμό αλλαγής ενθαλπίας λόγω πιθανής εξόδου ρευστού προς στοιβάδα $(m-1)$ –
- ρυθμό θερμικών απωλειών προς το περιβάλλον.

Διαφορετικά το παραπάνω ισοζύγιο μπορεί να γραφεί στη μορφή

$$\begin{aligned}
 (M_m C)_{st} \frac{dT_m}{dt} = & F_m \dot{m}_\Sigma C T_{f,out} + \dot{m}_\Sigma C T_{m-1} \sum_{l=1}^{m-1} F_l - \dot{m}_\Sigma C T_m \sum_{l=1}^m F_l - \\
 & - \dot{m}_\Sigma C T_m \sum_{l=1}^m F_l + G_m \dot{m}_L C T_{l,out} + \dot{m}_L C T_{m+1} \sum_{l=m+1}^n G_l - \\
 & - \dot{m}_L C T_m \sum_{l=m}^n G_l - U_{st} A_m (T_m - T_a)
 \end{aligned} \quad (2.65)$$

όπου

$F_m=1$ εάν $T_{m-1} > T_{f,out} > T_m$, διαφορετικά $F_m=0$

και

$G_m=1$ εάν $T_{m+1} < T_{l,out} < T_m$ διαφορετικά $G_m=0$

Τότε η θερμοκρασία κάθε στιβάδας προκύπτει από τη σχέση:

$$\begin{aligned}
 T_m(t + \Delta t) = & T_m(t) + \frac{\dot{m}_\Sigma \Delta t}{M_m} \left[F_m [T_{f,out}(t) - T_m(t)] + [T_{m-1}(t) - T_m(t)] \sum_{l=1}^{m-1} F_l \right] + \\
 & + \frac{\dot{m}_L \Delta t}{M_m} \left[G_m [T_{l,out}(t) - T_m(t)] + [T_{m+1}(t) - T_m(t)] \sum_{l=m+1}^n G_l \right] - \\
 & - \frac{U_{st} A_m \Delta t}{M_m} [T_m(t) - T_a(t)]
 \end{aligned} \quad (2.66)$$

Η εξίσωση 2.66 αντιπροσωπεύει n αλγεβρικές εξισώσεις οι οποίες λύνονται για κάθε χρονικό βήμα και προκύπτει η θερμοκρασία κάθε στιβάδας για τη δεξαμενή αποθήκευσης.

Απαραίτητο είναι το χρονικό βήμα Δt να είναι αρκετά μικρό ώστε η ενέργεια η οποία προστίθεται σε κάθε στιβάδα να είναι μικρότερη από τη θερμοχωρητικότητα της στιβάδας. Η ενέργεια που προστίθεται από τον συλλέκτη είναι:

$$\dot{m}_{\Sigma} c [T_{m+1}(t) - T_m(t)] \Delta t \leq (M_m C)_{st} [T_m(t + \Delta t) - T_m(t)] \quad (2.67)$$

εάν

$$\begin{aligned} [T_m(t + \Delta t) - T_m(t)] &< [T_{m+1}(t) - T_m(t)] \Rightarrow \\ \Rightarrow \Delta t &\leq \frac{(M_m C)_{st}}{\dot{m}_{\Sigma} C} = \frac{(MC)_{st}}{n \dot{m}_{\Sigma} C} \end{aligned} \quad (2.68)$$

Οι παραπάνω εξισώσεις χρησιμοποιούνται για την μελέτη και περιγραφή της διαστρωμάτωσης και έχουν συμπεριληφθεί στο κώδικα ο οποίος παρουσιάζεται ολοκληρωμένος στο παράρτημα.

2.2.6 Δυναμικό και στατικό φορτίο κατανάλωσης

Το φορτίο ZNX συνήθως είναι μεταβλητό και η παροχή νερού είναι διαφορετική για κάθε ώρα της ημέρας. Συνήθως το προφίλ του φορτίου είναι ίδιο για κάθε μέρα οπότε τα στοιχεία είναι για 24 ώρες.

Επειδή στο σύστημα ZNX τοποθετείτε βάνα ανάμιξης, το θερμικό φορτίο που μεταφέρεται από το δοχείο αποθήκευσης $Q_L(t)$ δίνεται από:

$$Q_L(t) = m_L(t) C (T_{st}(t) - T_{sup}) \quad (2.69)$$

με

$$m_L(t) = \frac{Q_{load}}{C (T_{st}(t) - T_{sup})} \quad (2.70)$$

Κεφάλαιο 3. Ανάλυση λειτουργίας αλγορίθμου

3.1 Λογισμικό Matlab

Το Matlab (από τα αρχικά Matrix Laboratory) είναι ένα αριθμητικό υπολογιστικό και προγραμματιστικό περιβάλλον τέταρτης γενιάς. Αποτελεί μία εμπορική γλώσσα προγραμματισμού που αναπτύχθηκε από την εταιρία MathWorks. Είναι διαδεδομένο επειδή επιτρέπει εύκολους χειρισμούς μήτρας, σχεδίαση λειτουργιών και δεδομένων, υλοποίηση αλγορίθμων, δημιουργία διεπαφών χρήστη και διασύνδεση με προγράμματα γραμμένα σε άλλες γλώσσες, όπως C, C ++, C #, Java, Fortran και Python. Αξίζει να αναφερθεί ότι συνεργάζεται εξαιρετικά και με λογισμικά από λογιστικά φύλλα όπως το Microsoft Excel, λειτουργία η οποία εφαρμόστηκε σε μεγάλο βαθμό στην παρούσα διπλωματική.

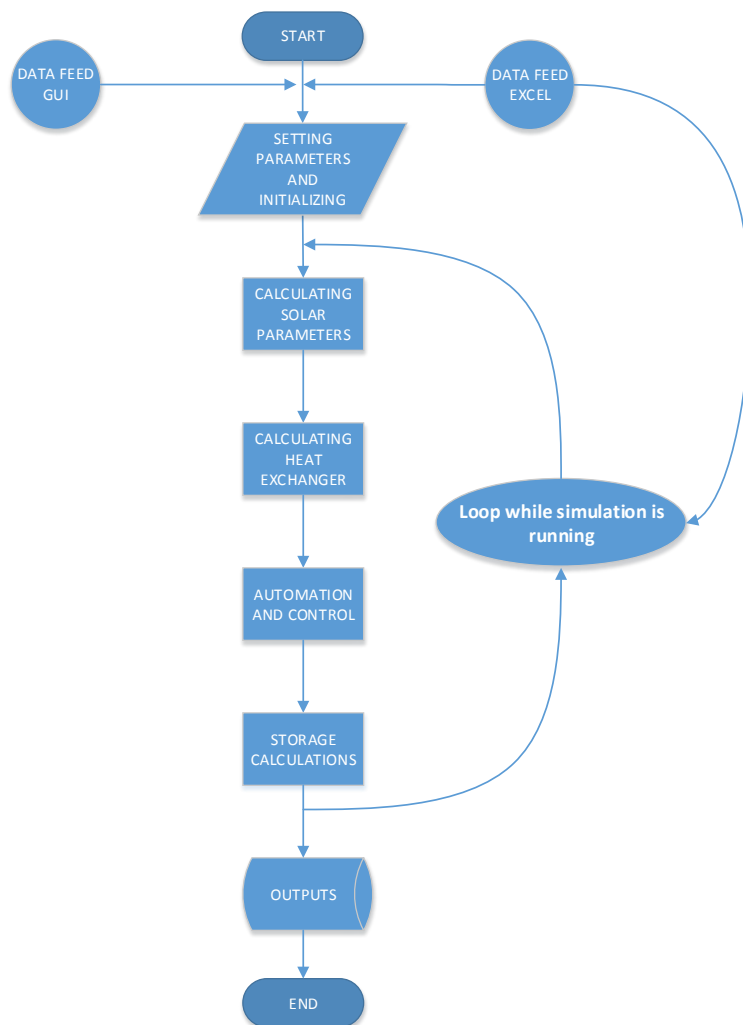
Παρόλο που το Matlab προορίζεται κυρίως για αριθμητική υπολογιστική, τα τελευταία χρόνια έχουν αναπτυχθεί αρκετές βιβλιοθήκες που αφορούν διάφορους τομείς και επιστήμες. Οι πιο γνωστές και διαδεδομένες είναι βιβλιοθήκες όπως η MuPAD που επιτρέπει την πρόσβαση σε συμβολικές υπολογιστικές ικανότητες, το Simulink που προσθέτει γραφικές προσομοιώσεις πολλαπλών τομέων και σχεδιασμό με βάση το μοντέλο για δυναμικά και ενσωματωμένα συστήματα, καθώς και το GUI Layout Toolbox το οποίο θα χρησιμοποιήσουμε για τη δημιουργία γραφικού περιβάλλοντος για τον αλγόριθμο μας [19].

Αξίζει να αναφερθεί πως από το 2017, η Matlab έχει πάνω από 2 εκατομμύρια χρήστες σε ολόκληρη τη βιομηχανία και τον ακαδημαϊκό κόσμο. Οι χρήστες του Matlab προέρχονται από διάφορα περιβάλλοντα της μηχανικής, της επιστήμης και της οικονομίας[19].

3.2 Δομή του αλγορίθμου

Ο κώδικας είναι γραμμένος σε Matlab και έχει δημιουργηθεί ώστε να γίνονται οι υπολογισμοί με χρονικό βήμα 1 δευτερόλεπτο. Αρχικά, πρέπει να δοθούν ορισμένα δεδομένα (π.χ. επιφάνεια συλλέκτη, ηλιακή ακτινοβολία κλπ.) τα οποία προέρχονται είτε από το χρήστη, είτε από το Excel, είτε από το GUI και στη συνέχεια υπολογίζονται βασικά στοιχεία όπως η θερμοκρασία εξόδου από τον συλλέκτη, η θερμοκρασία της δεξαμενής καθώς και τα αντίστοιχα ποσά ενέργειας που αποθηκεύονται και

καταναλώνονται. Στην Εικόνα 3.1 παρουσιάζονται επιγραμματικά οι βασικές λειτουργίες του αλγορίθμου, οι οποίες θα αναλυθούν στη συνέχεια του κεφαλαίου.



Εικόνα 3.1 Διάγραμμα ροής γενικής λειτουργίας του αλγορίθμου.

3.3 Μεταβλητές του αλγορίθμου

Σε αυτή την ενότητα παρατίθενται πίνακες που περιέχουν τις μεταβλητές και μία σύντομη περιγραφή για το τι αντιπροσωπεύει η κάθε μία στον αλγόριθμο. Οι μεταβλητές αναλύονται πρώτες διότι είναι πολλές σε αριθμό και χωρίς την επεξήγησή τους η κατανόηση του αλγορίθμου καθίσταται σχεδόν αδύνατη.

Πίνακας 3.1 Μεταβλητές ελέγχου διακοπών.

irv	Διακόπτης ελέγχου του τρόπου υπολογισμού της ακτινοβολίας, με την τιμή ένα/μηδέν επιλέγεται ο πρώτος ή ο δεύτερος τρόπος αντίστοιχα.
idhw	Διακόπτης ZNX, με την τιμή ένα/μηδέν ενεργοποιείται/ απενεργοποιείται το ZNX.

sload	Διακόπτης δυναμικού φορτίου, με την τιμή ένα/μηδέν ενεργοποιείται/ απενεργοποιείται το δυναμικό φορτίο.
iheat	Διακόπτης εναλλάκτη, με την τιμή ένα/μηδέν ενεργοποιείται/ απενεργοποιείται ο εναλλάκτης.
itypest	Διακόπτης διαστρωμάτωσης, με την τιμή ένα/μηδέν ενεργοποιείται/ απενεργοποιείται η διαστρωμάτωση της δεξαμενής.
idd	Διακόπτης βαθμομερών, με την τιμή ένα/μηδέν ενεργοποιείται/ απενεργοποιείται η λειτουργία των βαθμομερών.

Πίνακας 3.2 Μεταβλητές του χρόνου και τόπου της προσομοίωσης.

dday	Αναφέρεται στην τρέχουσα ημέρα της προσομοίωσης.
day	Αναφέρεται στην αρχική ημέρα της προσομοίωσης.
month	Αναφέρεται στο μήνα έναρξης της προσομοίωσης.
range	Αναφέρεται στη διάρκεια της προσομοίωσης.
city	Η πόλη της προσομοίωσης.
psist	Αναφέρεται στον μεσημβρινό του Greenwich.
psilocal	Το τοπικό γεωγραφικό πλάτος.
phi	Το τοπικό γεωγραφικό μήκος.
timestep	Αναφέρεται στο βήμα της προσομοίωσης και ισούται με 1/86400.

Πίνακας 3.3 Μεταβλητές κτιρίου, δεξαμενής αποθήκευσης, δικτύου ύδρευσης.

uresidence	Ο συντελεστής θερμοπερατότητας ενός κτηρίου σε $W/m^2 K$.
aresidence	Το εμβαδό ενός κτηρίου σε m^2 .
ust	Ο συντελεστής θερμικής αγωγιμότητας της δεξαμενής σε $W/m^2 K$.
ast	Το εμβαδό της δεξαμενής σε m^2 .
mst	Η μάζα του φέροντος ρευστού που περιέχει η δεξαμενή σε m^3 .
vstorage	Ο όγκος της δεξαμενής σε m^3 .
rinidia	Η εσωτερική διάμετρος της δεξαμενής σε m .
rho	Η πυκνότητα του νερού σε kg/m^3 .
i_diff	Ο συντελεστής θερμικής διάχυσης του χώματος της περιοχής.
diff	Δείκτης του GUI για την επιλογή της διάχυσης.
i_rr	Ο συντελεστής ακτινοβολίας της περιοχής.
rr	Δείκτης του GUI για την επιλογή της ακτινοβολίας.

d_depth	Το βάθος που βρίσκεται το δίκτυο ύδρευσης, τυπικά 0,5 m.
soil_day	Η ημέρα όπου το έδαφος έχει τη χαμηλότερη θερμοκρασία, τυπικά 30.

Πίνακας 3.4 Μεταβλητές συλλέκτη και εναλλάκτη.

a	Η επιφάνεια του συλλέκτη.
b	Τιμή b του συλλέκτη.
ta_eff	Συνολικό κλάσμα απορρόφησης του συλλέκτη.
fr	Το κλάσμα της ωφέλιμης ενέργειας ως προς τη θερμική ενέργεια του συλλέκτη.
fr_h	Το μέτρο απόδοσης του εναλλάκτη Fr'/Fr .
c_v	Η πυκνότητα του φέροντος ρευστού, αέρα ή νερού.
c	Δείκτης του GUI για την επιλογή της του φέροντος ρευστού.
u	Ο συντελεστής θερμικής αγωγιμότητας του συλλέκτη σε $W/m^2 K$.
eps	Ο συντελεστής ϵ του εναλλάκτη.
gq	Ο συντελεστής γ του εναλλάκτη.

Πίνακας 3.5 Μεταβλητές θερμικών φορτίων, ακτινοβολίας και παροχών.

qs	Η προσπίπτουσα ακτινοβολία στο συλλέκτη.
qsmax	Η μέγιστη προσπίπτουσα ακτινοβολία.
qsref	Η προσπίπτουσα ακτινοβολία της περιοχής.
qu	Η διαθέσιμη προσπίπτουσα ηλιακή ενέργεια.
ql	Το θερμικό φορτίο.
qload	Το θερμικό φορτίο για ZNX.
qst	Το θερμικό φορτίο αποθηκευμένο στη δεξαμενή.
ml	Η μαζική παροχή του φέροντος ρευστού προς την κατανάλωση σε kg/sec.
mload	Η μαζική παροχή του φέροντος ρευστού προς κατανάλωση για ZNX σε kg/sec.
mc	Η μαζική παροχή του φέροντος ρευστού προς το συλλέκτη σε kg/sec.
ml_u	Η μαζική παροχή του φέροντος ρευστού προς την κατανάλωση για δυναμικό φορτίο σε kg/sec.
dc	Ο διακόπτης για την παροχή της αντλίας του συλλέκτη.
dl	Ο διακόπτης για την παροχή της αντλίας της δεξαμενής.

kt	Ο συντελεστής αιθριότητας.
dd	Η τιμή των βαθμοημερών.
ho	Η ηλιακή ακτινοβολία στο όριο της ατμόσφαιρας.
h	Η τοπική ημισφαιρική ολική ακτινοβολία στο οριζόντιο επίπεδο.
hb	Η τοπική ημισφαιρική ολική ακτινοβολία κατά εκτίμηση.
io	Διόρθωση ηλιακής σταθεράς στη διάρκεια του χρόνου.
isc	Η ηλιακή σταθερά σε W/m^2 .
n	Ο βαθμός απόδοσης του συλλέκτη.

Πίνακας 3.6 Μεταβλητές γωνιών ηλίου – συλλέκτη.

beta	Η γωνία κλίσης συλλέκτη β .
gmr	Η αζιμούθια γωνία συλλέκτη γ .
delta	Η γωνία απόκλισης δ .
wss	Η γωνία ανατολής - δύσης ω_{SS} .
omega	Η ωριαία ηλιακή γωνία ω .
as	Η γωνία ηλιακού ύψους a_s .
gms	Η ηλιακή αζιμούθια γωνία γ_s .
theta	Η γωνία πρόσπτωσης θ .

Πίνακας 3.7 Μεταβλητές ηλιακού και πραγματικού χρόνου.

bq	Συντελεστής διόρθωσης τροχιάς.
etraj	Διόρθωση ώρας λόγω τροχιάς.
tlocal	Η τοπική-πραγματική ώρα.
tsolar	Η ηλιακή ώρα.
correction	Διόρθωση θερινής/χειμερινής ώρας.
day_night	Καθορίζει αν είναι ημέρα ή νύχτα.
sunshine	Η διάρκεια της ηλιοφάνειας.

Πίνακας 3.8 Μεταβλητές θερμοκρασιών του συστήματος.

tam	Η θερμοκρασία της ατμόσφαιρας για την συγκεκριμένη ημέρα σε K.
dta	Η διακύμανση της θερμοκρασίας στη διάρκεια της ημέρας σε K.
ta	Η θερμοκρασία του συλλέκτη σε K.

t _{sup}	Η θερμοκρασία του νερό από το δίκτυο παροχής σε Κ.
t _{fin}	Η θερμοκρασία εισόδου του ρευστού στον συλλέκτη σε Κ.
t _{lin}	Η θερμοκρασία εισόδου του ρευστού προς την κατανάλωση σε Κ.
d _{tl}	Η διαφορά της θερμοκρασίας εισόδου-εξόδου σε Κ.
t _{st}	Η θερμοκρασία του ρευστού αποθήκευσης σε Κ.
t _{stmax}	Η μέγιστη επιτρεπτή θερμοκρασία σε Κ.
t _{fout}	Η θερμοκρασία εξόδου του ρευστού από το συλλέκτη σε Κ.
t _{stag}	Η θερμοκρασία της πλάκας απορρόφησης για συνθήκες μηδενικής μαζικής παροχής σε Κ.
t _{lout}	Η θερμοκρασία εξόδου του ρευστού από την κατανάλωση σε Κ.

Πίνακας 3.9 Μεταβλητές λειτουργίας διαστρωμάτωσης.

layers	Ο αριθμός των επιπέδων της διαστρωματοποιημένης δεξαμενής.
t _{strat}	Η θερμοκρασία της δεξαμενής για συγκεκριμένο επίπεδο.
gm	Διακόπτης λειτουργίας της διαστρωμάτωσης.
fm	Διακόπτης λειτουργίας της διαστρωμάτωσης.
sum _{gm}	Αθροιστής.
sum _{fm}	Αθροιστής.
c _{ld}	Θερμοκρασιακή διαφορά μεταξύ των στρωμάτων.
l _{dd}	Θερμοκρασιακή διαφορά μεταξύ των στρωμάτων.
t _{sttmp}	Η θερμοκρασία της δεξαμενής σε Κ.
f _{rlcst}	Παροχή συλλέκτη.
f _{rldst}	Παροχή φορτίου.

Πίνακας 3.10 Μεταβλητές διαχείρισης και εκτύπωσης δεδομένων.

iupdate	Αθροιστικός πίνακας ημερών ανά μήνα του έτους.
sun _{duration}	Πίνακας πραγματικής ηλιοφάνειας ανά μήνα.
hr	Πίνακας ωρών ανά ημέρα.
imonth	Δείκτης για την ενημέρωση μεταβλητών ανά μήνα από Excel.
index	Δείκτης της πόλης για ενημέρωση μεταβλητών από Excel.
indexlatti	Δείκτης γεωγραφικού μήκους για ενημέρωση μεταβλητών από Excel.

indexmeri	Δείκτης γεωγραφικού πλάτους για ενημέρωση μεταβλητών από Excel.
vec	Πίνακας εκτύπωσης αποτελεσμάτων σε Excel.
cell	Δείκτης για εκτύπωση αποτελεσμάτων σε Excel.
TST	Μεταβλητή για γραφικό περιβάλλον της θερμοκρασίας της δεξαμενής στο GUI.
QU	Μεταβλητή για γραφικό περιβάλλον της ωφέλιμης θερμικής ενέργειας στο GUI.
N_A	Μεταβλητή για γραφικό περιβάλλον του βαθμού απόδοσης στο GUI.

Όπως θα δούμε παρακάτω, η αρχικοποίηση και ενημέρωση των μεταβλητών γίνεται είτε από το γραφικό περιβάλλον είτε από το Excel, είτε από τον ίδιο τον αλγόριθμο. Για το λόγο αυτό, σε αυτή την παράγραφο, καλό θα ήταν να τις διαχωρίσουμε.

Οι μεταβλητές οι οποίες ενημερώνονται από το γραφικό περιβάλλον είναι:

- Οι διακόπτες irv, idhw, sload, iheat, itypest, idd.
- Οι μεταβλητές που αφορούν τα στοιχεία της προσομοίωσης dday, month, range, city.
- Οι μεταβλητές που αφορούν το κτήριο uresidence, aresidence.
- Οι μεταβλητές που αφορούν τη δεξαμενή αποθήκευσης ast,ust,mst.
- Οι μεταβλητές που αφορούν το συλλέκτη a, u, ta_eff, fr, c_v.
- Οι μεταβλητές που αφορούν τα θερμικά φορτία και τις παροχές ml_u, ml, ql.
- Η μεταβλητή από τις θερμοκρασίες του συστήματος tmin.
- Τον συντελεστή θερμικής διάχυσης, το βάθος και η ημέρα της χαμηλότερης θερμοκρασίας που αφορούν το νερό παροχής δικτύου ύδρευσης i_diff, d_depth, soil_day αντίστοιχα.

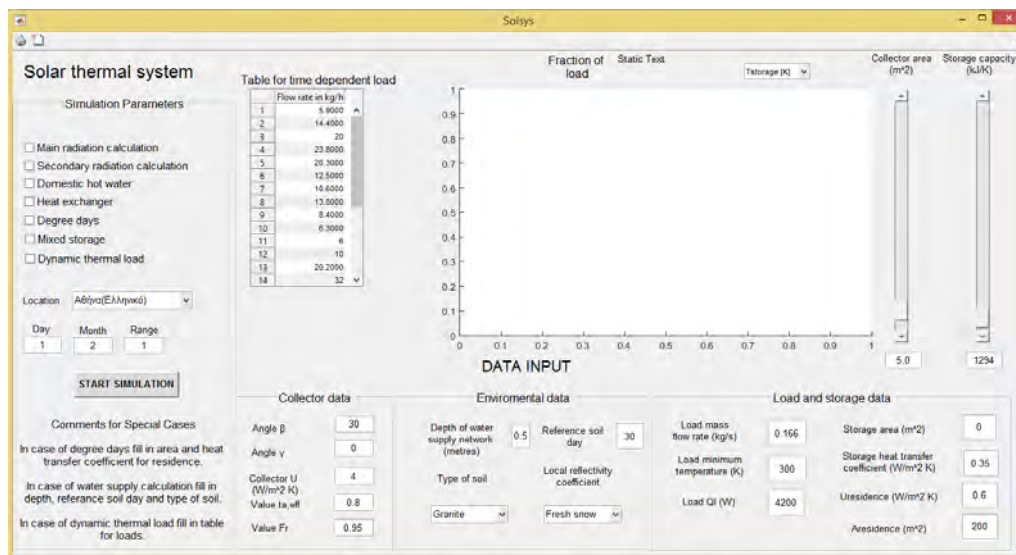
Οι μεταβλητές που ενημερώνονται από τα δεδομένα του Excel είναι:

- Οι μεταβλητές που αφορούν τη μέση θερμοκρασία ημέρας και τη θερμοκρασία παροχής δικτύου, tam και tsup.
- Η μεταβλητή που αφορά τις βαθμοημέρες dd.

- Οι μεταβλητές που αφορούν τη μέγιστη προσπίπτουσα ακτινοβολία και το συντελεστή αιθριότητας, qs_{ref} και kt .
- Το γεωγραφικό μήκος και πλάτος $psilocal, phi$.

3.4 Γραφικό περιβάλλον και λειτουργίες

Η αλγόριθμος όπως αναφέρθηκε νωρίτερα λειτουργεί μέσω ενός γραφικού περιβάλλοντος διεπαφής χρήστη (GUI), σε αυτή την παράγραφο περιγράφονται οι επιλογές που έχει ο χρήστης του αλγορίθμου, όπως φαίνονται στην Εικόνα 3.2.



Εικόνα 3.2 Απεικόνιση του γραφικού παραθύρου διεπαφής χρήστη.

Το γραφικό περιβάλλον είναι χωρισμένο στους έξι παρακάτω θεματικούς τομείς:

- Simulation Parameters
- Collector Data
- Environmental Data
- Load and Storage Data
- Plots
- Storage Capacity and Collector Area

Simulation Parameters

Στον τομέα αυτό ο χρήστης μπορεί να επιλέξει τις απαραίτητες παραμέτρους για την προσομοίωση, αυτές είναι:

- Ο ημερολογιακός χρόνος και η διάρκεια της προσομοίωσης.
- Η τοποθεσία της περιοχής υπό μελέτη.

- Τον τρόπο υπολογισμού της ακτινοβολίας.
- Αν θα προσομοιωθεί ΗΘΣ ζεστού νερού χρήσης ή θέρμανσης.
- Αν το φορτίο θα είναι σταθερό ή χρονομεταβαλλόμενο κατά την εκτέλεση του αλγορίθμου.
- Αν η δεξαμενή αποθήκευσης θα θεωρηθεί ομοιογενής η διαστρώματοποιημένη.
- Αν υπάρχει εναλλάκτης θερμότητας στην προσομοίωση.
- Αν τα θερμικά φορτία θα είναι με βάση τις βαθμοημέρες.

Collector Data

Ο χρήστης εδώ επιλέγει διάφορα στοιχεία που αφορούν τον συλλέκτη που μελετά, οι μεταβλητές των οποίων οι τιμές που μπορούν να τροποποιηθούν είναι:

- Η γωνία β του συλλέκτη
- Η γωνία γ του συλλέκτη.
- Η τιμή u του συλλέκτη.
- Το συντελεστή $(\tau\alpha)_{eff}$.
- Το κλάσμα της ωφέλιμης προς τη θερμική ενέργεια του συλλέκτη Fr .

Environmental Data

Στην περιοχή αυτή μπορούν να εισαχθούν διάφορα περιβαλλοντολογικά στοιχεία όπως:

- Το βάθος του νερού παροχής δικτύου.
- Η χαρακτηριστική ημέρα με τη μικρότερη θερμοκρασία του εδάφους.
- Ο συντελεστής μεταφοράς θερμότητας του χώματος.
- Ο συντελεστής αντανάκλασης ακτινοβολίας του περιβάλλοντος.
- Το εμβαδό ενός κτηρίου υπό μελέτη.
- Το συντελεστή μετάδοσης θερμότητας ενός κτιρίου υπό μελέτη.

Load and Storage Data

Οι επιλογές του τομέα αυτού αφορούν στοιχεία του φορτίου και της δεξαμενής αποθήκευσης, οι τιμές που μπορεί να επηρεάσει ο χρήστης είναι οι εξής:

- Το φορτίο ζήτησης.
- Η μαζική παροχή του φορτίου.

- Η ελάχιστη θερμοκρασία του φορτίου.
- Το μέγεθος της δεξαμενής αποθήκευσης.
- Ο συντελεστής μετάδοσης θερμότητας της δεξαμενής.

Επίσης σε αυτή την ενότητα ανήκει και ο πίνακας «Table for time dependent load» ο οποίος χρησιμοποιείτε για την περίπτωση όπου το φορτίο δεν είναι σταθερό.

Plots

Ο τομέας αυτός περιλαμβάνει διάφορα γραφήματα από τιμές που προκύπτουν κατά την διάρκεια της εκτέλεσης του αλγορίθμου. Οι μεταβλητές οι οποίες εκτυπώνονται είναι η θερμοκρασία αποθήκευσης από την επιλογή Tstorage [K], η χρήσιμη θερμική ενέργεια από την επιλογή Quseful [kW] και ο βαθμός απόδοσης από την επιλογή Efficiency [%]. Όλες οι παραπάνω μεταβλητές εμφανίζονται συνάρτηση του χρόνου εκτέλεσης του αλγορίθμου, παραδείγματος χάρη ένα γράφημα της επιλογής Efficiency [%] θα μπορούσε να είναι ο βαθμός απόδοσης του ΗΘΣ κατά τη διάρκεια ενός μήνα. Τέλος, ένα ακόμα στοιχείο που περιλαμβάνει ο τομέας αυτός είναι η μεταβλητή fraction load, που εμφανίζει το ποσοστό κάλυψης των αναγκών του συστήματος.

Αποτελέσματα Excel

Αξίζει να αναφερθεί ότι ο τομέας των γραφημάτων δεν είναι ο μόνος τρόπος που παρέχει αποτελέσματα ο αλγόριθμός, ένα μεγάλο πλήθος μεταβλητών τυπώνεται σε φύλλο Excel με το πέρας της εκτέλεσης του προγράμματος. Ο χρήστης με αυτό τον τρόπο μπορεί να δημιουργήσει γραφήματα και άλλα δεδομένα σύμφωνα με το αντικείμενο της μελέτης του, χρησιμοποιώντας τις δυνατότητες του Excel. Οι μεταβλητές που εμφανίζονται είναι η τοπική ωρολογιακή ώρα, η θερμοκρασία δεξαμενής αποθήκευσης στις περιπτώσεις όπου είναι ή όχι διαστρωματοποιημένη, η θερμοκρασία εισόδου και εξόδου του φέροντος ρευστού στον συλλέκτη, η θερμοκρασία του νερού εισόδου και εξόδου κατά τη χρήση ZNX και το φορτίο της δεξαμενής αποθήκευσης και του συστήματος. Επίσης εμφανίζονται οι τιμές που λαμβάνουν οι διακόπτες των αυτοματισμών και η μαζική παροχή κατά την εκτέλεση, ο συνολικός βαθμός απόδοσης και μερικά οικονομικά στοιχεία.

Storage Capacity and Collector Area

Το γραφικό περιβάλλον, εκτός των τομέων που περιγράφηκαν νωρίτερα, περιλαμβάνει δύο ρυθμιστές για τις τιμές των μεταβλητών. Ο πρώτος αναφέρεται στην επιφάνεια του συλλέκτη και ο δεύτερος στην χωρητικότητα της δεξαμενής αποθήκευσης. Οι μεταβλητές αυτές αποτελούν πολύ συχνά αντικείμενο μελέτης για αυτό και διαφοροποιούνται από τις υπόλοιπες. Στη συνέχεια αφού έχουν καθοριστεί όλες οι παράμετροι σύμφωνα με τη μελέτη, ο χρήστης το μόνο που απομένει να κάνει είναι να πατήσει το κουμπί Start Simulation το οποίο ξεκινά την προσομοίωση.

Διαχωρισμός εισαγωγής δεδομένων

Έχουν αναφερθεί όλες οι δυνατές επιλογές του παραθύρου λειτουργίας, για το λόγο αυτό πρέπει να σημειωθούν οι περιπτώσεις των δεδομένων εισαγωγής που προκύπτουν από τη δομή και τον τρόπο επεξεργασίας τους στον αλγόριθμο.

Σε περίπτωση που επιλεγεί η λειτουργία των βαθμομερών ο χρήστης πρέπει να συμπληρώσει το εμβαδόν και το συντελεστή θερμοπερατότητας του κτιρίου στην καρτέλα Load and Storage Data. Η επιλογή αυτή δεν έχει νόημα σε περίπτωση που η προσομοίωση δεν χρησιμοποιεί τη μέθοδο των βαθμομερών.

Ακόμη, στην περίπτωση που ο χρήστης επιλέξει το χρονικά μεταβαλλόμενο φορτίο πρέπει να συμπληρώσει τον πίνακα «Table for time dependent load».

3.5 Ορισμός αλγοριθμικού χρόνου λειτουργίας

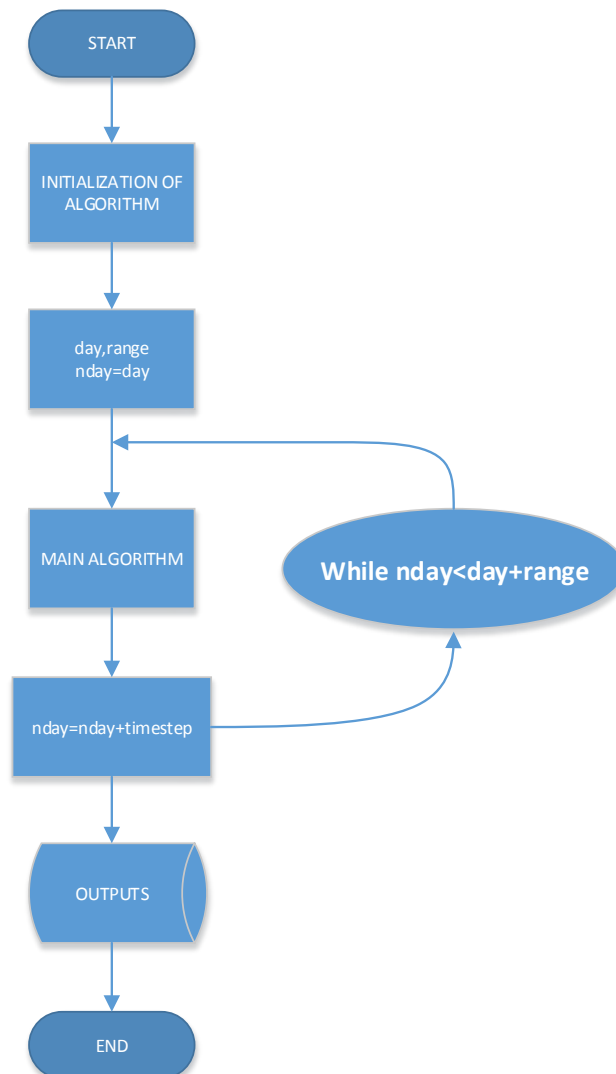
Εκτός από τον πραγματικό και τον ηλιακό χρόνο, που αναφέραμε νωρίτερα, στα πλαίσια της προσομοίωσης θα πρέπει να οριστεί και ο αλγοριθμικός χρόνος. Ο χρόνος αυτός θα χρησιμοποιηθεί από το πρόγραμμα ώστε να εκτελεστούν οι υπολογισμοί συνάρτηση του χρόνου.

Έστω ότι η διάρκεια μίας ημέρας είκοσι τεσσάρων ωρών ίση με τη μονάδα. Για τους υπολογισμούς ικανοποιητικό χρονικό βήμα θεωρήθηκε το βήμα ενός δευτερολέπτου, το οποίο υπολογίζεται αν διαιρέσουμε την διάρκεια της μίας ημέρας με το γινόμενο των ωρών επί τα δευτερόλεπτα τους, δηλαδή $1/(24*3600)=1/86400$ χρονικές μονάδες. Επομένως για να εκτελέσουμε τους την προσομοίωση ενός λεπτού ο αλγόριθμος θα τρέξει 60 φορές, μίας ώρας 3600 φορές και μίας ημέρας 86400 φορές.

Η χρήση του συγκεκριμένου χρονικού διαστήματος επιλέχθηκε γιατί διευκολύνει την κατανόηση του χρονικού βήματος αλλά και την εφαρμογή της

προσομοίωσης, αφού στην ουσία χωρίζει τις εικοσιτέσσερις ώρες της ημέρας σε 3600 δευτερόλεπτα, όπως ακριβώς γίνεται και στον πραγματικό κόσμο.

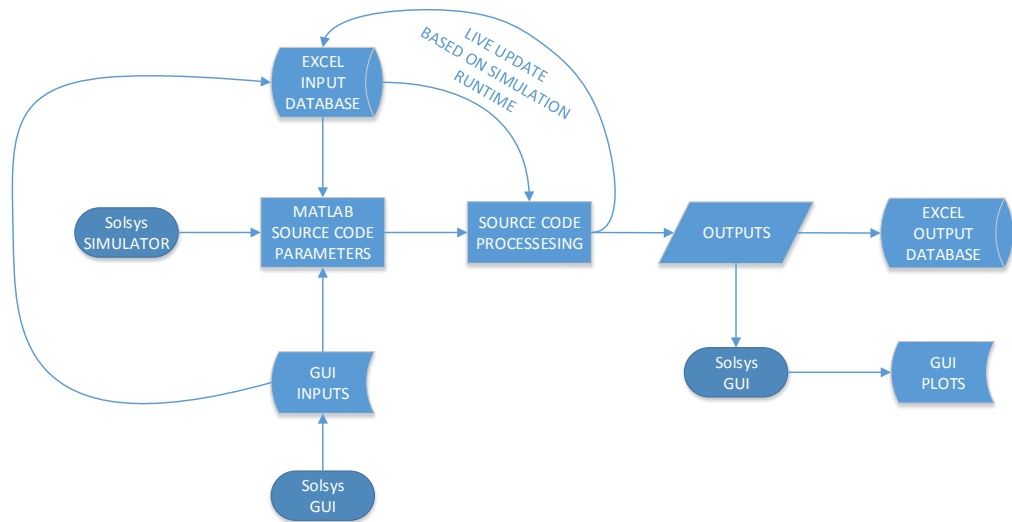
Ο τρόπος με τον οποίο λειτουργεί το χρονικό βήμα στην προσομοίωση γίνεται πιο κατανοητό βλέποντας το διάγραμμα ροής της Εικόνας 3.3. Χρησιμοποιούμε τις μεταβλητές `day` και `range`. Η μεταβλητή `day` δείχνει την ημέρα έναρξης της προσομοίωσης, ενώ η μεταβλητή `range` τη διάρκειά της προσομοίωσης. Στη συνέχεια η τιμή της πρώτης ημέρας αντιστοιχίζεται στην μεταβλητή `nday` και μέσω μίας επανάληψης αυξάνει με βήμα ενός δευτερολέπτου με τη μεταβλητή `timestep`, έως ότου φτάσει το άθροισμα των μεταβλητών `day` και `range` που υποδεικνύουν την τελευταία ημέρα της προσομοίωσης.



Εικόνα 3.3 Διάγραμμα χρονικού βήματος λειτουργίας του αλγορίθμου.

3.6 Αλληλεπίδραση Matlab Excel GUI

Η γενική μορφή του αλγορίθμου δίνεται από την Εικόνα 3.4:



Εικόνα 3.4 Διάγραμμα αλληλεπίδρασης Matlab, Excel και γραφικού περιβάλλοντος.

Από τη μία πλευρά έχουμε το βασικό αλγόριθμο Solsys Simulator για τους υπολογισμούς και από την άλλη τον αλγόριθμο Solsys GUI όπου διαχειρίζεται το γραφικό περιβάλλον. Ο χρήστης στο γραφικό περιβάλλον καθορίζει τον τρόπο λειτουργίας του αλγορίθμου και την πηγή των δεδομένων. Για ορισμένες μεταβλητές μπορεί η εισαγωγή να είναι χειροκίνητη ή να χρησιμοποιεί τιμές από το Excel.

Οι μεταβλητές επομένως ενημερώνονται από τον πηγαίο κώδικα και το γραφικό περιβάλλον, στη συνέχεια η βάση δεδομένων του Excel που περιέχει πίνακες του TEE τροφοδοτεί τον αλγόριθμο με πραγματικές τιμές. Όλες αυτές οι εισοδοι αντιστοιχούνται στις μεταβλητές του Solsys Simulator, το οποίο είναι υπεύθυνο για τους υπολογισμούς. Κατά τη διάρκεια των υπολογισμών, σε περίπτωση όπου οι τιμές των αλλάζουν με την πάροδο του χρόνου, ο αλγόριθμος ανατρέχει και πάλι στη βάση δεδομένων και ενημερώνεται.

Με την ολοκλήρωση των υπολογισμών το Solsys αλληλοεπιδρά με τον αλγόριθμο του γραφικού περιβάλλοντος για την δημιουργία των γραφημάτων και στη συνέχεια δημιουργεί ένα νέο αρχείο Excel που περιέχει όλα τα αποτελέσματα της εκτέλεσης, όπως αναφέραμε στο προηγούμενο κεφάλαιο.

3.7 Τροφοδοσία και ενημέρωση πραγματικών δεδομένων

Για την τροφοδοσία πραγματικών δεδομένων στον αλγόριθμο μέσω του Excel, αρχικά δημιουργούμε ένα αρχείο Excel. Στη συνέχεια αντιστοιχούμε κάθε ένα από τους πίνακες του κεφαλαίου 2.1.9 σε ένα φύλλο του, με συγκεκριμένη μορφή.

Για την περίπτωση της εισαγωγής κλιματικών δεδομένων τα οποία μεταβάλλονται κάθε μήνα χρησιμοποιούμε τη μορφή που φαίνεται στην Εικόνα 3.5, ενώ για την περίπτωση της εισαγωγής των τοπολογικών δεδομένων χρησιμοποιούμε τη μορφή της Εικόνας 3.8.

Twelve months from A to L

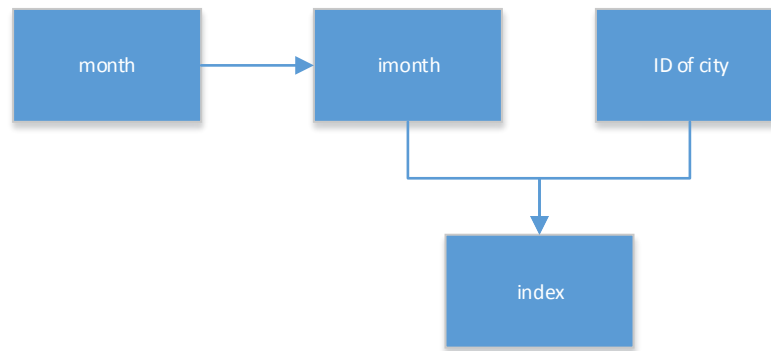
	A	B	C	D	E	F	...	L
1								
2								
3								
4								
...								
62								

Εικόνα 3.5 Απεικόνιση των πινάκων δεδομένων εισαγωγής

Προσπέλαση κλιματικών δεδομένων

Κάθε μία από τις πόλεις υπό μελέτη αντιστοιχίζεται σε έναν αριθμό από το 1 έως το 62, ενώ κάθε ένας από τους μήνες σε ένα γράμμα από το A έως το L. Επομένως χρειαζόμαστε ένα δείκτη ο οποίος αναφέρεται στο αναγνωριστικό της πόλης μέσω ενός αριθμού και στο μήνα μέσω ενός γράμματος.

Ο δείκτης index συνδυάζει τον αριθμό της πόλης και το μήνα αφού πρώτα τον αντιστοιχήσει σε γράμμα μέσω του πίνακα imonth. Στη συνέχεια ο αλγόριθμος ενημερώνεται μέσω της εντολής xlsread, όπως φαίνεται στην Εικόνα 3.6.



Εικόνα 3.6 Διάγραμμα αλληλεπίδρασης μεταβλητών προσπέλασης δεδομένων.

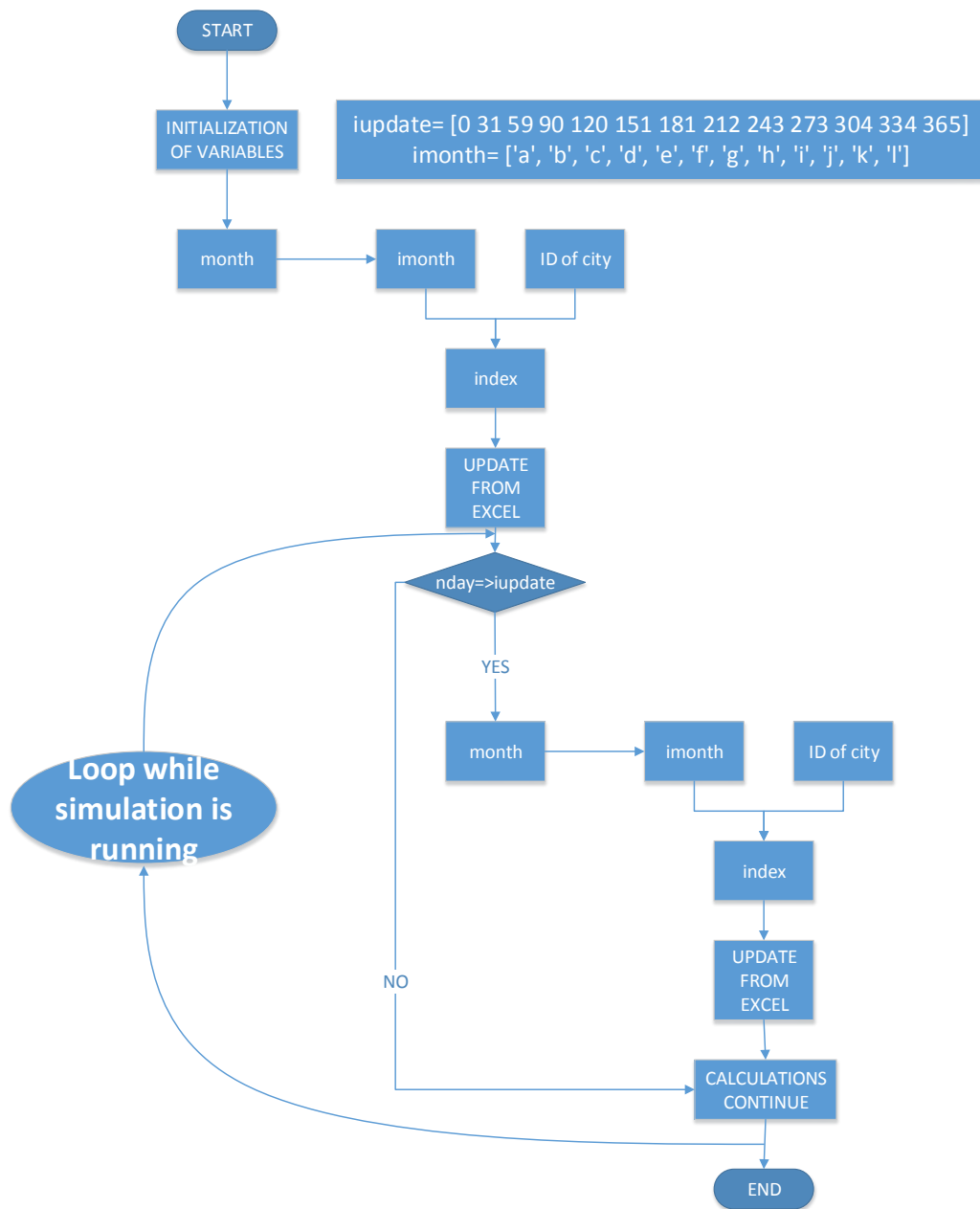
Για παράδειγμα, έστω ότι ο μήνας είναι ο Μάρτιος και η πόλη που προσομοιώνεται η Αθήνα.

- Ο δείκτης της πόλης θα πάρει την τιμή 1.
- Η τιμή του μήνα είναι 3, αφού ο Μάρτιος είναι ο τρίτος μήνας του έτους.
- Η τιμή 3 θα αντιστοιχιστεί μέσω του πίνακα imonth στο γράμμα c.
- Οι δύο τιμές θα συνδυαστούν και ο δείκτης index θα πάρει την τιμή c1.

Σε περίπτωση όπου κατά τη διάρκεια της εκτέλεσης του αλγορίθμου ο μήνας αλλάξει ελέγχουμε μέσω της μεταβλητής dday και του πίνακα imonth σε ποιο μήνα βρισκόμαστε. Αν ο μήνας αλλάξει, ο δείκτης ενημερώνεται και πάλι μέσω του iupdate ώστε να δείχνει στο σωστό κελί, ενώ το αριθμητικό μέρος παραμένει ίδιο με πριν. Ο τρόπος με τον οποίο συμβαίνει αυτό φαίνεται στην Εικόνα 3.7.

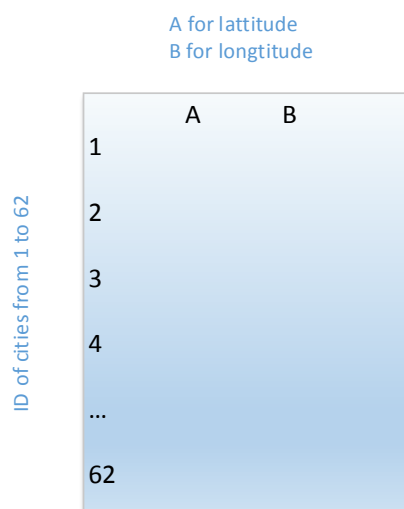
Όπως και προηγουμένως κάθε μία από τις πόλεις υπό μελέτη αντιστοιχίζεται σε έναν αριθμό από το 1 έως το 62, ενώ η στήλη A αναφέρεται στο γεωγραφικό πλάτος της κάθε πόλης και η στήλη B στο γεωγραφικό μήκος. Για την ενημέρωση του γεωγραφικού πλάτους χρησιμοποιούμε τον δείκτη indexmeri, ενώ για το γεωγραφικό μήκος τον δείκτη indexlatti. Η ανάγνωση δεδομένων του αλγορίθμου γίνεται μέσω της εντολής xlsread όπως φαίνεται στην Εικόνα 3.9:

Η ενημέρωση των δεικτών σε αυτή την περίπτωση είναι πιο απλή, ο δείκτης indexmeri ενώνει το αναγνωριστικό της πόλης με το γράμμα A, ενώ ο δείκτης indexlatti ενώνει το αναγνωριστικό της πόλης με το γράμμα B. Κατά τη διάρκεια της εκτέλεσης του αλγορίθμου οι τιμές αυτές παραμένουν σταθερές.

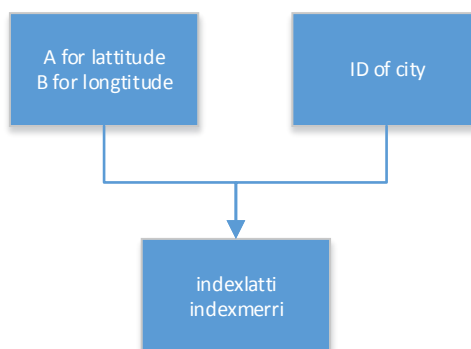


Εικόνα 3.7 Διάγραμμα ροής προσέλασης δεδομένων

Προσπέλαση τοπολογικών δεδομένων



Εικόνα 3.8 Απεικόνιση των πινάκων δεδομένων εισαγωγής για γεωγραφικό μήκος και πλάτος.



Εικόνα 3.9 Διάγραμμα αλληλεπίδρασης μεταβλητών προσπέλασης δεδομένων γεωγραφικού μήκους και πλάτους.

3.8 Διάγραμμα ροής και βηματική εκτέλεση αλγορίθμου

Στην παράγραφο αυτή θα γίνει μία συνοπτική περιγραφή του αλγορίθμου σε σύνολο μέσω ενός διαγράμματος ροής το οποίο περιγράφει βηματικά την εκτέλεσή του. Στις επόμενες παραγράφους θα αναλυθούν λεπτομερώς όλες οι δυνατότητες και λειτουργίες του, ενώ στο τέλος του κεφαλαίου θα παρατεθεί ο κώδικας του σε σύνδεση με τη θεωρία που αναπτύχθηκε έως τώρα.

Ορισμός παραμέτρων και αρχικοποίηση

Με την εκτέλεση του αλγορίθμου αρχικοποιούνται οι μεταβλητές που είδαμε στο κεφάλαιο 3.1, αυτό συμβαίνει μέσω του πηγαίου κώδικα του Matlab, του Excel και του γραφικού περιβάλλοντος.

Βασικό μας μέλημα για την εκτέλεση και τη συμπεριφορά του αλγορίθμου είναι οι μεταβλητές – διακόπτες που ορίσαμε νωρίτερα. Αρχικά εξετάζουμε αν ο αλγόριθμος θα προσομοιώσει ένα ΗΘΣ θέρμανσης ή ένα ΗΘΣ ζεστού νερού χρήσης. Σε περίπτωση που ο διακόπτης **IDHW** έχει την τιμή 1 έχουμε θέρμανση, ενώ αν έχει τιμή 0 τότε έχουμε ZNX.

Θα διασχίσουμε την Εικόνα 3.10 από τα αριστερά προς τα δεξιά, έστω ότι έχει επιλεγεί θέρμανση, τότε ο αλγόριθμος αρχικοποιεί για αυτή τη λειτουργία. Στη συνέχεια ελέγχουμε την τιμή του διακόπτη **IDD** ο οποίος υποδεικνύει αν θα εφαρμόσουμε τη μέθοδο των βαθμομερών.

Σε περίπτωση που ο διακόπτης **IDD** λάβει την τιμή 1 στους υπολογισμούς μας προστίθενται οι μεταβλητές *dd*, *uresidence*, *aresidence*, που εκφράζουν την τιμή των βαθμομερών, το θερμικό συντελεστή ενός κτιρίου και την εξωτερική του επιφάνεια αντίστοιχα, ώστε να μπορέσουμε να εφαρμόσουμε τη μέθοδο των βαθμομερών. Στη συνέχεια λαμβάνουμε την παροχή *m1* και υπολογίζουμε τα θερμικά φορτία *q1* και προχωράμε στο επόμενο βήμα του αλγορίθμου.

Διαφορετικά, αν ο διακόπτης **IDD** λάβει την τιμή 0, εξετάζουμε τον τύπο του φορτίου. Εάν το φορτίο και η παροχή *m1* είναι στατικά, δηλαδή **sload** είναι 0, προχωράμε στο επόμενο βήμα. Αν η παροχή είναι δυναμική - χρονομεταβαλλόμενη, δηλαδή **sload** είναι 1, αντιστοιχούμε την τιμή *m1* με το φορτίο κάθε ώρα που δίνεται από τη μεταβλητή *m1_u*. Στη συνέχεια υπολογίζουμε τα θερμικά φορτία *q1* και προχωράμε στο επόμενο βήμα.

Στην περίπτωση που επιλεγεί ZNX, δηλαδή **IDHW** ισούται με 1, η διαδικασία είναι ίδια με αυτή της θέρμανσης, με τη διαφορά αφενός ότι δεν υπάρχει η λειτουργία των βαθμομερών και αφετέρου ότι προστίθενται οι μεταβλητές *qload*, *mload* για τους υπολογισμούς λόγω της ύπαρξης βάνας ανάμιξης, όπως αναφέραμε στη θεωρία και θα δούμε παρακάτω.

Υπολογισμός ηλιακών παραμέτρων

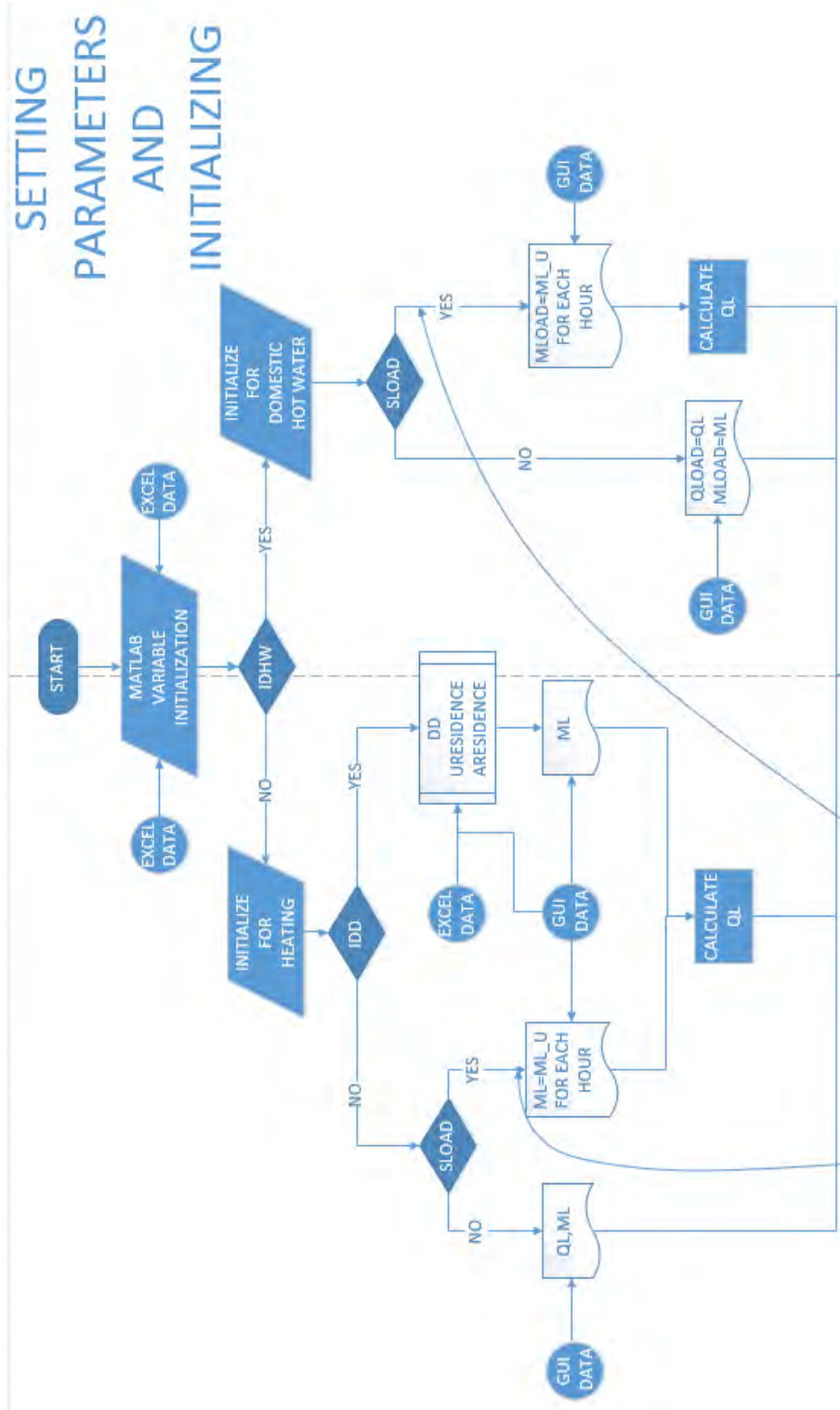
Το επόμενο βήμα του αλγορίθμου είναι ο υπολογισμός του χρόνου, των γωνιών του συλλέκτη και της ακτινοβολίας που δέχεται κατά την εκτέλεση της προσομοίωσης,

όπως φαίνεται στην Εικόνα 3.11. Αρχικά ελέγχουμε αν έχει αλλάξει ο μήνας της προσομοίωσης, όσο εκτελείτε η προσομοίωση, ώστε να ενημερώσουμε τον αλγόριθμο με νέες τιμές. Στη συνέχεια υπολογίζουμε το συντελεστή διόρθωσης λόγω τροχιάς bq , και έπειτα την τρέχουσα ηλιακή και πραγματική ώρα. Με βάση τα δεδομένα μας υπολογίζουμε τις γωνίες απόκλισης, ανατολής- δύσης, ωριαίας ηλιακής, ηλιακού ύψους, ηλιακού αζιμούθιου και πρόσπτωσης, δ , wss , ω , as , gms , θ αντίστοιχα για κάθε χρονική στιγμή.

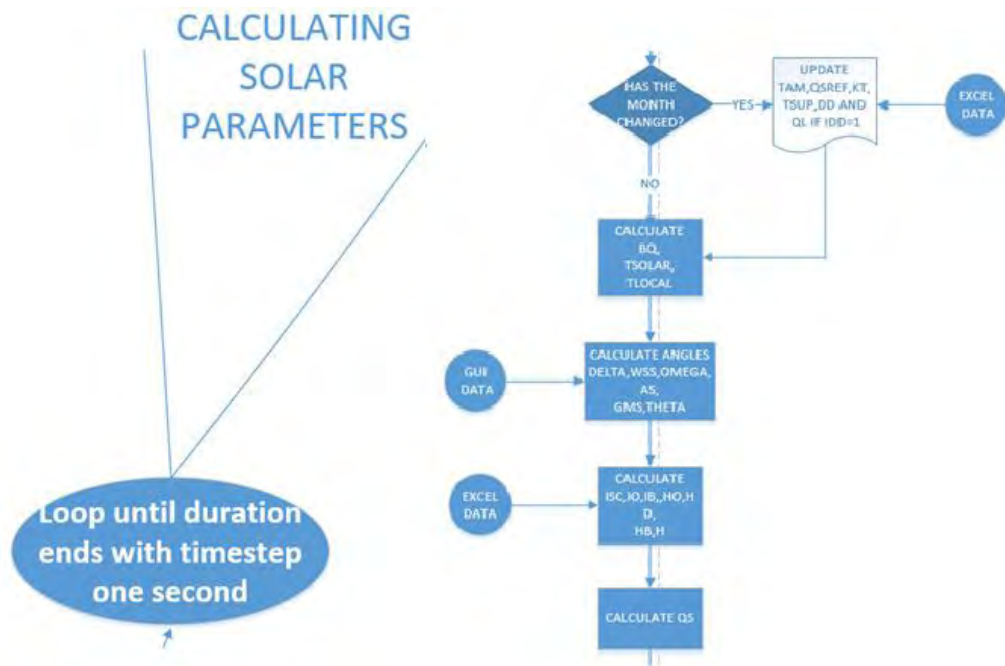
Συνδυάζοντας όλα τα παραπάνω στοιχεία με τα στοιχεία της ακτινοβολίας και τον τρόπο υπολογισμού που περιγράφηκε στην παράγραφο 2.1.5 για την περιοχή υπολογίζουμε την προσπίπτουσα ακτινοβολία στο συλλέκτη και προχωράμε στο επόμενο βήμα.

Υπολογισμοί παρουσία εναλλάκτη

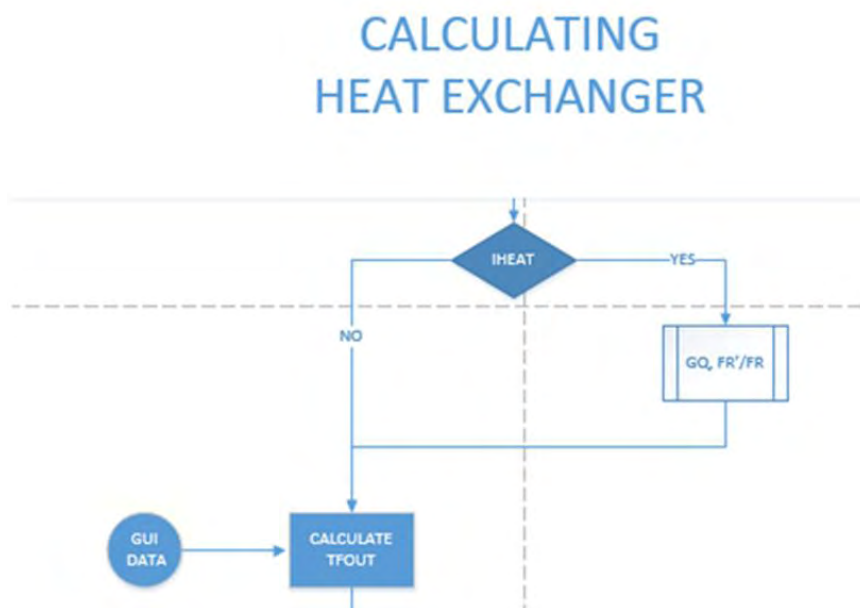
Το επόμενο βήμα είναι να εξετάσουμε την ύπαρξη ή όχι εναλλάκτη, μέσω του διακόπτη **IHEAT**, όπως φαίνεται στην Εικόνα 3.12. Αν ο διακόπτης **IHEAT** λάβει την τιμή 1 τότε υπολογίζουμε την τιμή γ που αντιστοιχεί στη μεταβλητή gq , και το μέτρο απόδοσης του εναλλάκτη F_R'/F_R . Στη συνέχεια υπολογίζεται η θερμοκρασία εξόδου του ρευστού t_{fout} και προχωράμε στο επόμενο βήμα.



Εικόνα 3.10 Διάγραμμα ροής ρύθμισης και αρχικοποίησης παραμέτρων του αλγορίθμου.



Εικόνα 3.11 Διάγραμμα ροής υπολογισμού ηλιακών παραμέτρων.



Εικόνα 3.12 Διάγραμμα ροής λειτουργίας εναλλάκτη.

Έλεγχος και αυτοματισμοί

Για να λειτουργήσει η προσομοίωση του ΗΘΣ, ανεξάρτητα των επιλογών της, πρέπει να υπάρχει ένα σύνολο αυτοματισμών που περιγράφει τον τρόπο συμπεριφοράς του συστήματος. Παρακάτω θα δούμε πως ακριβώς λειτουργεί το σύστημα μας και τι ακριβώς συμβαίνει στο συλλέκτη και τη δεξαμενή κατά την προσομοίωση (Εικόνα 3.13).

Κατά την περίπτωση που η διαφορά της θερμοκρασίας της δεξαμενής αποθήκευσης με τη θερμοκρασία της πλάκας απορρόφησης σε συνθήκες μηδενικής παροχής είναι μεγαλύτερη από 6 βαθμούς, αν η θερμοκρασία της δεξαμενής αποθήκευσης t_{st} υπερβεί τη μέγιστη επιτρεπτή θερμοκρασία t_{stmax} , τότε ο διακόπτης d_c κλείνει και παίρνει την τιμή μηδέν. Αν αυτό δεν ισχύει αλλά η διαφορά της θερμοκρασίας εξόδου t_{fout} με τη θερμοκρασία εισόδου t_{fin} είναι μικρότερη από δύο βαθμούς τότε ο διακόπτης d_c ανοίγει παίρνοντας την τιμή 1. Σε οποιαδήποτε άλλη περίπτωση ο διακόπτης d_c παραμένει κλειστός. Επομένως για τον έλεγχο του συλλέκτη και εναλλάκτη ισχύουν οι παρακάτω συνθήκες:

- Εάν $t_{stag}-t_{fin} \geq 6$ και $t_{fout}-t_{fin} \geq 2$ τότε $d_c=1$.
- Εάν $t_{fout}-t_{fin} < 2$ ή $t_{st} > t_{stmax}$ τότε $d_c=0$.

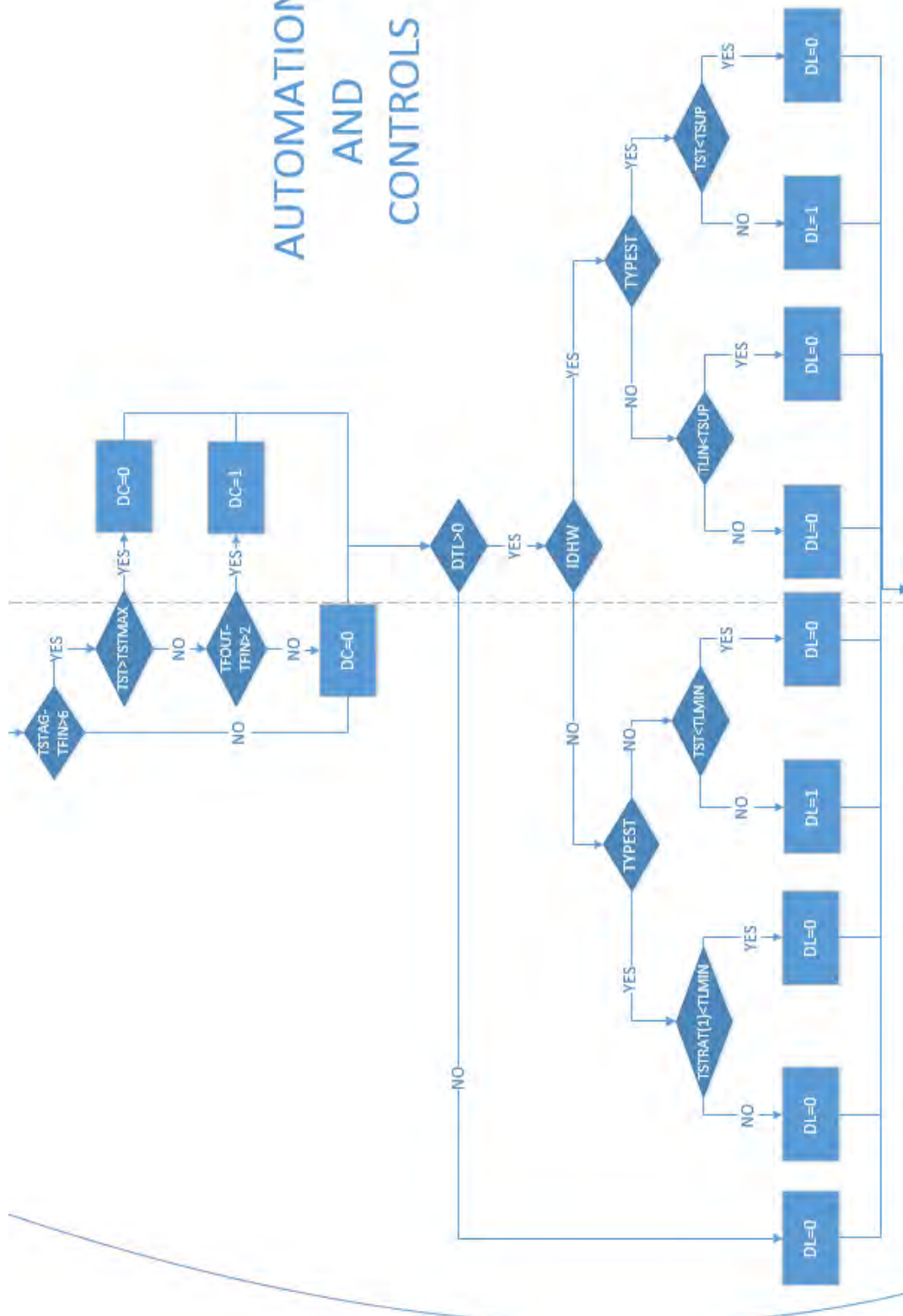
Στη συνέχεια αφού ορίσουμε την διαφορά της θερμοκρασίας εισόδου-εξόδου μέσω της μεταβλητής d_{tl} , εφαρμόζουμε τα εξής:

Αν η διαφορά θερμοκρασίας d_{tl} είναι μεγαλύτερη του μηδενός και αν η θερμοκρασία αποθήκευσης t_{st} είναι μεγαλύτερη από την ελάχιστη θερμοκρασία λειτουργίας του συστήματος t_{stmin} , τότε ο διακόπτης d_l γίνεται 1, είτε έχουμε διαστρωμάτωση είτε όχι.

Αν έχουμε ZNX και αν η θερμοκρασία αποθήκευσης είναι μεγαλύτερη από την ελάχιστη θερμοκρασία παροχής του συστήματος, τότε ο διακόπτης d_l γίνεται 1, είτε έχουμε διαστρωμάτωση είτε όχι. Σε όλες τις άλλες περιπτώσεις ο διακόπτης d_l παραμένει κλειστός. Επομένως για του φορτίου και του δοχείου αποθήκευσης ισχύουν οι παρακάτω συνθήκες:

- Εάν $t_{lin}-t_{lout}=d_{tl} < 0$ ή $t_{st} < t_{stmin}$ τότε $d_l=0$
- Εάν $t_{lin}-t_{lout}=d_{tl} \geq 0$ τότε $d_l=1$

AUTOMATION AND CONTROLS

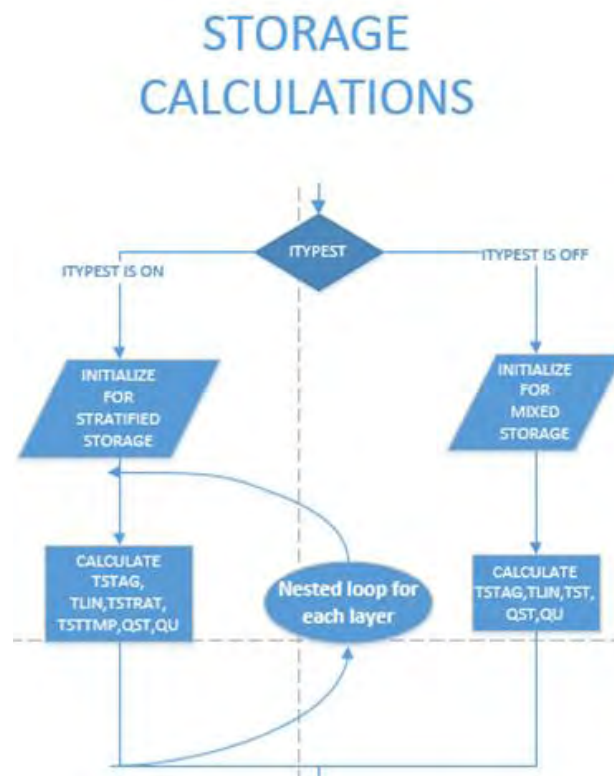


Εικόνα 3.13 Διάγραμμα ροής αυτοματισμών.

Υπολογισμοί δεξαμενής

Στο επόμενο βήμα ο ελέγχουμε το διακόπτη **ITYPEST**. Σε περίπτωση που έχει την τιμή 0 η δεξαμενή μας είναι αναμειγμένη και υπολογίζουμε τη θερμοκρασία της πλάκας απορρόφησης σε συνθήκες μηδενικής παροχής t_{stag} και τη θερμοκρασία εισόδου του ρευστού t_{lin} . Στην περίπτωση που ο διακόπτης λάβει την τιμή 1 η δεξαμενή μας είναι διαστρώματοποιημένη και υπολογίζουμε τη θερμοκρασία της πλάκας απορρόφησης σε συνθήκες μηδενικής παροχής t_{stag} , τη θερμοκρασία εισόδου του ρευστού t_{lin} και τις θερμοκρασίες t_{strat} και t_{stag} μέσω μίας επανάληψης για κάθε επίπεδο της δεξαμενής.

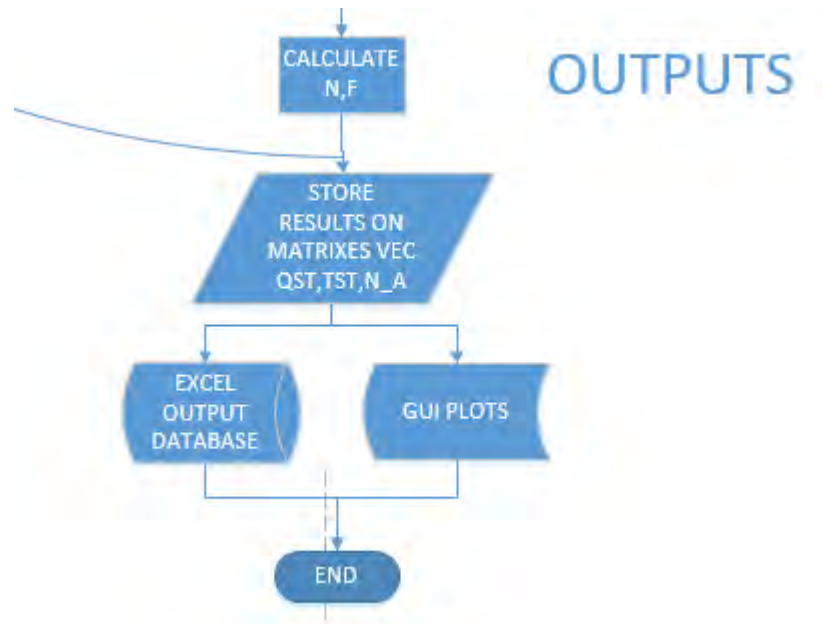
Τέλος και στις δύο περιπτώσεις υπολογίζουμε τη διαθέσιμη προσπίπτουσα ηλιακή ενέργεια και το αποθηκευμένο θερμικό φορτίο της δεξαμενής (Εικόνα 3.14).



Εικόνα 3.14 Διάγραμμα ροής υπολογισμών δεξαμενής αποθήκευσης.

Αποτελέσματα

Στο τελευταίο αυτό βήμα που παρουσιάζεται στην Εικόνα 3.15, ο αλγόριθμος υπολογίζει το βαθμό απόδοσης του συστήματος και το ποσοστό κάλυψης της ζήτησης. Τέλος συλλέγει τα αποτελέσματα στους πίνακες VEC,QST,TST,N_A ώστε να τα εκτυπώσει στο γραφικό περιβάλλον και στο Excel, όπως θα δούμε στις επόμενες παραγράφους.



Εικόνα 3.15 Διάγραμμα ροής εκτύπωσης αποτελεσμάτων.

3.9 Παρουσίαση πηγαίου κώδικα

Στην παράγραφο αυτή θα παρατίθενται αποσπάσματα του κώδικα και θα περιγράφεται η λειτουργία τους σύμφωνα με τα προηγούμενα κεφάλαια. Τα τμήματα που παρατίθενται δεν θα ακολουθούν απαραίτητα τη σειρά εκτέλεσης του αλγορίθμου, αλλά θα είναι χωρισμένα θεματικά, σύμφωνα με τη λειτουργία τους, ώστε να επιτευχθεί η καλύτερη επεξήγησή τους, αξίζει να αναφερθεί ότι ορισμένα αποσπάσματα μπορεί να εμφανίζονται περισσότερο από μία φορά, αρκεί να συμβάλουν στην επεξήγηση του αλγορίθμου. Με την επεξήγηση του αλγορίθμου ακολουθεί η πιστοποίησή του και η επίλυση μερικών παραδειγμάτων στο κεφάλαιο 4 ώστε η κατανόηση του να γίνει με σαφήνεια. Ο πλήρης πηγαίος κώδικας της εργασίας βρίσκεται στα παραρτήματα Α και Β.

3.9.1 Διακόπτες

Παρακάτω φαίνονται οι διακόπτες τους οποίους ο χρήστης ενεργοποιεί μέσω του γραφικού περιβάλλοντος και ενεργοποιούν τις διάφορες λειτουργίες της προσομοίωσης, όπως αναφέρθηκε στα προηγούμενα κεφάλαια.

```
%switches%

%for first way of calculating radiation 1, for second 0
irv=get(handles.Gui_data,'Value')
irv=get(handles.Real_env,'Value')
%for domestic hot water 1, otherwise 0%
idhw=get(handles.DHW,'Value');
%for load variation 1, otherwise 0%
sload=get(handles.Time_load,'Value');
%for heat exchanger 1, otherwise 0%
iheat=get(handles.Heat_exch,'Value');
%for stratified storage 1, mixed 0
itypest=get(handles.Stratified,'Value');
%switch for degree days
idd=get(handles.Degrees_days,'Value');
```

Η μέθοδος υπολογισμού της ακτινοβολίας ελέγχεται από τη μεταβλητή *irv*, η λειτουργία του ZNX από το διακόπτη *idhw*, το δυναμικό φορτίο από το διακόπτη *sload*. Ο εναλλάκτης θερμότητας, η διαστρωμάτωση της δεξαμενής και οι βαθμοημέρες ελέγχονται από τους διακόπτες *iheat*, *itypest* και *idd* αντίστοιχα. Οι διακόπτες αυτοί δέχονται τιμές μηδέν ή ένα. Η τιμή ένα ενεργοποιεί ενώ η τιμή μηδέν απενεργοποιεί τις αντίστοιχες λειτουργίες του κώδικα.

3.9.2 Χρονικό βήμα εκτέλεσης

Στον παρακάτω πίνακα εμφανίζονται οι μεταβλητές που χρησιμοποιούμε για τον ορισμό του χρονικού βήματος, όπως ακριβώς τις περιγράψαμε στο στην παράγραφο 3.5.

```
%day to begin the simulation
dday=get(handles.Day_input,'Value');

%month for the simulation
month=get(handles.Month_input,'Value');

%number of days to run the simulation
range=get(handles.Range_input,'Value');

%timestep for the simulation in minutes, current timestep is
1 second
timestep=1/86400;
```

Στο παρακάτω τμήμα ψευδοκώδικα εμφανίζεται η θέση της επαναληπτικής διαδικασίας στον αλγόριθμο. Σύμφωνα με τα στοιχεία που δόθηκαν αρχικοποιείται η προσομοίωση, ενώ όλοι οι υπολογισμοί που αναφέρθηκαν στο κεφάλαιο 2 και 3 γίνονται σύμφωνα με το χρονικό βήμα που ορίσαμε.

```
INITIALIZATION

NON-TIME DEPENDENT CALCULATIONS

while (nday<(day+range))

TIME DEPENDENT CALCULATIONS

nday=nday+timestep;

end
```

3.9.3 Ενημέρωση με πραγματικά δεδομένα

Για την ενημέρωση των δεδομένων του αλγορίθμου, όπως περιγράψαμε στην παράγραφο 3.7 χρειαζόμαστε τις παρακάτω βοηθητικές μεταβλητές και πίνακες, ορισμένους όπως φαίνεται στο παρακάτω τμήμα του κώδικα.

```
%vectors include day per year, month index and sun duration
per month
iupdate=[0 31 59 90 120 151 181 212 243 273 304 334 365];
imonth= ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j',
'k', 'l'];
sun_duration=[4.5 5.1 6 8 9.8 11.4 12.1 11.5 9.4 7 5.5 4.4
4.5];
%city for the simulation
city=get(handles.City,'Value');
```

Οι μεταβλητές – δείκτες που χρησιμοποιήσαμε για την προσπέλαση και τη μεταφορά των δεδομένων από το Excel στο Matlab εμφανίζονται παρακάτω.

```
%indexes for traversing the excel file
index=strcat(imonth(month),icity);
indexmeri=strcat('b',icity);
indexlatti=strcat('a',icity);
```

Ακολουθεί ο τρόπος αρχικοποίησης των μεταβλητών που παίρνουν πραγματικές τιμές και ενημερώνονται σύμφωνα με τις μεταβλητές - δείκτες.

```
%update variables from real values
tam = xlsread('TEE.xls','s2',index)+273;
month=month+1;
DD =
xlsread('TEE.xls','s3',index)*uresidence*aresidence/(iupdate(month)
-iupdate(month-1)); %degree days
```

```

qsref = xlsread('TEE.xls','s4',index)*1000/(iupdate(month)-
iupdate(month-1))/sun_duration(month-1) %fixme
month = month -1;
kt = xlsread('TEE.xls','s5',index); %kt coefficient
tsup = xlsread('TEE.xls','s6',index)+273 %supply temperature from
water network
if (tsup==273)
    tsup = (xlsread('TEE.xls','s2',index)-
xlsread('TEE.xls','s7',index)*exp(-
d_depth*((pi/(365*diff))^0.5))*cos((2*pi/365)*(nday-soil_day-
(d_depth/2)*((365/(pi*diff))^0.5))))+273 %estimation of temperature
from water network
end

```

Κατά την περίπτωση που ο μήνας της προσομοίωσης αλλάξει οι μεταβλητές πρέπει και πάλι να ενημερωθούν. Αυτό συμβαίνει μέσω των συνθηκών ελέγχου του παρακάτω κώδικα.

```

%This section updates variables based on real values in case the
month has changed
for ii=12:-1:1
    if mod(nday,iupdate(ii))<=timestep
        if (nday >= iupdate(ii))
            index=strcat(imonth(ii),icity);
            tam = xlsread('TEE.xls','s2',index)+273;
            month=month+1;
            DD =
xlsread('TEE.xls','s3',index)*uresidence*aresidence/(iupdate(month)
-iupdate(month-1));
            if (idd==1)
                ql=DD;
            end
            qsref =
xlsread('TEE.xls','s4',index)*1000/(iupdate(month)-iupdate(month-
1))/sun_duration(month);
            month=month-1;
            kt = xlsread('TEE.xls','s5',index);
            tsup = xlsread('TEE.xls','s6',index)+273;
            if (tsup==273)
                tsup = (xlsread('TEE.xls','s2',index)-
xlsread('TEE.xls','s7',index)*exp(-
d_depth*((pi/(365*diff))^0.5))*cos((2*pi/365)*(nday-soil_day-
(d_depth/2)*((365/(pi*diff))^0.5))))+273;
            qsmax=qmax(nday,kt,correction,omega_correction,phi,gmr,b
eta,rr,irv)
            end
        end
    end
end
end

```

3.9.4 Ηλιακή και πραγματική ώρα

Αρχικά για τη λειτουργία του αλγορίθμου πρέπει να οριστεί ο ηλιακός και πραγματικός χρόνος, όπως περιγράφηκε στο κεφάλαιο 2.1.6. Αρχικά υπολογίζεται ο συντελεστής διόρθωσης τροχιάς, στη συνέχεια υπολογίζεται η διόρθωση της ηλιακής

ώρας ενώ τέλος υπολογίζεται ο πραγματικός και ηλιακός χρόνος και γίνεται διόρθωση του με τη θερινή και χειμερινή ώρα.

```
%This section calculates solar and local time

%trajectory coefficient%
bq=(360*(floor(nday-1)))/365;

%time correction factor based on trajectory coefficient
etraj=0.0172+0.3278*cosd(bq)*7.3456*sind(bq)-
3.3468*(cosd(bq))^2-9.3544*(sind(bq)^2);

%change of time +-1 hour at 26/3 and 29/10
if (nday>85 && nday<302)
    correction=1;
else
    correction=0;
end

%calculation of local time%
tlocal= mod(nday,floor(nday))*24-correction;

%calculation of solar time%
tsolar=tlocal-(4*(PSist-PSilocal)+etraj)/60;
```

3.9.5 Υπολογισμός ακτινοβολίας και ηλιακών γωνιών

Για τον υπολογισμό των ηλιακών γωνιών, όπως περιγράψαμε στα κεφάλαια 2.1.3, 2.1.4, 2.15 είναι απαραίτητες οι παρακάτω μεταβλητές

```
%Angle beta%
beta=20;
%Angle gamma%
gmr=0;
%Angle omega correction when the day changes%
omega_correction=180;
```

Σύμφωνα με τις παραπάνω μεταβλητές και το χρονικό μας βήμα υπολογίζουμε αρχικά τη γωνία δ , στη συνέχεια μέσω αυτής υπολογίζουμε τη γωνία ανατολής-δύσης ω_{SS} και τη διάρκεια της ηλιοφάνειας. Ακολουθεί ο υπολογισμός της ωριαίας ηλιακής γωνίας ω ο οποίος διορθώνεται με το πέρασμα κάθε ημέρας. Στη συνέχεια με τα παραπάνω δεδομένα υπολογίζουμε τις γωνίες ηλιακού ύψους, την αζιμούθια γωνία και τη γωνία πρόσπτωσης αντίστοιχα.

```
%This section calculates solar radiation based on collector angles
and location

%delta calculations%
delta=23.45*sind(360*((284+floor(nday))/365));

%wss calculations%
wss=acosd(-tand(phi)*tand(delta));
```

```

%sunshine duration%
sunshine=wss/15;

%omega calculation
if (mod(count_timestep,1)==0)
    omega_correction=180;
end
omega=(15*(tsolar-12))+omega_correction;

%as calculation%
as=(asind(sind(delta)*sind(phi)+cosd(delta)*cosd(phi)*cosd(omega)));
;
%gamma s angle%
gms=asind(cosd(delta)*sind(omega)/cosd(as));

%theta angle%
theta=acosd(sind(as)*cosd(beta)+cosd(as)*sind(beta)*cosd(gms-gmr));

```

Στη συνέχεια χρησιμοποιώντας τους υπολογισμούς για τις γωνίες του συστήματός μας εφαρμόζουμε κατά κανόνα τη μέθοδο που περιγράψαμε στο κεφάλαιο 2.1.5 χρησιμοποιώντας τα δεδομένα της παραγράφου 2.1.9. Λόγω το ότι η εξίσωση της ακτινοβολίας που χρησιμοποιούμε στην προσομοίωση, στην αρχή και το τέλος της ημέρας, λαμβάνει αρνητικές τιμές χρησιμοποιούμε το διακόπτη `day_night`, ώστε να μην αφαιρεί ενέργεια από το σύστημα, ενώ η αρχή της ημέρας ορίζεται με την έναρξη της εξωγήινης ακτινοβολίας.

```

%solar stable
isc=1367;

%extraterrestrial solar radiation
io=isc*(1+0.034*cosd((360*(nday))/365));
ho=io*sind(as);
if ho>0
if (irv==0)

    h=kt*ho;
    if (kt<= 0.85 && kt>=0.30)
        ib=-520+1800*kt;
    elseif (kt<0.3)
        ib=0;
    else
        ib=0;
    end

    hb=ib*sind(as);
    hd=h-hb;

else

    h=kt*ho;
    if (kt<0.35)
        hd=(1-0.249*kt)*h;
    elseif (kt>=0.35 && kt<=0.75)
        hd=(1.557-1.84*kt)*h;
    elseif (kt<0.85)

```

```

        hd=0.177*h;
    else
        hd=0;
        h=0;
    end
    ib=(h-hd)/sind(as);
    hb=ib*sind(as);
end

%calculates the actual solar radiation%
qs=(cosd(theta)/sind(as))*hb+hd*((cosd(beta/2))^2)+h*rr*((sind(beta/2))^2);
else
    qs=0;
end

```

Στα πλαίσια του υπολογισμού της ακτινοβολίας, πρέπει να αναφέρουμε ότι και η θερμοκρασία κατά τη διάρκεια της ημέρας παρουσιάζει διακύμανση. Για να την προσομοιώσουμε χρησιμοποιούμε ημιτονοειδή κατανομή μέσω των μεταβλητής της ημέρας.

```

%insert temperature variation dta [k]%
dta=get(handles.Input_dta,'Value');
%ambient temperature variation during the day
ta=tam+dta*sin(-pi*(floor(nday)-nday));

```

Ακόμη, είναι δυνατό να βρεθεί η μέγιστη ακτινοβολία για την ημέρα, χρησιμοποιώντας το παρακάτω τμήμα κώδικα.

```

function
qsmx=qmax(ndaymax,ktmax,correctionmax,omega_correctionmax,phimax,g
mrmax,betamax,rrmax,irvmx)

%delta angle for the day
deltamax=23.45*sind(360*((284+floor(ndaymax))/365));

%max omega angle%
omegamax=180+omega_correctionmax; %set it up for multiple days

%max as calculation%
asmx=(asind(sind(deltamax)*sind(phimax)+cosd(deltamax)*cosd(phimax)
)*cosd(omegamax));

%max gamma s angle%
gmsmx=asind(cosd(deltamax)*sind(omegamax)/cosd(asmx));

%max theta angle%
thetamax=acosd(sind(asmx)*cosd(betamax)+cosd(asmx)*sind(betamax)*
cosd(gmsmx-gmrmax));

%solar stable
isc=1367;

%max extraterrestrial solar radiation
iomx=isc*(1+0.034*cosd((360*(ndaymax))/365));
homx=iomx*sind(asmx);
if (irvmx==0)

```

```

hmax=ktmax*homax;

if (ktmax<= 0.85 && ktmax>=0.30)
    ibmax=-520+1800*ktmax;
elseif (ktmax<0.3)
    ibmax=0;
else
    ibmax=0;
end

hbmax=ibmax*sind(asmx);
hdmax=hmax-hbmax;

else

hmax=ktmax*homax;

if (ktmax<0.35)
    hdmax=(1-0.249*ktmax)*hmax;
elseif (ktmax>=0.35 && ktmax<=0.75)
    hdmax=(1.557-1.84*ktmax)*hmax;
elseif (kt<0.85)
    hdmax=0.177*hmax;
else
    hdmax=0;
    hmax=0;
end
ibmax=(hmax-hdmax)/sind(asmx);

end

%calculates the maximum solar radiation%
qsmax=(ibmax*cosd(thetamax) + (hdmax) * (cosd(betamax/2) ^2) +hmax*rrmax*
(sind(betamax/2) ^2));

end

```

3.9.6 Υπολογισμοί διαστάσεων δεξαμενής

Σε περίπτωση που οι διαστάσεις της δεξαμενής δεν μας δίνονται, μπορούν να υπολογιστούν. Για να συμβεί αυτό χρειαζόμαστε τις παρακάτω μεταβλητές.

```

%StOrAge dAtA iNPut%

%insert storage surface ast [m^2]%
ast=get(handles.Input_ast, 'Value');
%insert storage heat transfer coefficient ust [W/m^2*k]%
ust=get(handles.Input_ust, 'Value');
%insert storage capacity mst*cp {J/k}%
mst=get(handles.Input_mst_slid, 'Value');
%insert maximum storage temperature tstmax allowed [k]%
tstmax=373;
%Water density%
rho=1000*(1-
(tstmax+288.9414)/(508929.2*(tstmax+68.12963))*(tstmax-
3.9863)^2);
%Storage volume
vstorage=0.32;
%Storage inner diameter
rindia=3;

```


Με δεδομένη τη χωρητικότητα της δεξαμενής και την πυκνότητα του φέροντος ρευστού στη μέγιστη δυνατή θερμοκρασία λειτουργίας της δεξαμενής, εφαρμόζουμε τον παρακάτω αλγόριθμο. Να σημειωθεί ότι για ενεργοποιηθεί η λειτουργία αυτή ο χρήστης πρέπει να δώσει αρνητική τιμή στην επιφάνεια της δεξαμενής κατά την εισαγωγή δεδομένων.

```
if (ast<0) %Storage area calculation if Ast is not given and
the volume storage is given
    vstorage=mst/(rho*c);
    lstorage=(4*vstorage*(rindia^2)/pi)^0.33;
    dst=lstorage/rindia;
    ast=pi*(dst/2+lstorage)
end
```

3.9.7 Στοιχεία για τους θερμικούς υπολογισμούς

Στους παρακάτω πίνακες παρατίθενται οι μεταβλητές εισόδου του προγράμματος που χρησιμοποιούνται για τους υπολογισμούς των φορτίων και του συλλέκτη, ομαδοποιημένες. Αρκετές από αυτές τροποποιούνται κατά την εκτέλεση του προγράμματος όπως θα δούμε παρακάτω.

```
%Collector data input%

%insert collector surface a [m^2]%
a=get(handles.depthSlider,'Value');
%insert collector value b%
b=get(handles.Input_b,'Value');
%insert collector property ta_eff%
ta_eff=get(handles.Input_taeff,'Value');
%insert collector heat removal parameter fr%
fr=get(handles.Input_fr,'Value');
%insert specific heat of working fluid [J/kg*k]%
c_v=get(handles.collector,'Value');

if c_v==1 % water collector
    c=4180;
elseif c_v==2 %air collector
    c=1;
end

%Enviromental data input%

%insert max heat flux qs,max [W/m^2]%
qsmax=get(handles.Input_q,'Value');

%Load data input%

%insert load ql%
ql=get(handles.Input_ql,'Value');
%insert load mass flow ml [kg/s]%
ml=get(handles.Input_ml,'Value');
```

```
%insert load minimum temperature tl,min [k]%  
tlmin=get(handles.Input_tlmin,'Value');
```

Εδώ πρέπει να αναφέρουμε πώς η μεταβλητή c_v ελέγχει την πυκνότητα του φέροντος ρευστού. Στην παρούσα εργασία έχει χρησιμοποιηθεί μόνο η προσομοίωση του νερού, έχει αφαιρεθεί όμως η επιλογή για μελλοντική χρήση.

3.9.8 Βαθμομημέρες

Για την περίπτωση των βαθμομημερών, όπως ορίστηκαν στο κεφάλαιο 2.1.8 χρειαζόμαστε επιπρόσθετα τις παρακάτω μεταβλητές.

```
%residence heat transfer coefficient  
uresidence=get(handles.Uresidence,'Value');  
  
%area of residence  
aresidence=get(handles.Aresidence,'Value');
```

Με τη χρήση των δεδομένων του κεφαλαίου 2.1.9 υπολογίζουμε το θερμικό φορτίο που απαιτείται για τη θέρμανση του χώρου και στη συνέχεια το αντιστοιχούμε με τη μεταβλητή του φορτίου του αλγορίθμου.

```
DD = xlsread('TEE.xls','s3',index)*uresidence*aresidence/  
(iupdate(month)-iupdate(month-1));  
  
if (idd==1)  
    ql=DD;  
end
```

3.9.9 Θερμοκρασία νερού δικτύου ύδρευσης

Για την εκτίμηση της θερμοκρασίας του νερού δικτύου ύδρευσης, σε περίπτωση που τα στοιχεία του αντίστοιχου πίνακα από το κεφάλαιο 2.1.9 είναι ελλιπή, χρειαζόμαστε επιπρόσθετα τις μεταβλητές του πίνακα που ακολουθεί.

```
%depth for the temperature calculation  
d_depth=0.5;  
%day with the lowest temperature of the soil  
soil_day=30;  
%soil heat transfer coefficient for water calculations  
i_diff=get(handles.collector,'Value');  
if i_diff==1 %ΓΡΑΝΙΤΗΣ  
    diff=0.49*0.000001;  
elseif i_diff==2 %ΑΣΒΕΣΤΟΛΙΘΟΣ  
    diff=0.8*0.000001;  
elseif i_diff==3 %ΜΑΡΜΑΡΟ  
    diff=1.39*0.000001;  
elseif i_diff==4 %ΨΑΜΜΑΜΙΤΗΣ  
    diff=1.06*0.000001;  
elseif i_diff==5 %ΑΡΓΙΛΟΣ  
    diff=0.139*0.000001;  
end
```

Χρησιμοποιώντας τα παραπάνω στοιχεία και τις εξισώσεις του κεφαλαίου 2.1.7 μπορούμε να εκτιμήσουμε την επιθυμητή θερμοκρασία. Η λειτουργία αυτή ενεργοποιείται όταν το φύλλο του Excel είναι κενό ή μηδέν, δηλαδή η θερμοκρασία του δικτύου ύδρευσης είναι $t_{sup}=273$.

```
tsup = xlsread('TEE.xls','s6',index)+273;
if (tsup==273)
    tsup = (xlsread('TEE.xls','s2',index) -
xlsread('TEE.xls','s7',index))*exp(-
d_depth*((pi/(365*diff))^0.5))*cos((2*pi/365)*(nday-
soil_day-(d_depth/2)*((365/(pi*diff))^0.5))))+273;
end
```

3.9.10 Ζεστό νερό χρήσης

Κατά την περίπτωση όπου το ηλιακό θερμικό μας σύστημα έχει ως σκοπό τη θέρμανση ζεστού νερού χρήσης, η χρήση και αρχικοποίηση των μεταβλητών αλλάζει, χωρίς όμως να αλλάζει η γενική δομή του αλγορίθμου.

Η θερμοκρασία εξόδου t_{lout} αρχικοποιείται παίρνοντας την τιμή της παροχής από το νερό του δικτύου t_{sup} είτε αναφερόμαστε στην περίπτωση της διαστρωμάτωσης είτε όχι. Μία ακόμη διαφορά είναι πώς όταν η θερμοκρασία εισόδου είναι μεγαλύτερη από την ελάχιστη θερμοκρασία λειτουργίας του συστήματος t_{lmin} η παροχή δίνεται από τον τύπο $m_l=q_{load}/((t_{lin}-t_{sup})*c)$ και το φορτίο από τον τύπο $q_l=m_l*(t_{lin}-t_{sup})*c$, ενώ κατά την περίπτωση όπου η θερμοκρασία t_{lin} είναι μικρότερη ή ίση με την ελάχιστη θερμοκρασία τη θέση της t_{lin} την παίρνει η θερμοκρασία t_{lmin} .

```
if (idhw==1 && sload==0)
    qload=get(handles.Input_ql,'Value');
end

if (idhw==1) %domestic hot water%
    tlout=tsup;
else
    tlout=300;
end
%different initialization in case we have dhw
if (idhw==1)
    if (itypest==0)
        tlin=tstrat(1);
    end
    if (tlin>tlmin)
        ml=qload/((tlin-tsup)*c);
        ql=ml*(tlin-tsup)*c;
    else
        ml=qload/((tlmin-tsup)*c);
        ql=ml*(tlmin-tsup)*c;
    end
end
```

3.9.11 Εναλλάκτης

Κατά την περίπτωση όπου έχουμε εναλλάκτη θερμότητας πρέπει πρώτα να υπολογιστεί ο συντελεστής γ , στη συνέχεια πρέπει να υπολογιστεί η μέγιστη παροχή μάζας ανάμεσα στο κύκλωμα του συλλέκτη και το φορτίου, ώστε να βρούμε το κλάσμα F_R'/F_R το οποίο αλλάζει με τη χρήση εναλλάκτη. Μία ακόμη ποσότητα που αλλάζει είναι η θερμοκρασία εξόδου. Όλα αυτά αντικατοπτρίζονται στον κώδικα που ακολουθεί.

```
gq=mc*c*tam/qsmax/a;%gamma coefficient for heat exchanger

%if we have heat exchanger, then fr'/fr is calculated%
    if (iheat==1)
        if (mc*c>ml*c && ml>0) %find minimum mass flow rate%

fr_h=fr/(1+((ast*fr*u)/(mc*c))*((mc*c)/(eps*ml*c))-1));
        else

fr_h=fr/(1+((ast*fr*u)/(mc*c))*((mc*c)/(eps*mc*c))-1));
        end
    end
    if (iheat==1) %tfout is different if we have heat exchanger%
        tfout=tfin-
        tlout*(fr_h*b/gq)+fr_h*tam/gq*(ta_eff*qs/qsref+b*ta/tam);
    else
        tfout=tfin*(1-
        fr*b/gq)+fr*tam/gq*(ta_eff*qs/qsref+b*ta/tam);
    end
```

3.9.12 Δυναμικό φορτίο

Για την περίπτωση του δυναμικού φορτίου που αναλύθηκε στην παράγραφο 2.2.7 χρειαζόμαστε τον δυναμικό πίνακα φορτίων και τον πίνακα που δείχνει σε ποιες ώρες αντιστοιχούν.

```
%Time table
hr = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
22 23 0];
%matrix of loads in case choice dynamic load is selected
ml_u=get(handles.Time_dependent_load,'Data')/3600;
```

Υπολογίζουμε την παροχή και το δυναμικό φορτίο συγκρίνοντας τον πίνακα των ωρών με τον πραγματικό χρόνο του αλγορίθμου, όπως φαίνεται στον παρακάτω τμήμα.

```
if (sload==1 && idhw==1)
    for i=1:24
        if (hr(i)==(floor(tlocal)))
            mload=ml_u(i);
            qload=mload*(tlmin-tsup)*c;
        end
    end
end
```

3.9.13 Υπολογισμοί με ή χωρίς διαστρωμάτωση δοχείου αποθήκευσης

Στον παρακάτω πίνακα παρατίθεται ο κώδικας των υπολογισμών του ΗΘΣ με ή χωρίς διαστρωμάτωση, όπως παρουσιάστηκε στο κεφάλαιο 2.

Σε περίπτωση της δεξαμενής με ενιαία θερμοκρασία που αναλύσαμε στο κεφάλαιο 2.2.5.1 ο αλγόριθμος χρησιμοποιεί τις τιμές που έχει λάβει ήδη σύμφωνα με όσα αναφέρθηκαν παραπάνω.

Κατά την περίπτωση της διαστρωμάτωσης που αναλύσαμε στο κεφάλαιο 2.2.5.2 εισάγονται νέες μεταβλητές για την προσομοίωση της και η αρχικοποίησή της αλλάζει. Βασικότερη από αυτές τις μεταβλητές είναι αυτή των layers που καθορίζει τον αριθμό των στρωμάτων της δεξαμενής.

```
%Stratified Water Tank%

%Number of layers of storage tank%
layers=5;

%initialize variables%
if (itypest==0)
    for i = 1:(layers)
        tstrat(i)=300;
        gm(i)=0;
        fm(i)=0;
        cld(i)=0;
        ldd(i)=0;
        tsttmp(i)=283;
    end
end

%initialization for stratified storage
if (itypest==1)
    tfin=tst;
elseif (itypest==0)
    tfin=tstrat(layers);
    tlin=tstrat(1);
else
    tfin=tst;
end
```

```
%temperature of storage
tstag=ta+ta_eff*qs/b/qsmax*tam;

%calculates usefull solar radiation%
qu=dc*mc*c*(tfout-tfin);

%This area calculates what happens when the storage is mixed
or statrified
if (itypest==1)
```

```

tlin=tst;
qst=dc*qu-dl*ql-ust*ast*(tst-ta);
tst=tst+(qst)/mst;
elseif (itypest==0)
tlin=tstrat(1);
frlcst=mc*c*dc/(mst/layers);
frldst=ml*c*dl/(mst/layers);
qst=dc*qu-dl*ql-ust*ast*(tst-ta);

    for i=1:layers

        sum_fm=0;
        sum_gm=0;
        fm(i)=0;
        gm(i)=0;

        if (i>1)

            if (tfout>tstrat(i) && tfout<tstrat(i-1))
                fm(i)=1;
            else
                fm(i)=0;
            end

            for j=1:(i-1)
                sum_fm=sum_fm+fm(j);
            end

            cld(i)=(tstrat(i-1)-tstrat(i))*sum_fm;

        else

            if (tfout>tstrat(i))
                fm(i)=1;
            else
                fm(i)=0;
            end
            sum_fm=0;
            cld(i)=0;

        end

        if (i<layers)
            if (tlout<tstrat(i) && tlout>tstrat(i+1))
                gm(i)=1;
            else
                gm(i)=0;
            end

            for j=(i+1):layers
                sum_gm=sum_gm+gm(j);
            end

            ldd(i)=(tstrat(i+1)-tstrat(i))*sum_gm;

        else

            if (tlout<tstrat(i))

```

```

        gm(i)=1;
    else
        gm(i)=0;
    end

    sum_gm=0;
    ldd(i)=0;

    end
    tsttmp(i)=tstrat(i)+frlcst*(fm(i)*(tfout-
tstrat(i))+cld(i))+frldst*(gm(i)*(tlout-tstrat(i))+ldd(i))-
ust*(ast/layers)*(tstrat(i)-ta)/(mst/layers);
    qst=qst+mst*(tsttmp(i)-tstrat(i))/layers;
    tstrat(i)=tsttmp(i);
    end
    tst=tst+qst/mst;
end

```

Το μικρό χρονικό βήμα του αλγορίθμου καλύπτει τις συνθήκες χρονικού βήματος που αναφέραμε κατά την ανάλυση της δεξαμενής οπότε δεν χρειάζονται περαιτέρω συνθήκες ελέγχου.

3.9.14 Αυτοματισμοί

Το σύστημα των αυτοματισμών αποτελεί ένα από τα βασικότερα στοιχεία της προσομοίωσης. Σύμφωνα με το κεφάλαιο 3.8 παρακάτω παρατίθεται ο αντίστοιχος κώδικας λειτουργίας του μέσω των διακοπών dc,dl.

```

if (m1==0) %when the flow rate of the load is zero, then the
temperature difference is also zero in order to avoid NAN
    dtl=0;
else
    dtl=q1/m1/c;
end

if (tstag-tfin>6)
    if (tst>tstmax)
        dc=0;
        if (itypest==0)
            if (tsttmp(1)<tstmax)
                dc=1;
            end
        end
    else
        if (tfout-tfin>2)
            dc=1;
        else
            dc=0;
        end
    end
else
    dc=0;
end

if (dtl>0)

```

```

    if (idhw==1) %if the system is dhw then the control
about the load is different
        if (itypest==1) %Mixed tank
            if (tst<tsup)
                dl=0;
            else
                dl=1;
            end
        else %! Stratified tank
            if (tlin<tsup)
                dl=0;
            else
                dl=1;
            end
        end
    else
        if (itypest==1) %Mixed tank
            if (tst<tlmin)
                dl=0;
            else
                dl=1;
            end
        else
            if (tstrat(1)<tlmin)
                dl=0;
            else
                dl=1;
            end
        end
    end
else
    dl=0;
end

```

3.9.15 Αποτελέσματα Excel και γραφικού περιβάλλοντος

Αφού ολοκληρωθούν όλοι οι υπολογισμοί, το μόνο που απομένει είναι να εκτυπώσουμε τα δεδομένα μας. Τα συνοπτικά και τυποποιημένα αποτελέσματα εμφανίζονται στο γραφικό περιβάλλον υπό μορφή γραφήματος, ενώ οι υπόλοιπες μεταβλητές αποθηκεύονται σε ένα αρχείο Excel. Αξίζει να αναφέρουμε ότι ο αλγοριθμικός χρόνος εκτύπωσης είναι κάθε 10 λεπτά. Στους επόμενους πίνακες παρατίθεται η αρχικοποίηση των μεταβλητών και στη συνέχεια ο τρόπος αποθήκευσης.

```

%plot variables

global TST;
TST=zeros(1,250*range);
TST(:)=300;
global QU;
QU=zeros(1,250*range);
global N_A;
N_A=zeros(1,250*range);
xmin=0*range;
xmax=250*range;
global x;
x=linspace(xmin,xmax,250*range);

```



```
%vector of results%
vec=zeros(30);
```

```
if mod(loopNr,360)==0
    cell=cell+1;

    TST(1,cell)=tst;
    QU(1,cell)=qu/1000;
    N_A(1,cell)=n*100;
    QS(1,cell)=qs/1000;
    if (itypest==0)
        TST_str(1,cell)=tstrat(1);
    end

    vec(cell,1)=tlocal;      %a
    vec(cell,2)=tst;        %b
    vec(cell,3)=tfin;       %c
    vec(cell,4)=tfout;      %d
    vec(cell,5)=qst;        %e
    vec(cell,6)=qu;         %f
    vec(cell,7)=dtl*ml*c;   %g
    vec(cell,8)=n;          %h
    vec(cell,9)=ql;         %i
    vec(cell,10)=qs;        %j
    vec(cell,11)=tlin;      %k
    vec(cell,12)=tlout;     %l
    vec(cell,13)=dc;        %m
    vec(cell,14)=dl;        %n
    vec(cell,15)=ta;        %o
    vec(cell,16)=ml;        %p
    vec(cell,17)=mc;        %q
    vec(cell,28)=omega;
    vec(cell,29)=rrq;
    % vec(cell,30)=;

    if (itypest==0)
        vec(cell,18)=tstrat(1);
        vec(cell,19)=tstrat(2);
        vec(cell,20)=tstrat(3);
        vec(cell,21)=tstrat(4);
        vec(cell,22)=tstrat(5);
        vec(cell,23)=tsttmp(1);
        vec(cell,24)=tsttmp(2);
        vec(cell,25)=tsttmp(3);
        vec(cell,26)=tsttmp(4);
        vec(cell,27)=tsttmp(5);
    end

end

varname={'tlocal','tst','tfin','tfout','qst','qu','dtl*ml*c','n','q
l','qs','tlin','tlout','dc','dl','ta','ml','mc','tsrtat1','tstrat2'
,'tstrat3','tstrat4','tstrat5','tsttmp1','tsttmp2','tsttmp3','tsttm
p4','tsttmp5','omega','rrq'};
xlRange = 'A1:AC1';
xlswrite ('SOLSYS',varname,xlRange);
```

```
xlRange = 'A2:AC60000';
xlswrite ('SOLSYS',vec,xlRange);
xlswrite ('radiation',ve);
set(handles.fText,'String',num2str(f));

hax=handles.Tst_plot;
plot(hax,x,TST,x,QU);
xlabel(hax,'t (hours x 10)');
ylabel(hax,'Tst(K)');
grid(hax,'on');
end
```

Κεφάλαιο 4. Αποτελέσματα και πιστοποίηση πηγαίου κώδικα

Στο κεφάλαιο αυτό θα πιστοποιηθεί ο πηγαίος κώδικας Matlab που παρουσιάστηκε στο 3^ο κεφάλαιο. Αρχικά στη παράγραφο 4.1 θα γίνει μία σύγκριση με τον πηγαίο κώδικα Fortran του Εργαστηρίου Φυσικών και Χημικών Διεργασιών, με ορισμένες μετατροπές έτσι ώστε να εκτελούν υπολογισμούς με τις ίδιες παραμέτρους, στο βαθμό που αυτό είναι εφικτό. Με τον τρόπο αυτό θα βεβαιωθεί η ορθή λειτουργία και μεταγλώττιση των χωρίων που ήδη υπήρχαν από Fortran σε Matlab. Στη συνέχεια στη παράγραφο 4.2 θα συγκριθούν οι δύο μέθοδοι υπολογισμού της ακτινοβολίας που αναφέρθηκαν στο 2ο κεφάλαιο (παράγραφος 2.1.5). Αφού διαπιστώσουμε τις διαφορές που προκύπτουν, η βασική μέθοδος θα συγκριθεί με τα δεδομένα του κώδικα Fortran και στη συνέχεια με τον κώδικα του Κοινού Κέντρου Ερευνών της Ευρωπαϊκής Επιτροπής (JRC/ΚΚΕρ) και τον κώδικα Trnsys για διάφορες επιφάνειες και κλίσεις συλλέκτη αντίστοιχα. Επιπρόσθετα θα γίνει μία συνοπτική επαλήθευση της εκτίμησης των θερμοκρασιών νερού δικτύου και των ωρών ανατολής και δύσης που προκύπτουν από τον αλγόριθμο σε σχέση με τις πραγματικές από στοιχεία του ΤΕΕ. Θα ακολουθήσει στη παράγραφο 4.3 η πιστοποίηση του κώδικα Matlab σε ΗΘΣ για ζεστό νερό χρήσης. Καταλήγοντας στη παράγραφο 4.4 θα επιλυθεί ως παράδειγμα ΗΘΣ θέρμανσης στην Αθήνα. Ο κώδικας κατά την εκτέλεση του τελικού παραδείγματος θα βρίσκεται στην κανονική του μορφή χωρίς μετατροπές, όπως ακριβώς περιγράφηκε κατά τα κεφάλαια 2 και 3. Ιδιαίτερη έμφαση κατά την εκτέλεση θα δοθεί στις νέες προσθήκες του κώδικα που αναπτύχθηκε.

4.1 Πιστοποίηση κώδικα θέρμανσης και στατικού φορτίου

Για την πιστοποίηση γίνεται προσομοίωση ΗΘΣ θέρμανσης με τα εξής δεδομένα:

Ηλιακή ακτινοβολία και θερμοκρασία περιβάλλοντος

$$T_a(t) = \dot{T}_a + \Delta T_a \sin \frac{2\pi(t - (trise + 360))}{1440}$$

$$q_{sref} = 800 \text{ W/m}^2$$

$$q_s(t) = q_{sref} \sin \frac{\pi(t - trise)}{td} \text{ W/m}^2$$

$$t_{rise} = 7 \text{ π. μ.} = 420 \text{ min}$$

$$t_d = 10 \text{ h} = 600 \text{ min}$$

$$\dot{T}_a = 290 \text{ K}$$

$$\Delta T_a = 8 \text{ K}$$

Συλλέκτης

$$A = 30 \text{ m}^2$$

$$F_R = 0.95$$

$$b = 1.2$$

$$\dot{m}_C = 20 \text{ kg/min}$$

Δοχείο αποθήκευσης

$$U_{st} = 0 \text{ W}/(\text{m}^2 \text{ C})$$

$$T_{st,initial} = 300 \text{ K}$$

$$m_{st}C_p = 5000 \text{ kJ}/(\text{kJ K})$$

Ζήτηση

$$Q_L = 4.2 \text{ kW}$$

$$T_{L,in} \geq 300 \text{ K}$$

$$\dot{m}_L = 10 \text{ kg/min}$$

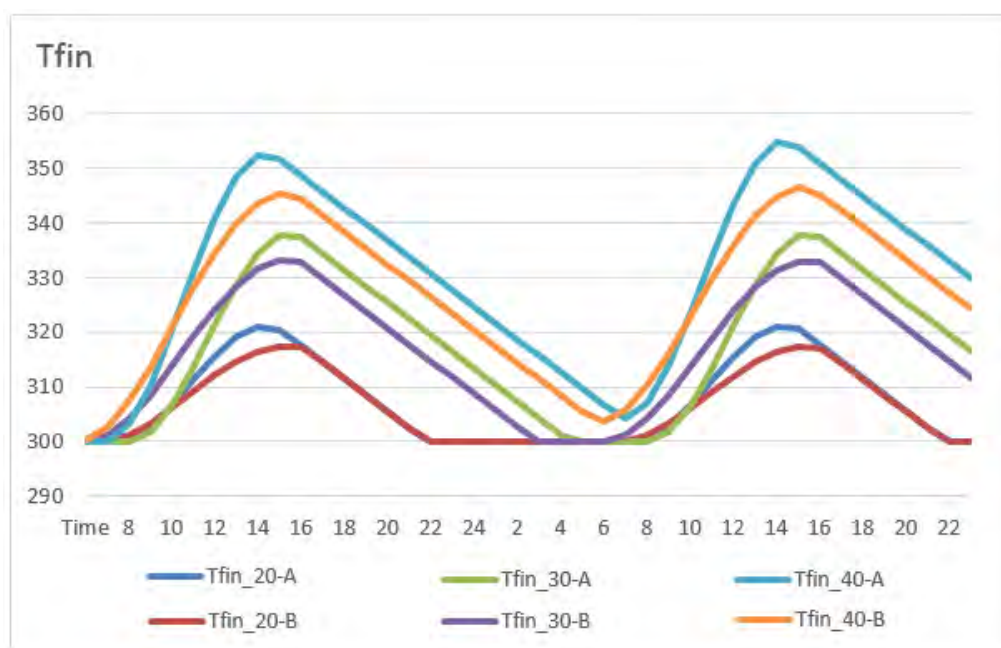
$$T_{st,max} \geq 373 \text{ K}$$

Τα δεδομένα του συλλέκτη, του δοχείου αποθήκευσης και της ζήτησης εισάγονται ακριβώς όπως στον κώδικα Fortran που μεταγλωττίστηκε. Το ίδιο συμβαίνει για την ακτινοβολία αναφοράς, τη θερμοκρασία περιβάλλοντος και τη διακύμανσή της. Η προσέγγιση της ακτινοβολίας κατά τη διάρκεια της ημέρας, όπως αναφέρθηκε σε προηγούμενα κεφάλαια, υπολογίζεται με διαφορετικό τρόπο στον αλγόριθμο Fortran, για αυτό διατηρούμε τον τρόπο υπολογισμού του αλγορίθμου Matlab που αναλύσαμε με ορισμένες μετατροπές. Αυτό συμβαίνει ρυθμίζοντας τον αλγόριθμο μας ώστε η ηλιοφάνεια των ημερών που θα προσομοιώσουμε να διαρκεί κατά προσέγγιση 10 ώρες. Με τον τρόπο οι ακτινοβολίες συγκλίνουν και επομένως οποιεσδήποτε διαφορές αφορούν τις ποσότητες που προκύπτουν από αυτές θα γίνουν αμέσως εμφανείς.

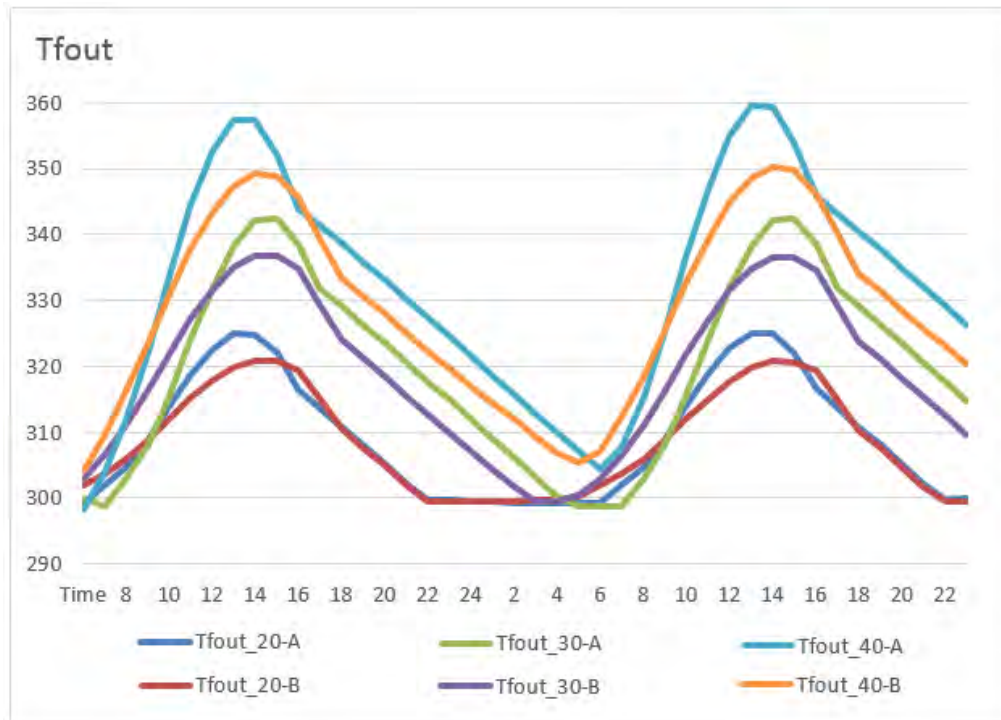
Τα αποτελέσματα των αλγορίθμων Fortran και Matlab εμφανίζονται στα γραφήματα που ακολουθούν. Οι λεζάντες με δείκτη A υποδεικνύουν τα στοιχεία του πιστοποιημένου αλγορίθμου Fortran, ενώ οι λεζάντες με δείκτη B υποδεικνύουν τις εκτιμήσεις του αλγορίθμου προσομοίωσης Matlab που αναλύουμε.

Πιστοποίηση βασικών λειτουργιών για επιφάνεια συλλέκτη $A=20, 30, 40 \text{ m}^2$

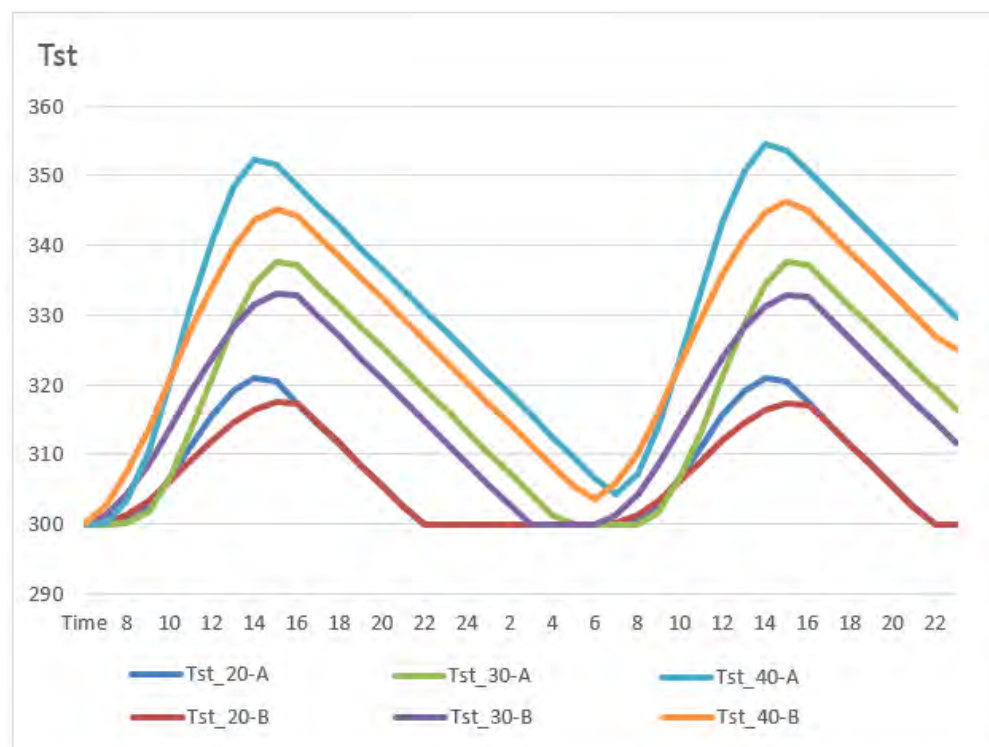
Εκτελώντας τον αλγόριθμο προκύπτουν τα παρακάτω αποτελέσματα στα Σχήματα 4.1 – 4.7.



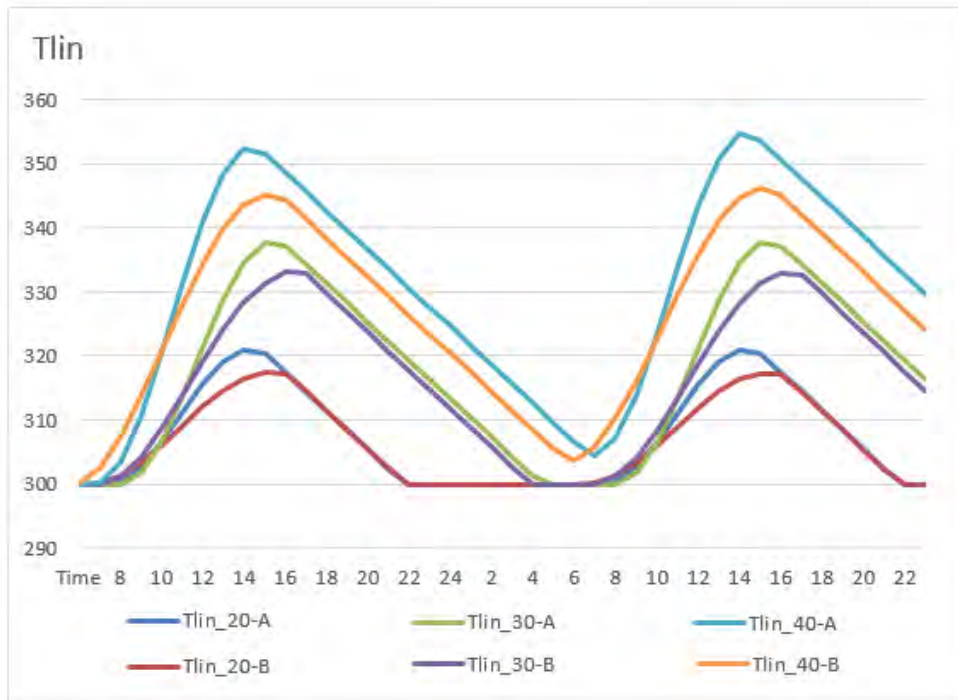
Σχήμα 4.1 Θερμοκρασία εισόδου του φέροντος ρευστού στο συλλέκτη για περίοδο δύο ημερών για επιφάνειες 20,30,40 m².



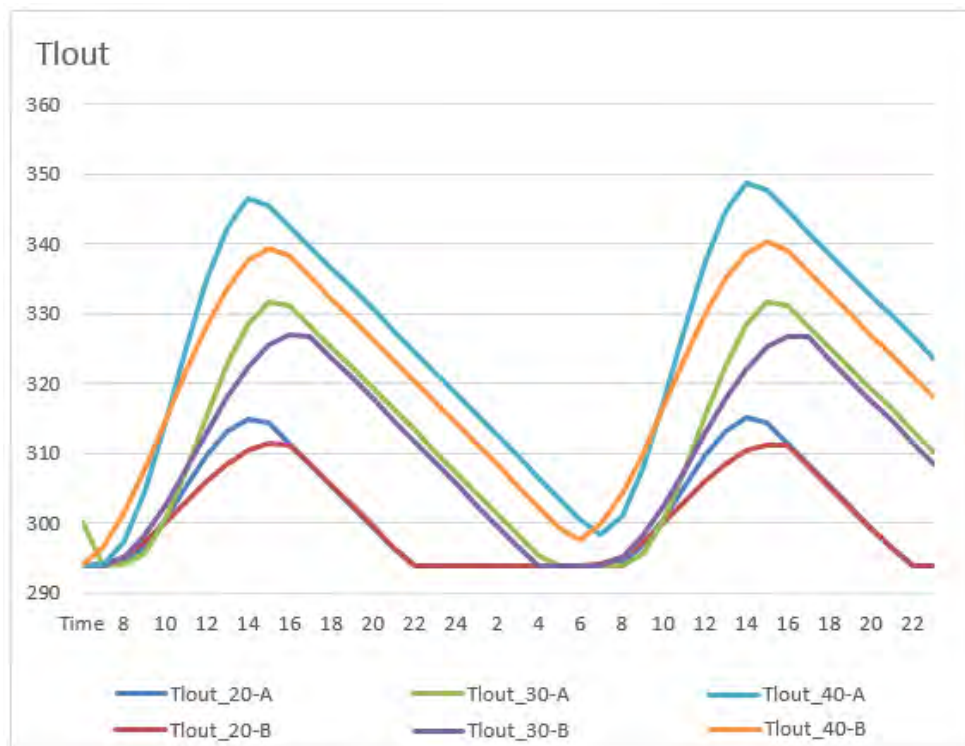
Σχήμα 4.2 Θερμοκρασία εξόδου του φέροντος ρευστού από το συλλεκτη για περίοδο δύο ημερών για επιφάνειες 20,30,40 m².



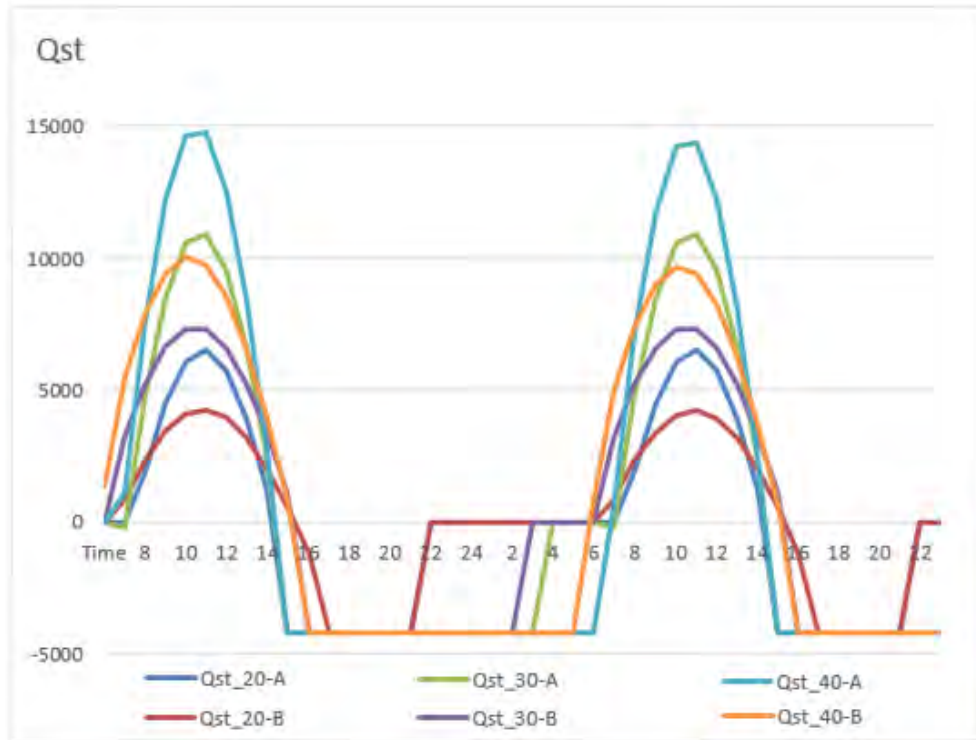
Σχήμα 4.3 Θερμοκρασία αποθήκευσης του φέροντος ρευστού για περίοδο δύο ημερών για επιφάνειες 20,30,40 m².



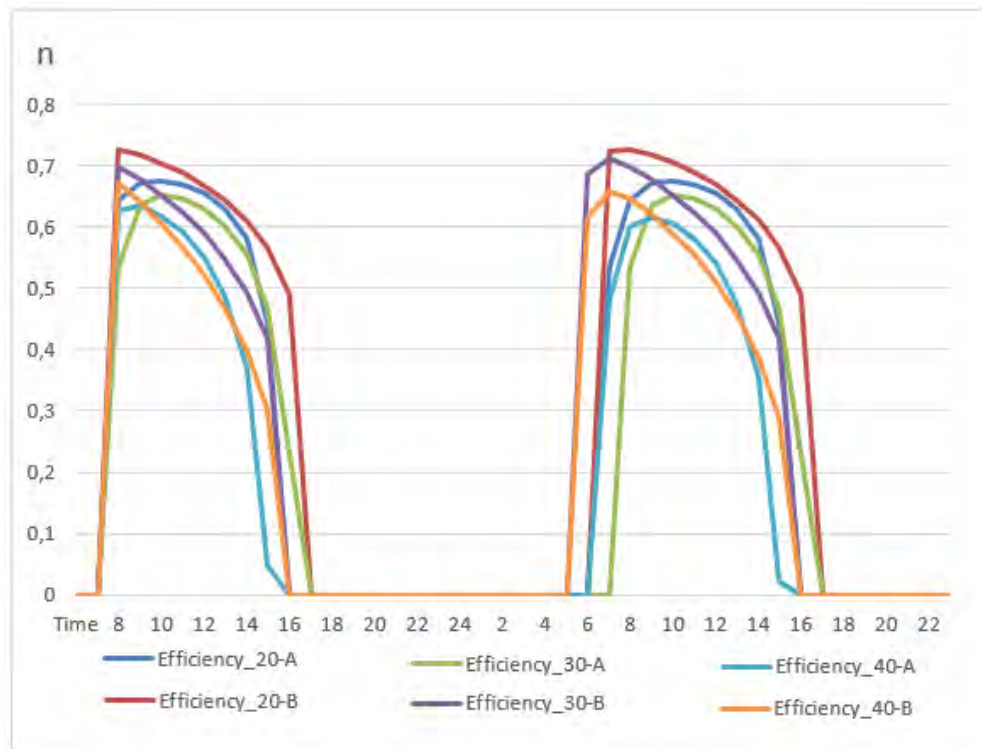
Σχήμα 4.4 Θερμοκρασία εισόδου του φέροντος ρευστού στην κατανάλωση για περίοδο δύο ημερών για επιφάνειες 20,30,40 m².



Σχήμα 4.5 Θερμοκρασία εξόδου του φέροντος ρευστού από την κατανάλωση για περίοδο δύο ημερών για επιφάνειες 20,30,40 m².



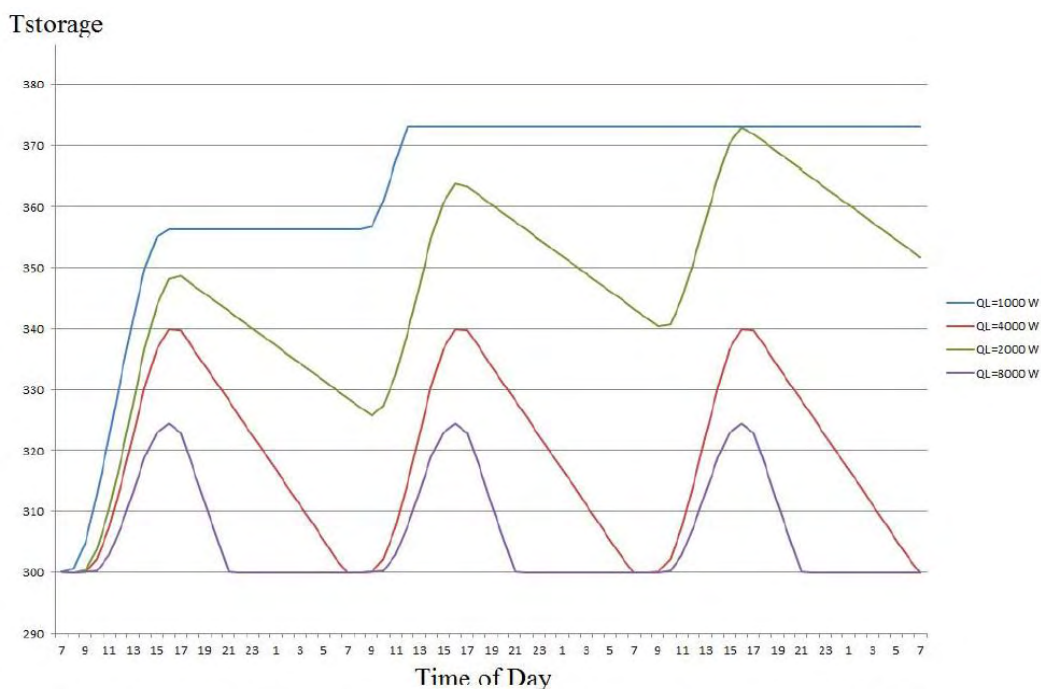
Σχήμα 4.6 Αποθηκευμένη ενέργεια στη δεξαμενή αποθήκευσης για περίοδο δύο ημερών για επιφάνεια 20,30,40 m².



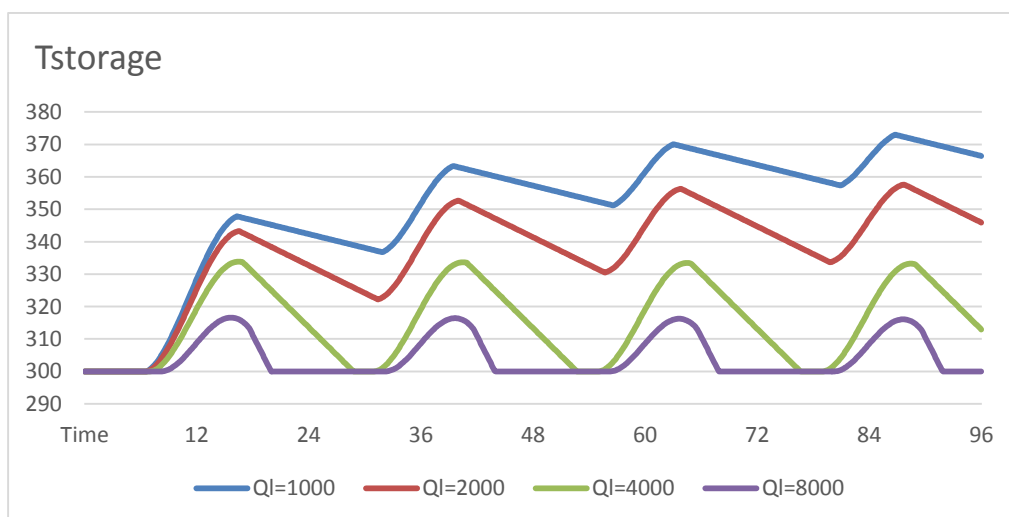
Σχήμα 4.7 Βαθμός απόδοσης του συστήματος για περίοδο δύο ημερών για επιφάνειες 20,30,40 m².

Σύγκριση στατικών φορτίων με αναφορά τη θερμοκρασία αποθήκευσης

Τα αποτελέσματα των αλγορίθμων Fortran και Matlab ως προς τη θερμοκρασία αποθήκευσης για διαφορετικά φορτία, με κοινή επιφάνεια $A=30 \text{ m}^2$ δίδονται στα Σχήματα 4.8 και 4.9 αντίστοιχα.



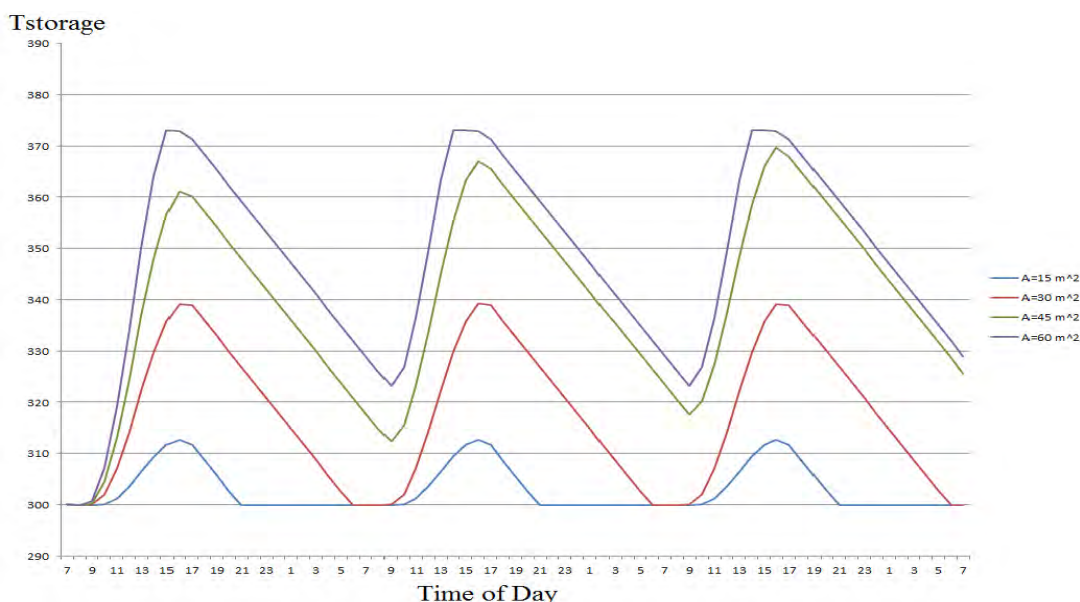
Σχήμα 4.8 Θερμοκρασία αποθήκευσης πιστοποιημένου αλγορίθμου Fortran ως προς το χρόνο για 3 ημέρες με θερμικό φορτίο 1000,2000,4000,8000 W.



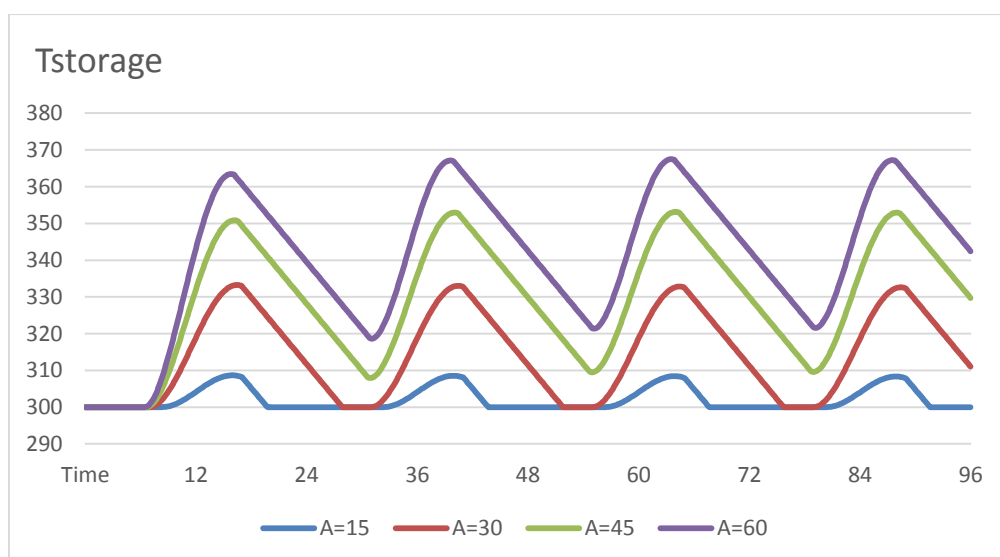
Σχήμα 4.9 Θερμοκρασία αποθήκευσης του αλγορίθμου Matlab ως προς το χρόνο για 4 ημέρες με θερμικό φορτίο 1000,2000,4000,8000 W.

Σύγκριση αποτελεσμάτων για επιφάνειες $A=15,30,45,60 \text{ m}^2$ με αναφορά τη θερμοκρασία αποθήκευσης

Αντίστοιχα, τα αποτελέσματα των αλγορίθμων Fortran και Matlab ως προς τη θερμοκρασία αποθήκευσης για τέσσερις διαφορετικές επιφάνειες συλλέκτη παρουσιάζονται στα Σχήματα 4.10 και 4.11 αντίστοιχα.

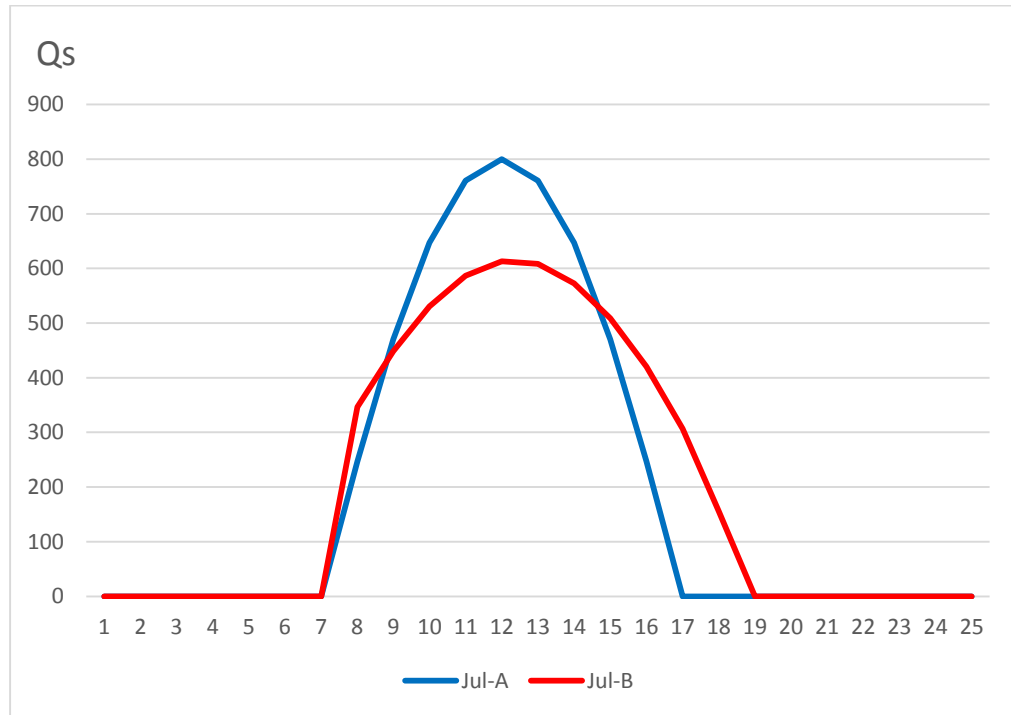


Σχήμα 4.10 Θερμοκρασία αποθήκευσης αλγορίθμου Fortran ως προς το χρόνο για 3 ημέρες με $A=15,30,45,60 \text{ m}^2$.



Σχήμα 4.11 Θερμοκρασία αποθήκευσης του αλγορίθμου Matlab ως προς το χρόνο για 4 ημέρες με $A=15,30,45,60 \text{ m}^2$.

Στο Σχήμα 4.12 παρατίθεται η ακτινοβολία των αλγορίθμων Fortran και Matlab.



Σχήμα 4.12 Σύγκριση ακτινοβολιών μεταξύ των δύο αλγορίθμων για 2 ημέρες του Ιουλίου.

Χρησιμοποιώντας τον κανόνα του τραπεζίου

$$\int_a^b f(x)dx \approx 1/2 \sum_{n=1}^N (x_{n+1} - x_n) [(f(x_n) + f(x_{n+1}))]$$

Υπολογίζονται οι εξής ολικές και μέσες ποσότητες ακτινοβολίας:

Case	Case A	Case B
total radiation	5051	5098
hours of sunshine	10	11
radiation/hour of sunshine	505.1	463.4545455
radiation/sec	0.140305556	0.128737374

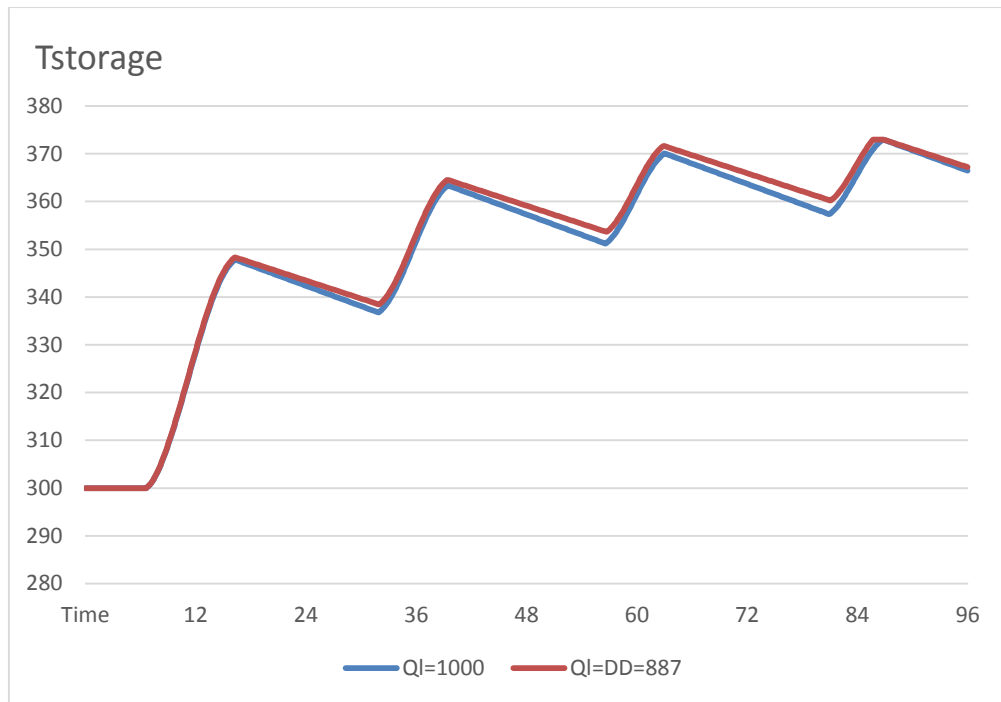
Εξετάζοντας τις πέντε δοκιμές που εκτελέστηκαν παρατηρούμε πως σε όλες τις περιπτώσεις, για τα γραφήματα που καταλήξαμε, ο αλγόριθμος μας ακολουθεί πιστά την κατανομή του αλγορίθμου αναφοράς και επιβεβαιώνεται η ομαλή λειτουργία του για τις τιμές T_{fin} , T_{out} , T_{lin} , T_{lout} , T_{st} , n , και Q_{st} . Παρόλα αυτά οι τιμές που προσομοιώνονται παρουσιάζουν μικρές αποκλίσεις και είναι πιο συντηρητικές. Οι

αποκλίσεις αυτές φαίνονται να είναι πιο έντονες όσο η κλίση του συλλέκτη μεγαλώνει. Η μεγαλύτερη διαφορά εμφανίζεται στο γράφημα της αποθηκευμένης ενέργειας που συμβολίσαμε με τη μεταβλητή Q_{st} . Εξετάζοντας τα διαφορετικά φορτία και για τις διαφορετικές επιφάνειες οι καμπύλες φαίνονται να μεταβάλλονται ομαλά και σύμφωνα με τη βιβλιογραφία. Παρατηρείται απόκλιση από 1 έως 7 βαθμών, ανάλογα με την επιφάνεια του συλλέκτη. Οι αποκλίσεις αυτές μπορούν να εξηγηθούν μέσω του διαφορετικού τρόπου υπολογισμού της ακτινοβολίας και του μικρού χρονικού βήματος υπολογισμού, όπως θα δούμε αναλυτικότερα στην επόμενη παράγραφο. Μία ακόμη παράμετρος η οποία διαφοροποιεί τα γραφήματα, είναι ότι ο αλγόριθμος Fortran ορίζει την έναρξη της ημέρας στις 7 π.μ., ενώ ο αλγόριθμος Matlab στις 12 π.μ.

Συγκρίνοντας τις ακτινοβολίες πρόκύπτει ότι ο αλγόριθμος Matlab χρειάζεται μία επιπλέον ώρα για το επιθυμητό αποτέλεσμα των 5000 W. Αυτό σημαίνει πως ο ρυθμός πρόσπτωσης της ακτινοβολίας του αλγορίθμου Fortran είναι μεγαλύτερος σε σχέση με αυτόν του αλγορίθμου Matlab. Η σύγκριση του διαγράμματος κάνει ξεκάθαρη την διαφορά στο θερμικό φορτίο Q_{st} , αφού η κορυφή του γραφήματος του αλγορίθμου της πιστοποίησης βρίσκεται πολύ υψηλότερα από ότι του αλγορίθμου της παρούσας εργασίας, ενώ η καμπύλη γίνεται φθίνουσα σε συντομότερο χρονικό διάστημα. Επιπρόσθετα η διαφορά που εμφανίζουν οι κορυφές των δύο γραφημάτων της ακτινοβολίας εξηγούν τις αποκλίσεις στις θερμοκρασίες που εμφανίστηκαν στα προηγούμενα αποτελέσματα.

Πιστοποίηση βαθμομερών

Αφού πιστοποιήθηκαν τα αποτελέσματα του αλγορίθμου Matlab για τα διάφορα θερμικά φορτία και διάφορες επιφάνειες συλλέκτη επαληθεύονται και τα φορτία ζήτησης για τις βαθμομερές. Το μόνο που διαφέρει κατά τη λειτουργία των βαθμομερών στην εκτέλεση του αλγορίθμου είναι το θερμικό φορτίο, επομένως ενεργοποιείται ο διακόπτης βαθμομερών για το μήνα Φεβρουαρίου. Το θερμικό φορτίο που υπολογίζει είναι για τις 2/1/2017 για τη θέρμανση μίας κατοικίας στην Αθήνα (Σχήμα 4.13) και επαληθεύεται με σύγκριση από το γράφημα των θερμοκρασιών αποθήκευσης που ελέγχθηκε νωρίτερα.



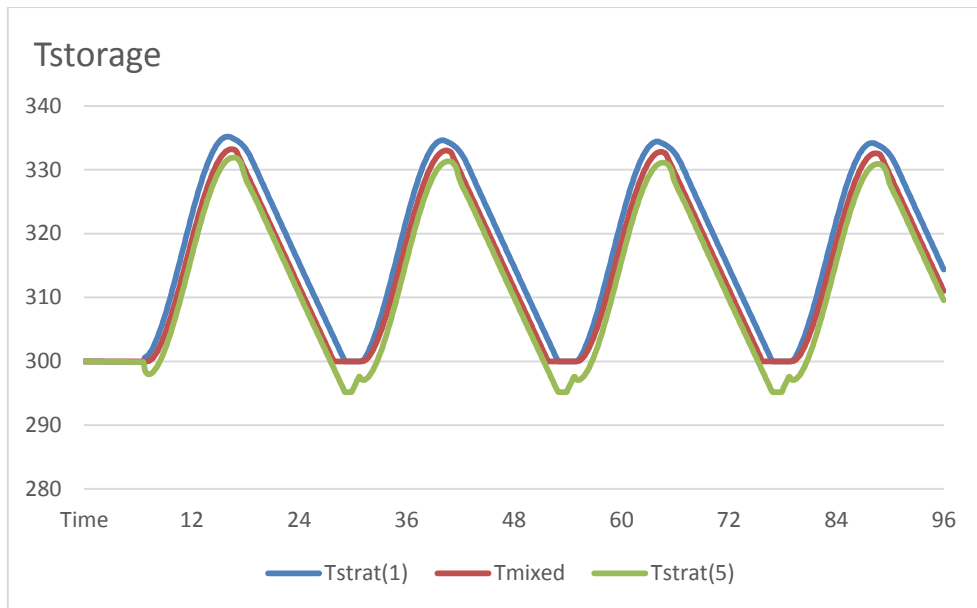
Σχήμα 4.13 Θερμοκρασία αποθήκευσης ως προς το χρόνο για 4 ημέρες με σταθερό θερμικό φορτίο 1000 W και φορτίο 887 W για τις βαθμοημέρες.

Όπως είναι εμφανές από το Σχήμα 4.13 η θερμοκρασία στην περίπτωση των βαθμοημερών, όπου η ζήτηση είναι χαμηλότερη από αυτή της προσομοίωσης σταθερού φορτίου, η θερμοκρασία αποθήκευσης παραμένει υψηλότερη. Η κατανομή των θερμοκρασιών και η συμπεριφορά του αλγορίθμου δεν αλλοιώνεται από αυτή τη λειτουργία.

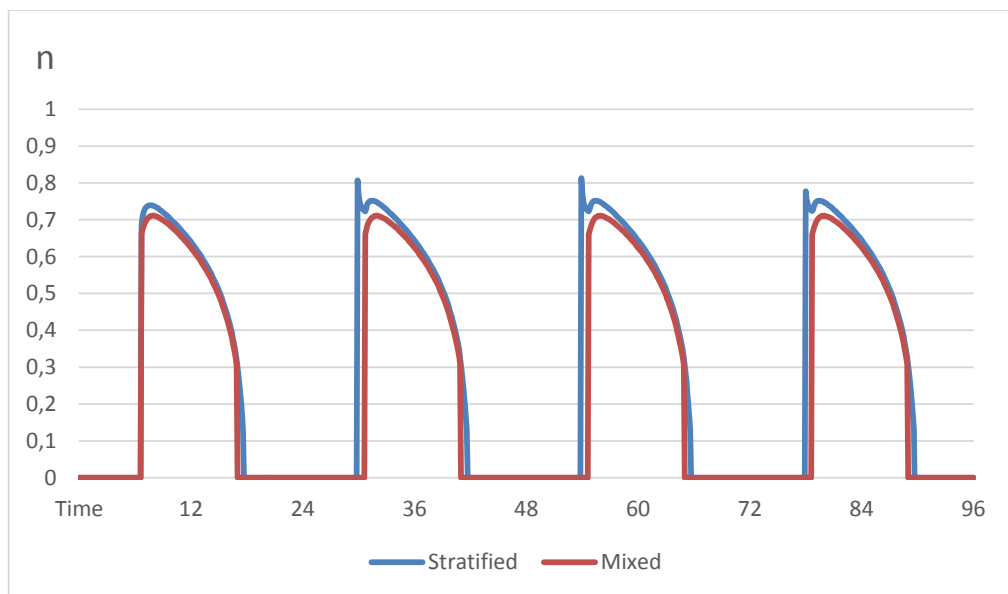
Σύγκριση της διαστρωμάτωσης

Για τη σύγκριση της διαστρωμάτωσης ενεργοποιείται ο αντίστοιχος διακόπτης και στη συνέχεια το πρόγραμμα υπολογίζει την επιφάνεια της δεξαμενής, η οποία προκύπτει να είναι δέκα τετραγωνικά μέτρα.

Όπως φαίνεται στο Σχήμα 4.14, η πρώτη στοιβάδα, στην κορυφή της δεξαμενής βρίσκεται στην κορυφή του γραφήματος ενώ η τελευταία στιβάδα, που βρίσκεται στον πάτο της δεξαμενής, στο κάτω άκρο του. Η μη διαστρωματοποιημένη δεξαμενή βρίσκεται μεταξύ των δύο.



Σχήμα 4.14 Θερμοκρασία αποθήκευσης ως προς το χρόνο για 4 ημέρες με ή χωρίς διαστρωμάτωση.

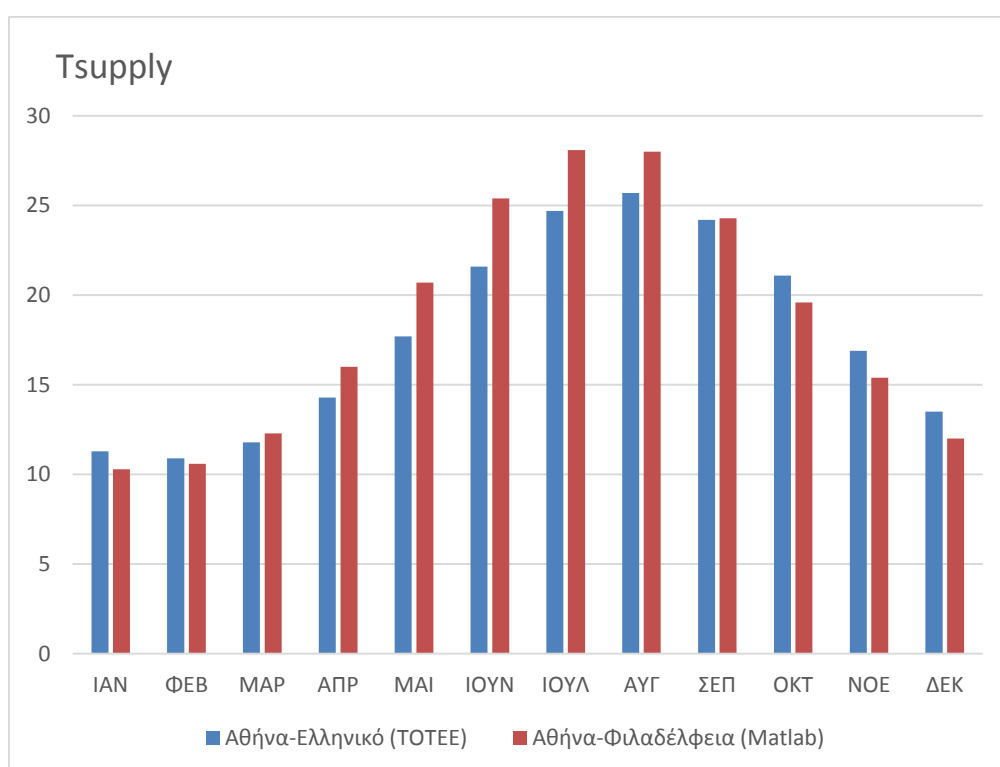


Σχήμα 4.15 Βαθμός απόδοσης για τα δύο είδη δεξαμενών για προσομοίωση 4 ημερών.

Άλλη μία παράμετρος που αλλάζει με τη χρήση της διαστρωμάτωσης είναι ο βαθμός απόδοσης, ο οποίος όπως φαίνεται και στο Σχήμα 4.15, στην περίπτωση της διαστρωματοποιημένης δεξαμενής είναι υψηλότερος.

Σύγκριση αποτελεσμάτων νερού παροχής δικτύου

Μία ακόμη παράμετρος η οποία πρέπει να ελεγχθεί είναι η θερμοκρασία παροχής νερού δικτύου. Συγκρίνονται τα αποτελέσματα της εκτίμησης του αλγορίθμου Matlab που αναλύθηκε στην παράγραφο 2.1.7 με ένα τυπικό συντελεστή θερμικής διάχυσης θερμοκρασίας εδάφους, για την πόλη της Αθήνας στην περιοχή της Φιλαδέλφειας με την περιοχή του Ελληνικού σύμφωνα με τις πραγματικές τιμές του δικτύου όπως δίδονται από ΤΟΤΕΕ. Για τη σύγκριση χρησιμοποιείται ο συντελεστής θερμικής διάχυσης του Ψαμμίτη, ενώ ως ψυχρότερη ημέρα του έτους λαμβάνεται η 30η Ιανουαρίου.

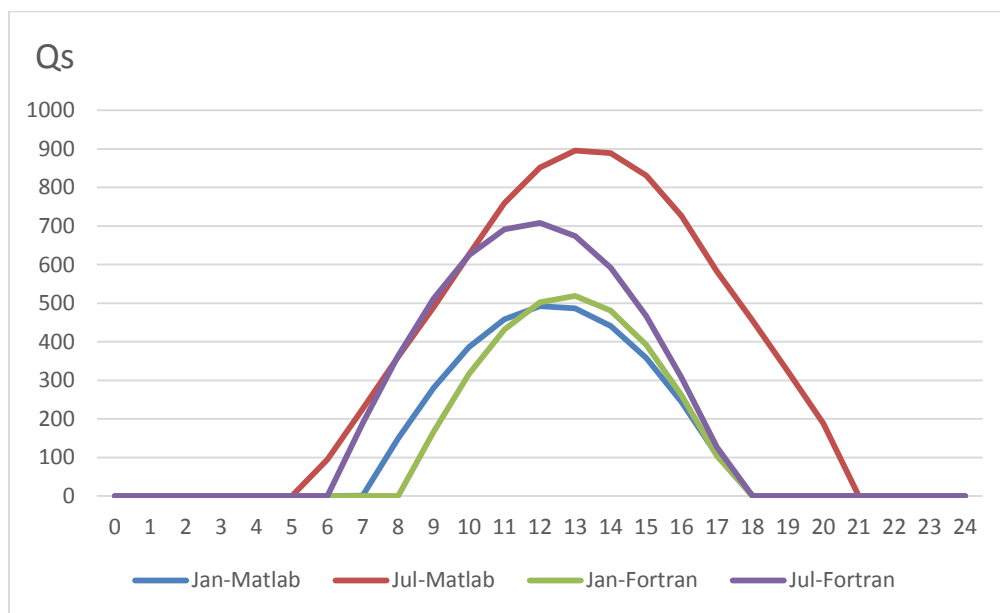


Σχήμα 4.16 Σύγκριση θερμοκρασιών νερού παροχής δικτύου.

Όπως φαίνεται στο Σχήμα 4.16 οι δύο θερμοκρασίες ακολουθούν την ίδια κατανομή κατά τη διάρκεια του έτους. Η εκτίμηση της θερμοκρασίας φαίνεται να είναι πιο αισιόδοξη για τους καλοκαιρινούς μήνες, με μέγιστη διαφορά τον Ιούνιο με 3.8 βαθμούς, ενώ πιο απαισιόδοξη το χειμώνα με μέγιστη διαφορά 1.5 βαθμούς για τους μήνες Οκτώβρη, Νοέμβρη και Δεκέμβρη.

4.2 Σύγκριση μεθόδων υπολογισμού ακτινοβολιών

Ο υπολογισμός της ακτινοβολίας του αλγορίθμου, γίνεται όπως περιγράφηκε στη παράγραφο 2.1.5, ενώ οι αλγόριθμοι που χρησιμοποιήθηκαν για τις δοκιμές εφαρμόζουν την ημιτονοειδή κατανομή που παρατέθηκε νωρίτερα. Εδώ, γίνεται ανασκόπηση της ακτινοβολίας που λαμβάνουμε από την κάθε προσέγγιση, για τους μήνες Ιανουάριο και Ιούλιο με ακτινοβολίες αναφοράς 520 W/m^2 και 710 W/m^2 αντίστοιχα (Σχήμα 4.17).



Σχήμα 4.17 Ωριαία ολική ηλιακή ακτινοβολία σε κεκλιμένη επιφάνεια για τον Ιανουάριο και Ιούλιο σύμφωνα με τους αλγορίθμους Matlab και Fortran.

Χρησιμοποιώντας τον κανόνα του τραπεζιού προκύπτουν οι εξής ολικές και μέσες ποσότητες ακτινοβολίας:

Case	Jan-Matlab	Jul-Matlab	Jan-Fortran	Jul-Fortran
total radiation	3405.1	8301.4	3175.3	5255.2
hours of sunshine	10	14	9	11
radiation/hour of sunshine	340.51	592.9571429	352.8111111	477.7454545
radiation/sec	0.094586111	0.164710317	0.098003086	0.132707071
Jan-Matlab/Jan-Fortran	0.965134003			
Jul-Matlab/Jul-Fortran	1.241157058			

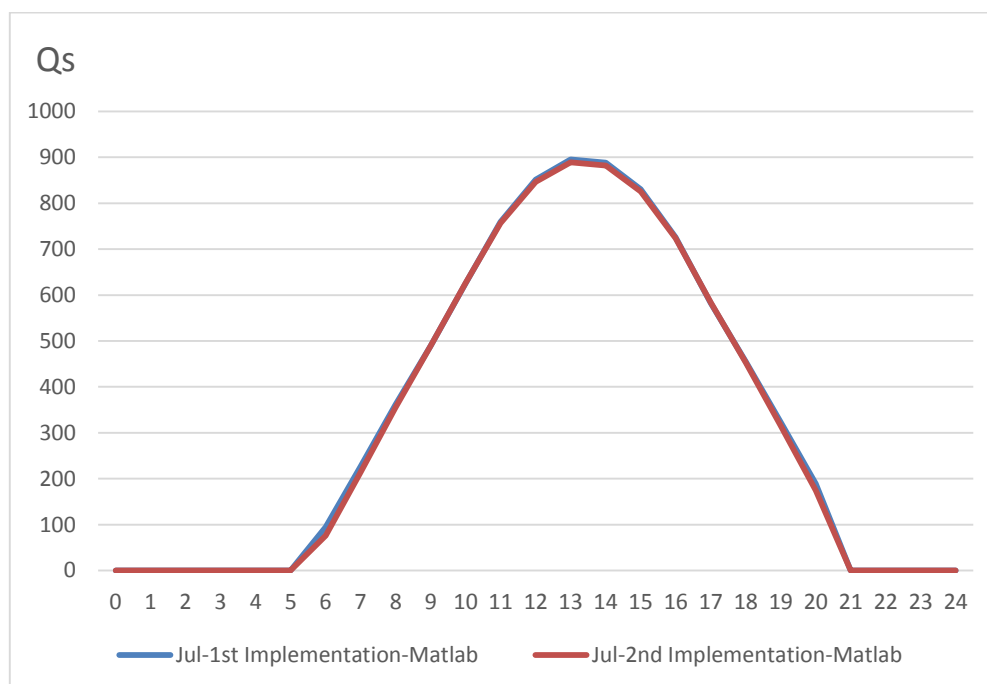
Σύμφωνα με τον Πίνακα 4.1 και τα αποτελέσματα σύγκρισης ο αλγόριθμος με τις τιμές που εισάγονται ως ακτινοβολία αναφοράς παρουσιάζει διαφορές με αυτόν της βιβλιογραφίας. Το χειμώνα οι τιμές συγκλίνουν αρκετά, ενώ το καλοκαίρι η απόκλιση είναι εμφανής. Η πρώτη διαφορά που παρατηρείται είναι αυτή των ωρών ακτινοβολίας.

Κατά την προσομοίωση του Ιανουαρίου η έναρξη της ακτινοβολίας στον αλγόριθμό μας προηγείται κατά μία ώρα. Κατά την προσομοίωση του Ιουλίου προηγείται μία ώρα ενώ λήγει τρεις ώρες αργότερα. Τα αποτελέσματα σε αυτή την δοκιμή του αλγορίθμου φαίνονται απαισιόδοξα, όμως αν συγκρίνει κανείς την διάρκεια της ημέρας τους καλοκαιρινούς μήνες με τον παρακάτω πίνακα, που περιέχει πραγματικές τιμές, θα διαπιστώσει πως ανταποκρίνονται στην πραγματική διάρκεια, η οποία κυμαίνεται στις 14 ώρες για τον μήνα Ιούλιο. Αξίζει επίσης να αναφερθεί πως η ώρα έναρξης, λήξης αλλά και διάρκεια της ημέρας και για τους δύο μήνες προσομοίωσης σχεδόν ταυτίζεται με τις πραγματικές ώρες. Το αποτέλεσμα αυτό πιστοποιεί τη διάρκεια, το χρονικό βήμα λειτουργίας και τις γωνίες που χρησιμοποιήσαμε για τους υπολογισμούς.

Πίνακας 4.1 Ώρες ανατολής και δύσης του ηλίου για την περιοχή της Ελλάδας κατά τη διάρκεια του έτους.

ΜΗΝΑΣ	ΑΝΑΤΟΛΗ	ΔΥΣΗ	ΔΙΑΡΚΕΙΑ ΗΜΕΡΑΣ
1 Ιανουαρίου	07.42	17:16	9ωρ.34'
15 Ιανουαρίου	07:40	17:28	9ωρ.48'
1 Φεβρουαρίου	07:30	17:47	10ωρ.07'
15 Φεβρουαρίου	07:17	18:02	10ωρ.45'
1 Μαρτίου	06:57	18:17	11ωρ.20'
15 Μαρτίου	06:36	18:32	11ωρ.56'
1 Απριλίου	(06:11) 07:11	(18:48) 19:48	12ωρ.37'
15 Απριλίου	(05:50) 06:50	(19:01) 20:01	13ωρ.11'
1 Μαΐου	(05:30) 06:30	(19:15) 20:15	13ωρ.45'
15 Μαΐου	(05:15) 06:15	(19:28) 20:28	14ωρ.13'
1 Ιουνίου	(05:04) 06:04	(19:42) 20:42	14ωρ.38'
15 Ιουνίου	(05:01) 06:01	(19:49) 20:49	14ωρ.48'
1 Ιουλίου	(05:05) 06:05	(19:52) 20:52	14ωρ.47'
15 Ιουλίου	(05:14) 06:14	(19:47) 20:47	14ωρ.33'
1 Αυγούστου	(05:27) 06:27	(19:35) 20:35	14ωρ.08'
15 Αυγούστου	(05:39) 06:39	(19:19) 20:19	13ωρ.40'
1 Σεπτεμβρίου	(05:55) 06:55	(18:54) 19:54	12ωρ.59'
15 Σεπτεμβρίου	(06:06) 07:06	(18:32) 19:32	12ωρ.26'
1 Οκτωβρίου	(06:20) 07:20	(18:08) 19:08	11ωρ.48'
15 Οκτωβρίου	(06:34) 07:34	(17:47) 18:47	11ωρ.13'
1 Νοεμβρίου	06:51	17:26	10ωρ.35'
15 Νοεμβρίου	07:06	17:13	10ωρ.07'
1 Δεκεμβρίου	07:22	17:05	9ωρ.43'
15 Δεκεμβρίου	07:35	17:07	9ωρ.32'

Συγκρίνοντας τους δύο τρόπους υπολογισμού ακτινοβολίας του αλγορίθμου της εργασίας της παραγράφου 2.1.5 προκύπτει το γράφημα του Σχήματος 4.18.



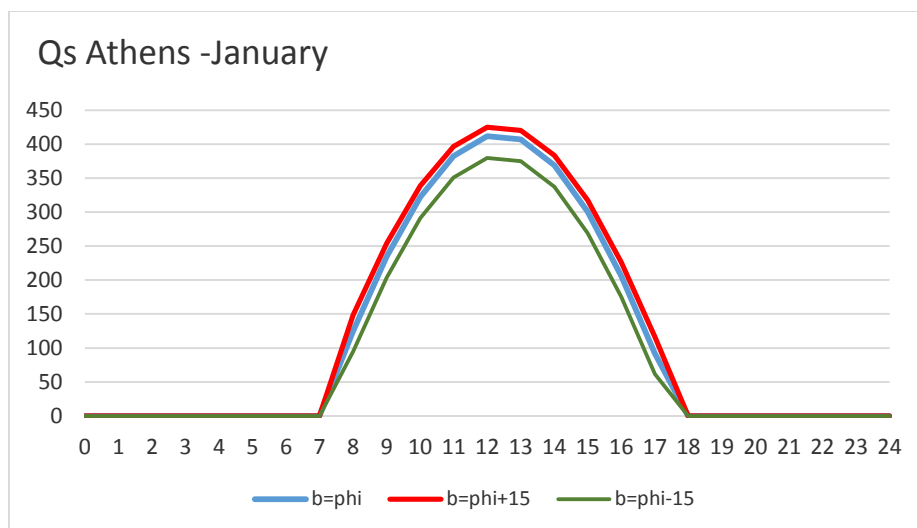
Σχήμα 4.18 Ακτινοβολία κατά τη διάρκεια της ημέρας για τα δεδομένα του παραδείγματος σύμφωνα με τον πρώτο και δεύτερο τρόπο υπολογισμού της ακτινοβολίας του αλγορίθμου.

Case	Jan-1st Implementation	Jan-2nd Implementation
total radiation	8301.4	8213.6
hours of sunshine	14	14
radiation/hour of sunshine	592.9571429	586.6857143
radiation/sec	0.164710317	0.162968254
Jan-1st Implementation/Jan-2nd Implementation	1.010689588	

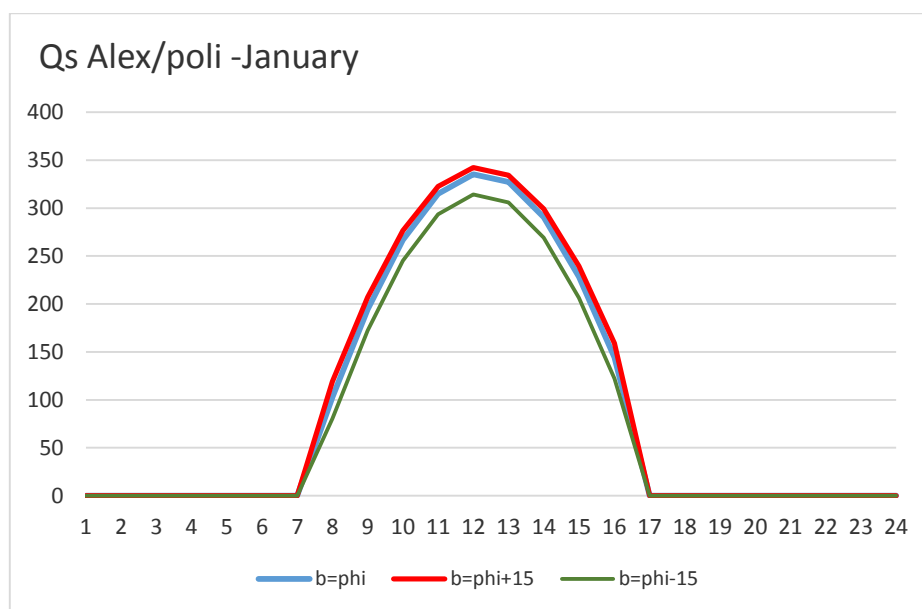
Είναι εμφανές ότι οι δύο εκτιμήσεις σχεδόν ταυτίζονται τόσο για τη διάρκεια της ημέρας, όσο και για την κατανομή της ακτινοβολίας. Από τον πίνακα των αποτελεσμάτων προκύπτει ότι η βασική εκτίμηση της ακτινοβολίας είναι πιο αισιόδοξη κατά 1%.

Αποτελέσματα ακτινοβολίας υπό διαφορετική κλίση συλλέκτη

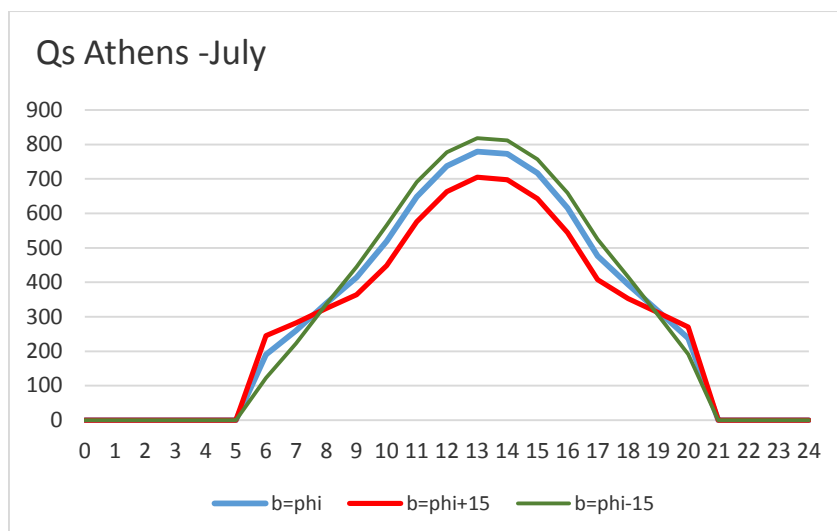
Εξετάζεται η μεταβολή της ακτινοβολίας σύμφωνα με την κλίση του συλλέκτη για τον αλγόριθμο της εργασίας. Θα εξεταστούν τρεις γωνίες ανάλογες με την τοποθεσία υπό μελέτη, για τις πόλεις της Αθήνας και Αλεξανδρούπολης. Οι προκύπτουσες ακτινοβολίες φαίνονται στα Σχήματα 4.19 – 4.22.



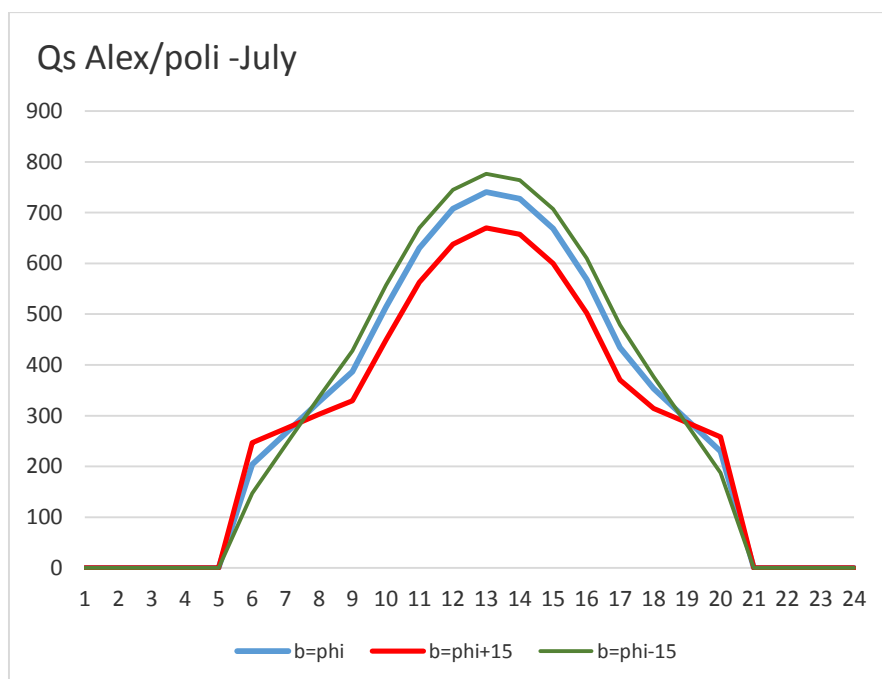
Σχήμα 4.19 Ακτινοβολία στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιανουαρίου για γωνίες $\beta=\varphi$, $\beta= \varphi+15$, $\beta=\varphi-15$.



Σχήμα 4.20 Ακτινοβολία στην περιοχή της Αλεξανδρούπολης για την πρώτη ημέρα του Ιανουαρίου για γωνίες $\beta=\varphi$, $\beta= \varphi+15$, $\beta=\varphi-15$.



Σχήμα 4.21 Ακτινοβολία στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιουλίου για γωνίες $\beta=\varphi$, $\beta= \varphi+15$, $\beta=\varphi-15$.

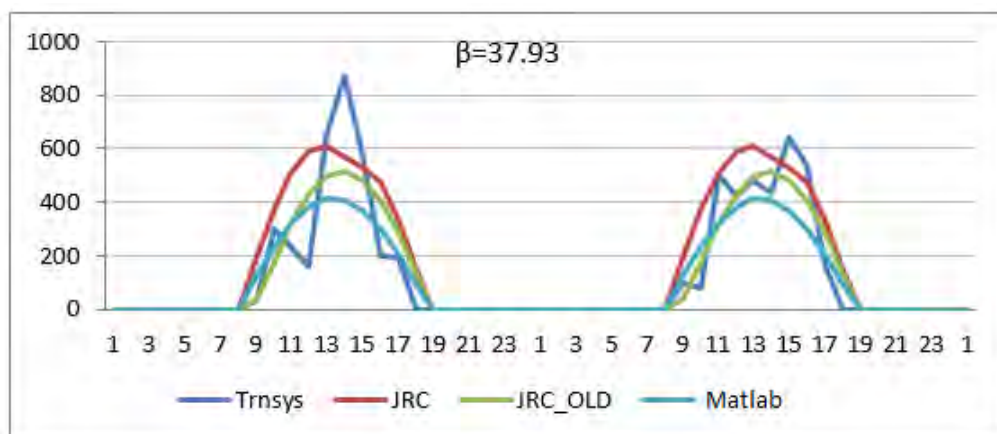


Σχήμα 4.22 Ακτινοβολία στην περιοχή της Αλεξανδρούπολης για την πρώτη ημέρα του Ιουλίου για γωνίες $\beta=\varphi$, $\beta= \varphi+15$, $\beta=\varphi-15$.

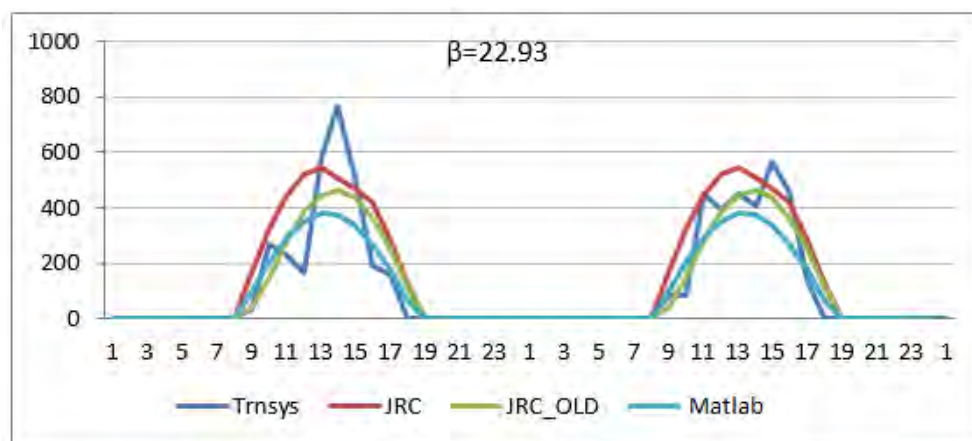
Παρατηρούμε ότι η μεγαλύτερη κλίση συλλέκτη ευνοεί τους χειμερινούς μήνες, ενώ η μικρότερη τους καλοκαιρινούς. Η σταδιακή αύξηση ή μείωση της ακτινοβολίας στο συλλέκτη με την κλίση του αποδεικνύει την ομαλή συμπεριφορά του συστήματος. Η απότομη έναρξη της ακτινοβολίας που εμφανίζεται ωφελείται στην κάθετη κλίση του συλλέκτη που ευνοεί την πρόσληψη της ακτινοβολίας τις πρωινές και βραδινές ώρες που ο ήλιος βρίσκεται χαμηλά στον ορίζοντα.

Σύγκριση αποτελεσμάτων με αλγόριθμους του διαδικτύου.

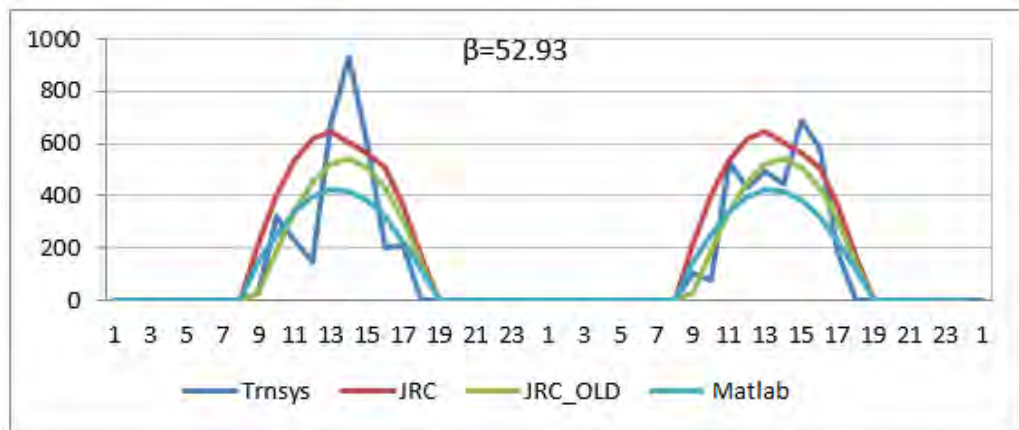
Χρησιμοποιώντας τα αποτελέσματα της προηγούμενης δοκιμής για την περιοχή της Αθήνας συγκρίνεται ο αλγόριθμος με δεδομένα από το διαδικτυακό χώρο του Κοινού Κέντρου Ερευνών της Ευρωπαϊκής Επιτροπής (JRC/ΚΚΕρ) [20] από το 1960 - 1990 και από το 1998 - 2010, όπως επίσης και με το πρόγραμμα προσομοίωσης Trnsys. Τα αποτελέσματα συνοψίζονται στα Σχήματα 4.23 - 4.28.



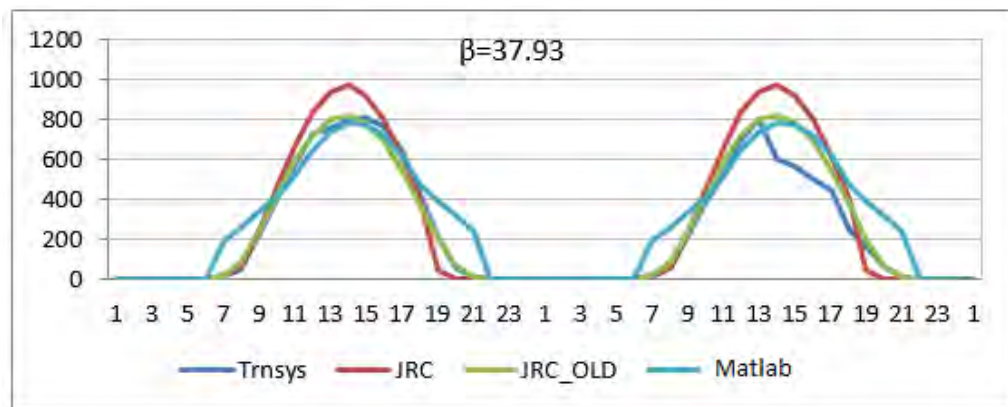
Σχήμα 4.23 Ακτινοβολία αλγορίθμου Matlab στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιανουαρίου για γωνία $\beta=\varphi$ σε σχέση με Trnsys και JRC.



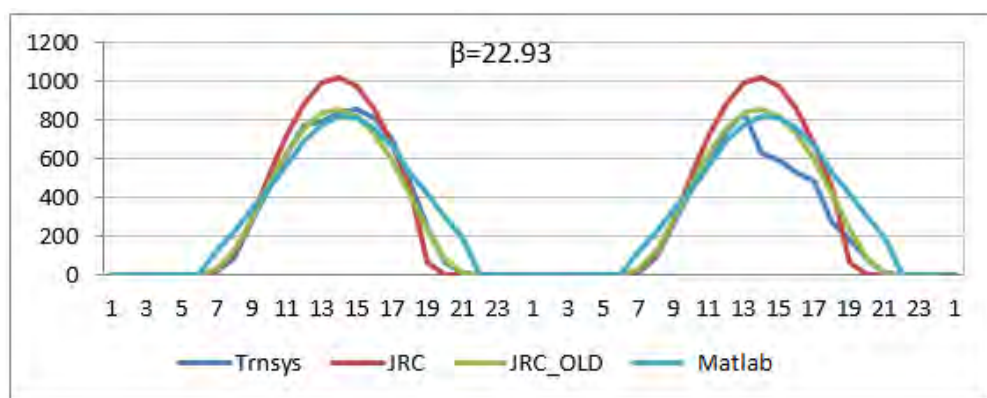
Σχήμα 4.24 Ακτινοβολία αλγορίθμου Matlab στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιανουαρίου για γωνία $\beta=\varphi-15$ σε σχέση με Trnsys και JRC.



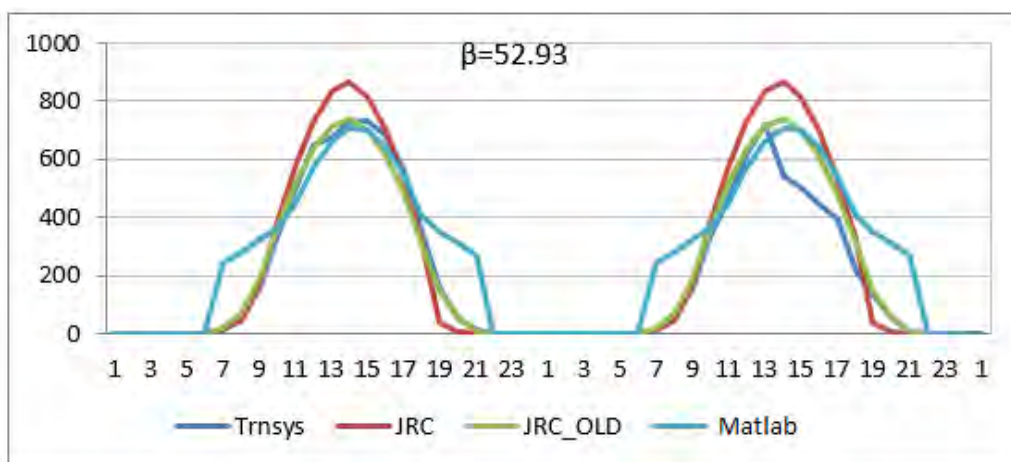
Σχήμα 4.25 Ακτινοβολία αλγορίθμου Matlab στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιανουαρίου για γωνία $\beta = \varphi + 15$ σε σχέση με Trnsys και JRC.



Σχήμα 4.26 Ακτινοβολία αλγορίθμου Matlab στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιουλίου για γωνία $\beta = \varphi$ σε σχέση με Trnsys και JRC.



Σχήμα 4.27 Ακτινοβολία αλγορίθμου Matlab στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιουλίου για γωνία $\beta = \varphi - 15$ σε σχέση με Trnsys και JRC.



Σχήμα 4.28 Ακτινοβολία αλγορίθμου Matlab στην περιοχή της Αθήνας για την πρώτη ημέρα του Ιουλίου για γωνία $\beta = \varphi + 15$ σε σχέση με Trnsys και JRC.

Συγκρίνοντας με το Trnsys, παρατηρείται ότι οι τιμές είναι πολύ πιο κοντά στον κώδικα Matlab, ενώ οι τιμές του Κοινού Κέντρου Ερευνών της Ευρωπαϊκής Επιτροπής εμφανίζονται πολύ υψηλότερες. Η διαφορά σύμφωνα με το ΚΚΕρ οφείλεται στην κλιματική αλλαγή και σε λάθη των προηγούμενων μετρήσεων [21]. Αξίζει να σημειωθεί ότι σε όλες τις περιπτώσεις οι ώρες έναρξης και λήξης της ακτινοβολίας φαίνονται να είναι σύμφωνες. Μία ακόμη διαφορά που γίνεται εμφανής από τα διαγράμματα είναι ότι το Trnsys προσπαθεί να προσομοιώσει την ακτινοβολία της ημέρας με ρεαλιστικό τρόπο, ενώ ο κώδικας του ΚΚΕρ χρησιμοποιεί ομαλή κατανομή.

Τον Ιανουάριο ο κώδικας Matlab υστερεί στη μέγιστη τιμή ενώ τον Ιούλιο φαίνεται να ξεκινά απότομα σε σχέση με τις υπόλοιπες λύσεις για μεγάλες κλίσεις. Ο λόγος που αυτό συμβαίνει είναι πιθανώς ο διαφορετικός συντελεστής αντανάκλασης ακτινοβολίας του περιβάλλοντος, η αδυναμία της εξίσωσης προσομοίωσης για μεγάλες κλίσεις, ο τρόπος διαχείρισης των στοιχείων των πινάκων από το TEE, ο τρόπος ορισμού έναρξης και λήξης της ακτινοβολίας ή ένας συνδυασμός όλων των παραπάνω.

4.3 Πιστοποίηση κώδικα ZNX και Δυναμικού Φορτίου

Για την πιστοποίηση του κώδικα Matlab εφαρμόζεται ο κώδικας Fortran για προσομοίωση ΗΘΣ ζεστού νερού χρήσης με τα εξής δεδομένα:

Ηλιακή ακτινοβολία και θερμοκρασία περιβάλλοντος

$$T_a(t) = \dot{T}_a + \Delta\dot{T}_a \sin \frac{2\pi(t - (trise + 360))}{1440}$$

$$q_{sref} = 520 \text{ W/m}^2$$

$$q_s(t) = q_{sref} \sin \frac{\pi(t - trise)}{td} \text{ W/m}^2$$

$$t_{rise} = 9 \text{ π. μ.} = 540 \text{ min}$$

$$t_d = 9,2 \text{ h} = 557 \text{ min}$$

$$\dot{T}_a = 283 \text{ K}$$

$$\Delta\dot{T}_a = 8 \text{ K}$$

Συλλέκτης

$$A = 5 \text{ m}^2$$

$$U = 5.33 \text{ W/m}^2\text{K}$$

$$F_R = 0.9$$

$$b = 1.2$$

$$\dot{m}_C = A/60 \text{ kg/min}$$

Δοχείο αποθήκευσης

$$V_{st} = 0.45 \text{ m}^3$$

$$U_{st} = 0.3 \text{ W/m}^2\text{K}$$

$$T_{st,initial} = 300 \text{ K}$$

Ζήτηση

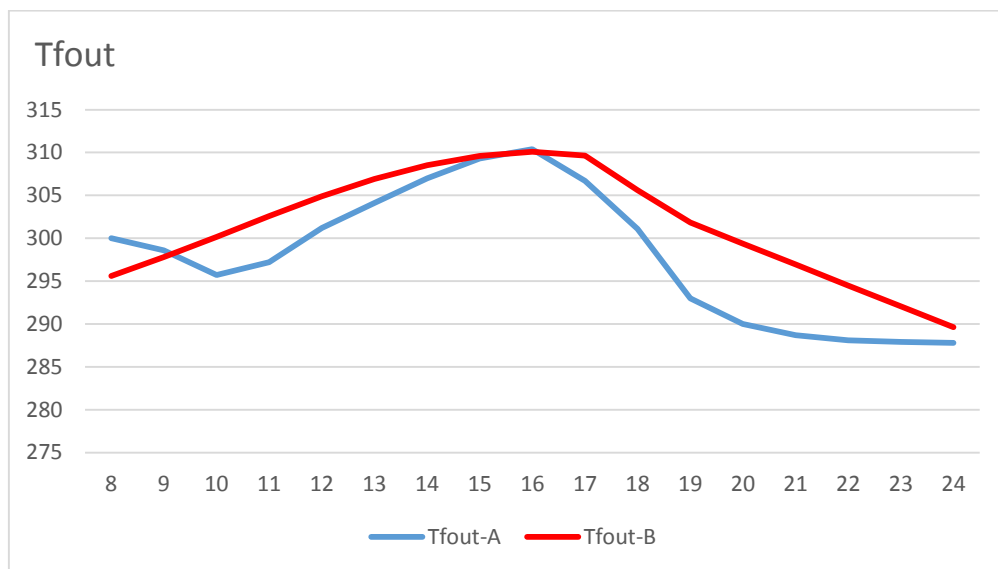
$$\dot{T}_{sup} = 288 \text{ K}$$

$$T_{L,min} \geq 333 \text{ K}$$

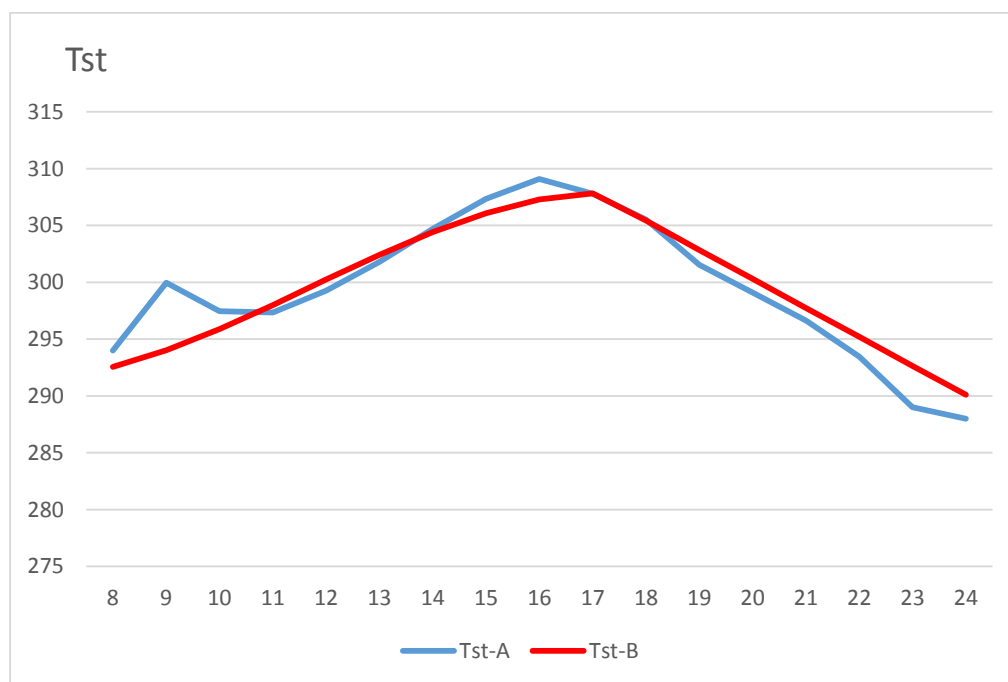
$$T_{st,max} \geq 373 \text{ K}$$

Δυναμικό φορτίο 435 W έως τις 6 μμ. και 1306 W έως τις 12 πμ.

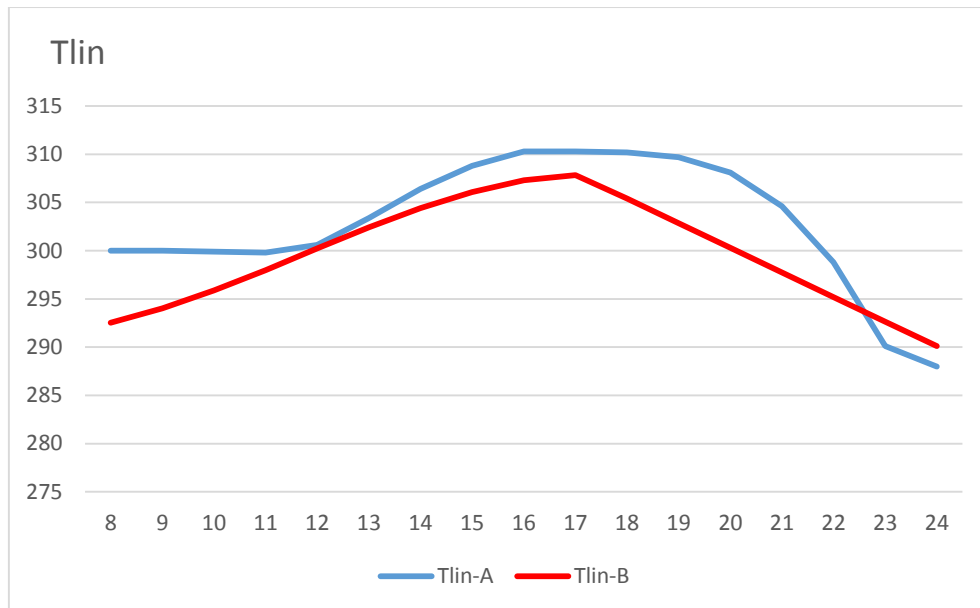
Τα αποτελέσματα των δύο κωδίκων με λεζάντα Α για τον αλγόριθμο Matlab και Β για τον αλγόριθμο Fortran δίδονται στα Σχήματα 4.29 – 4.31.



Σχήμα 4.29 Θερμοκρασία εξόδου του φέροντος ρευστού από το συλλέκτη προς τη δεξαμενή αποθήκευσης για μία ημέρα.



Σχήμα 4.30 Θερμοκρασία αποθήκευσης του φέροντος ρευστού για μία ημέρα.

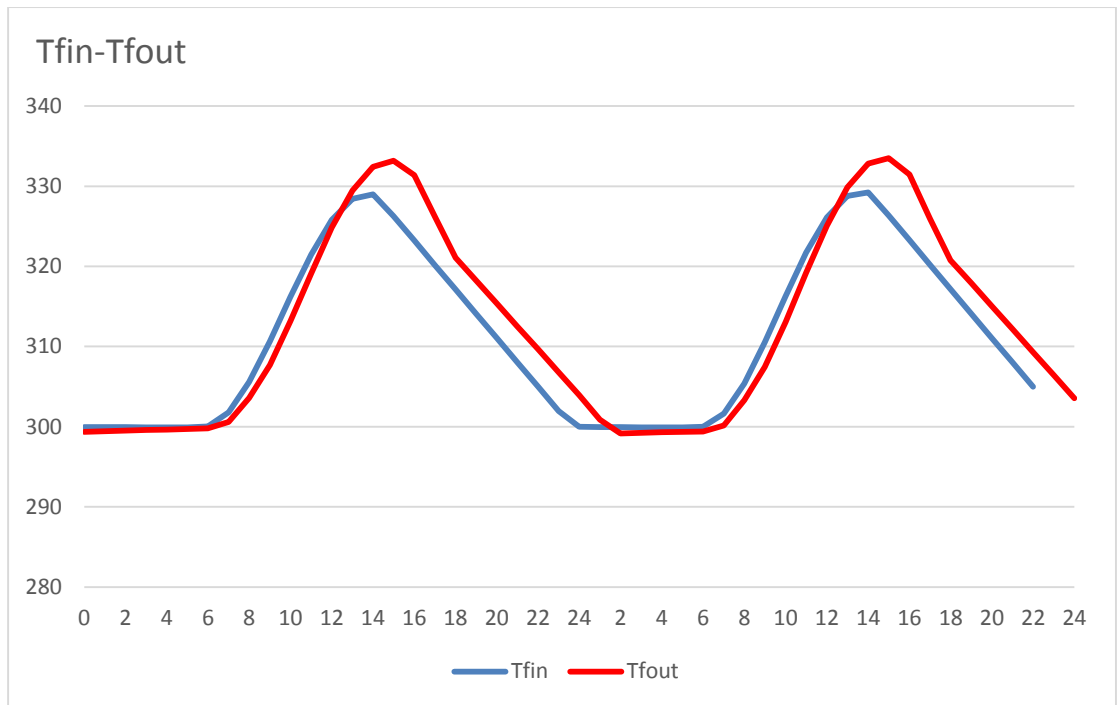


Σχήμα 4.31 Θερμοκρασία εισόδου του φέροντος ρευστού στην κατανάλωση από τη δεξαμενή αποθήκευσης για μία ημέρα.

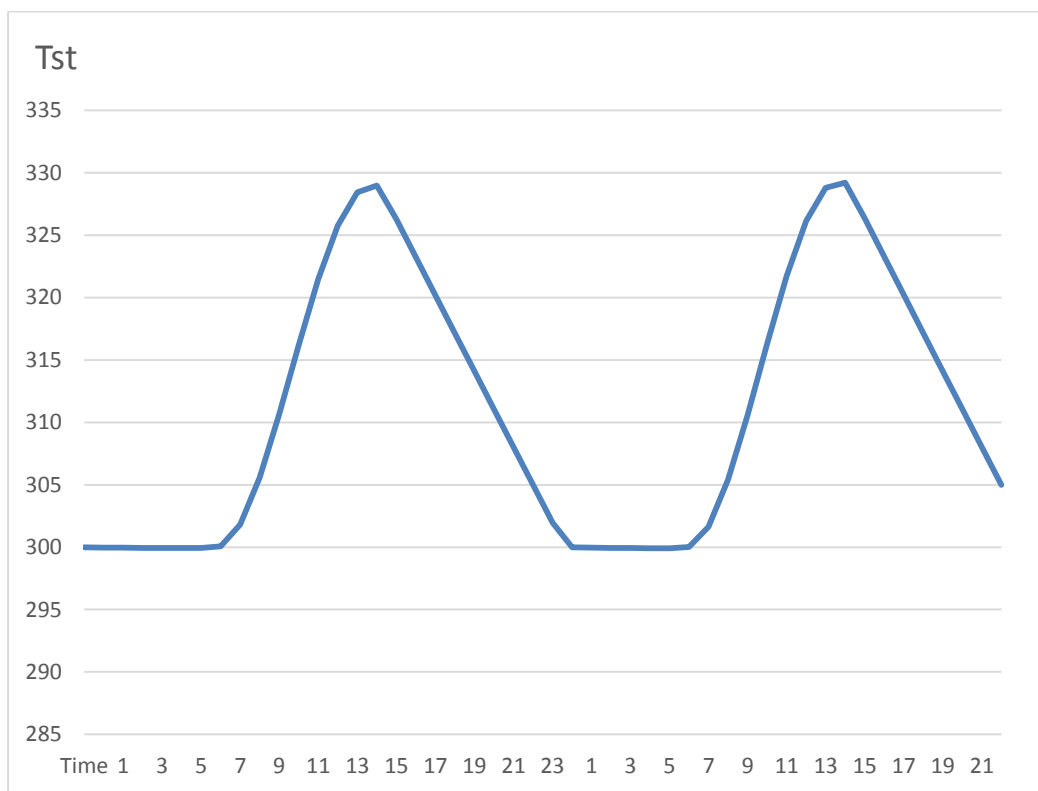
Κατά την εκτέλεση της προσομοίωσης για τη λειτουργία του ZNX, σύμφωνα με όσα αποδείξαμε νωρίτερα, διαπιστώνεται και πάλι η ομαλή λειτουργία του κώδικα Matlab. Οι απότομες καμπύλες που εμφανίζονται στην αρχή της εκτέλεσης οφείλονται αποκλειστικά στην διαφορά του τρόπου αρχικοποίησης των δύο αλγορίθμων και μπορεί να αμεληθεί.

4.4 Παράδειγμα λειτουργίας κώδικα στην ολοκληρωμένη του μορφή

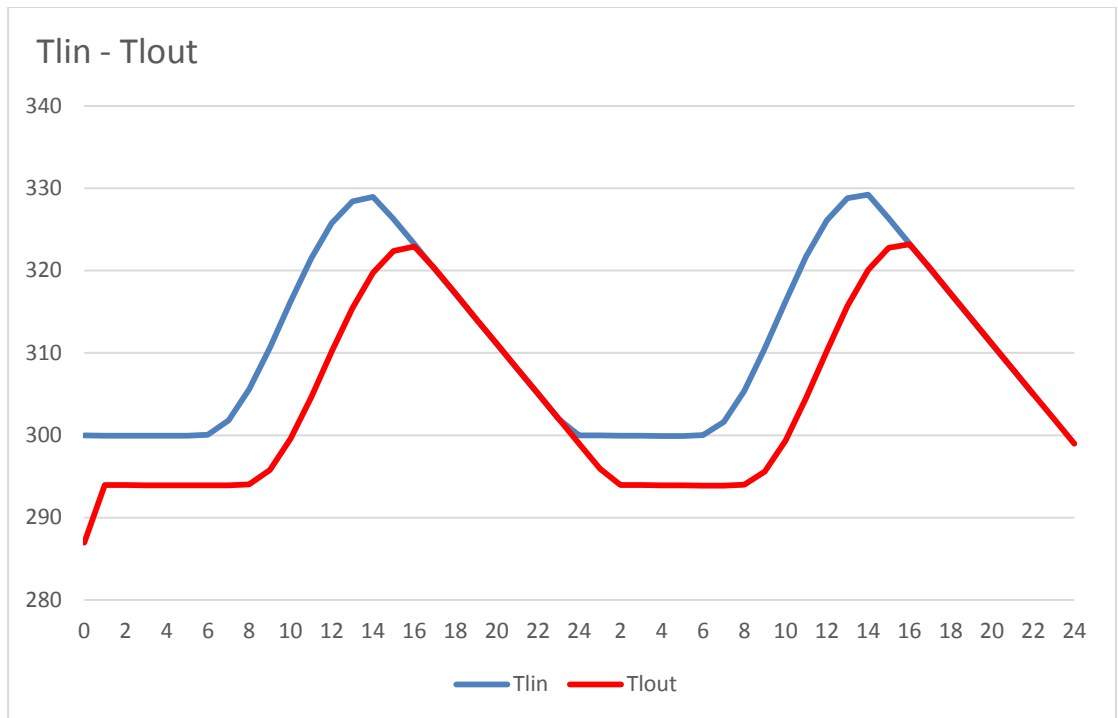
Με την ολοκλήρωση της πιστοποίησης του αλγορίθμου, παρατίθεται ένα συνοπτικό παράδειγμα ΗΘΣ θέρμανσης για δύο ημέρες, οι οποίες τυγχάνουν σε αλλαγή του μήνα. Με αυτό τον τρόπο θα παρουσιαστεί η λειτουργία του αλγορίθμου Matlab στην πλήρη του μορφή, χωρίς τροποποιήσεις στα δεδομένα εισαγωγής. Επιπρόσθετα με την αλλαγή του μήνα θα αποδειχτεί ότι ο αλγόριθμος ενημερώνεται σωστά από τα κλιματικά μας δεδομένα. Τέλος θα ακολουθήσει ένας σύντομος σχολιασμός σχετικά με τα αποτελέσματα που προέκυψαν. Ο αλγόριθμος θα εκτελεστεί από την 31η Οκτωβρίου έως την 1η Νοεμβρίου με επιφάνεια συλλέκτη $A=30 \text{ m}^2$, γωνία συλλέκτη $\beta=30$ μοίρες, χωρητικότητα δεξαμενής 5000 KJ/KW , φορτίο ζήτησης $Ql=4200\text{W}$, χωρίς τη λειτουργία της διαστρωμάτωσης. Οι υπόλοιπες παράμετροι αφήνονται στην προκαθορισμένη τους τιμή από το πρόγραμμα. Τα αποτελέσματα δίδονται στα Σχήματα 4.32 – 4.38.



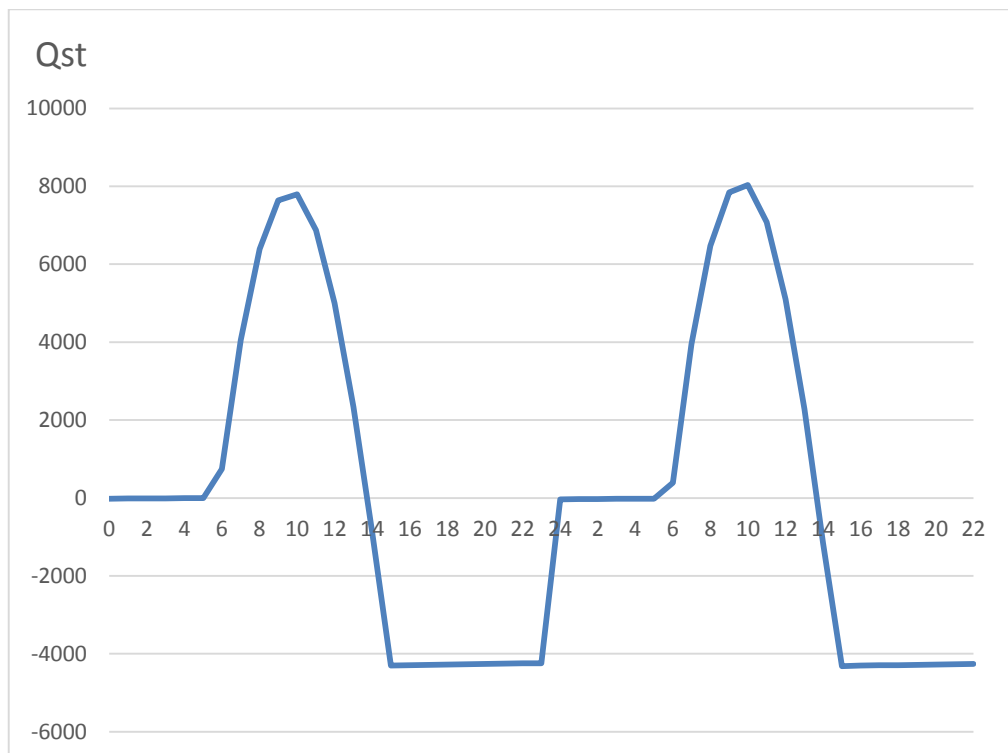
Σχήμα 4.32 Θερμοκρασία εισόδου και εξόδου του φέροντος ρευστού από το συλλέκτη προς τη δεξαμενή αποθήκευσης για την τελευταία ημέρα του Οκτωβρίου και την πρώτη ημέρα του Νοεμβρίου.



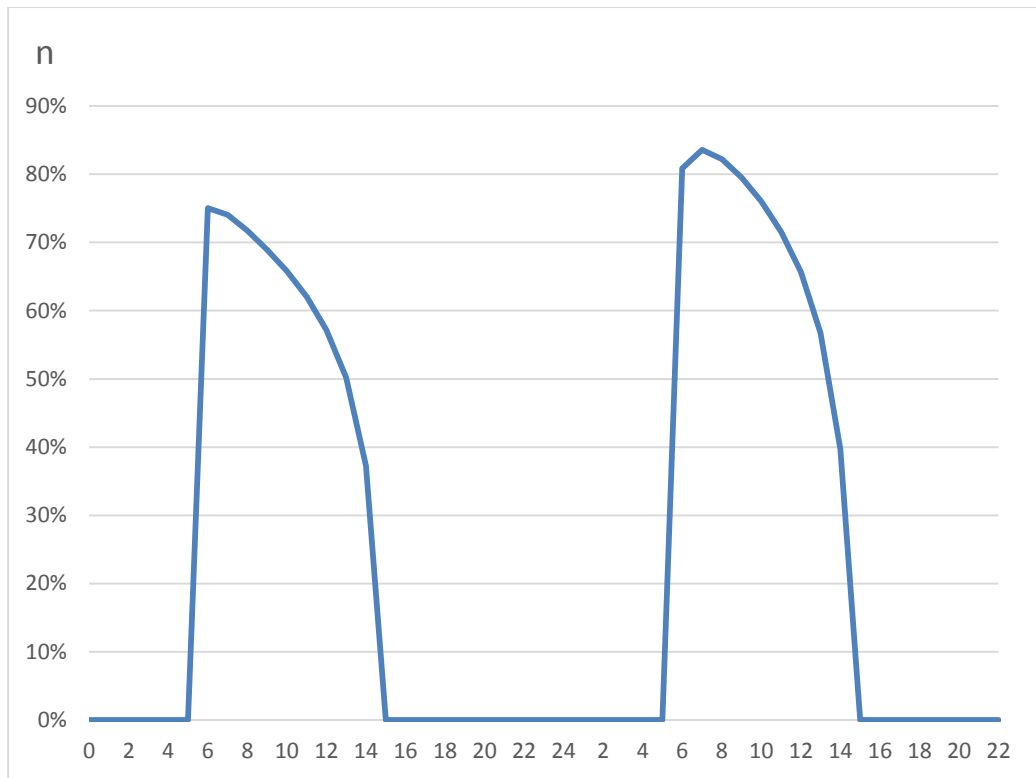
Σχήμα 4.33 Θερμοκρασία αποθήκευσης του φέροντος ρευστού για την τελευταία ημέρα του Οκτωβρίου και την πρώτη ημέρα του Νοεμβρίου.



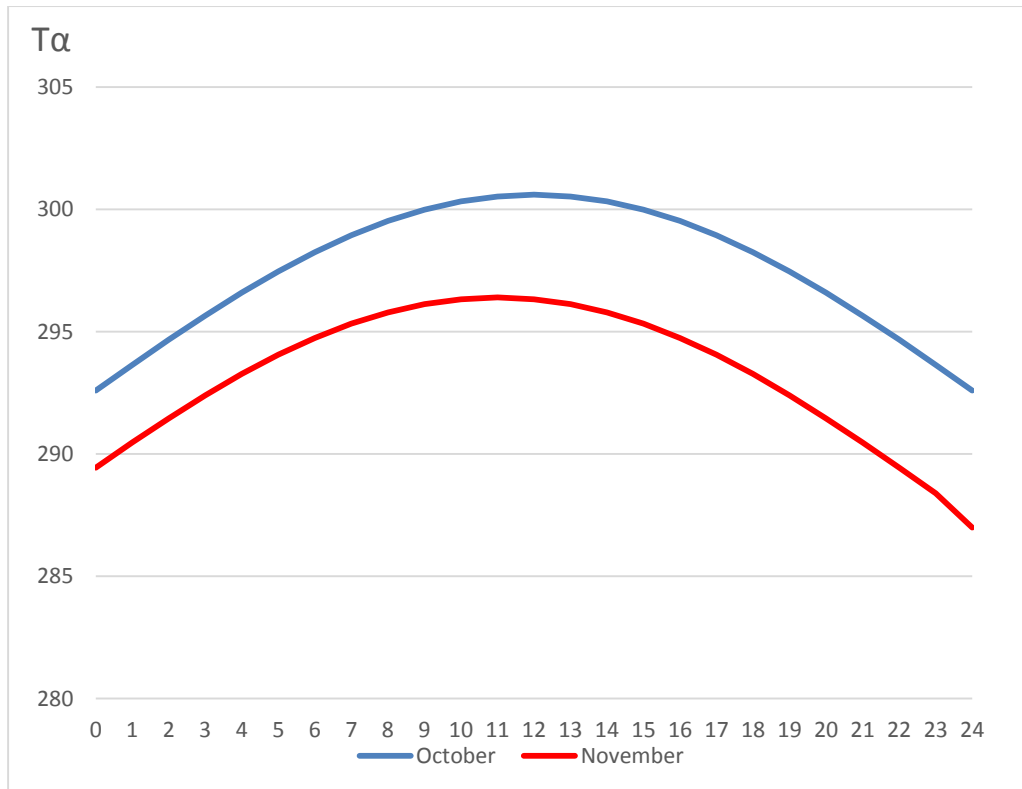
Σχήμα 4.34 Θερμοκρασία εισόδου και εξόδου του φέροντος ρευστού από την κατανάλωση προς τη δεξαμενή αποθήκευσης για την τελευταία ημέρα του Οκτωβρίου και την πρώτη ημέρα του Νοεμβρίου.



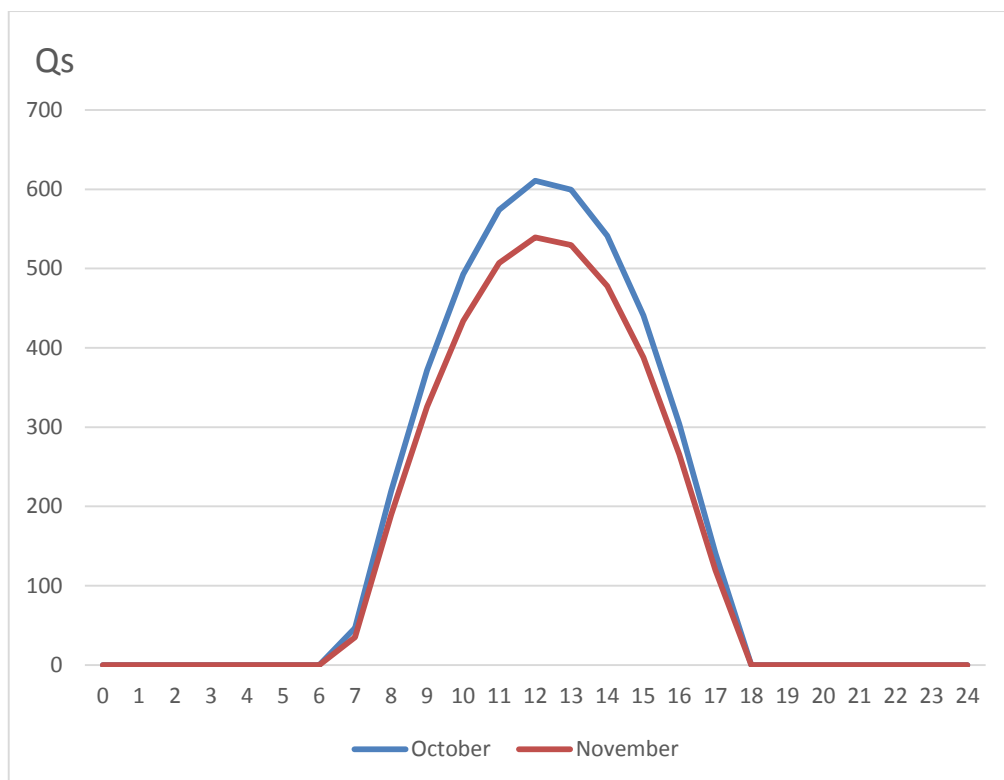
Σχήμα 4.35 Αποθηκευμένη ενέργεια στη δεξαμενή αποθήκευσης για την τελευταία ημέρα του Οκτωβρίου και την πρώτη ημέρα του Νοεμβρίου.



Σχήμα 4.36 Βαθμός απόδοσης του συστήματος για περίοδο δύο ημερών για την τελευταία ημέρα του Οκτωβρίου και την πρώτη ημέρα του Νοεμβρίου.



Σχήμα 4.37 Θερμοκρασία περιβάλλοντος για την τελευταία ημέρα του Οκτωβρίου και την πρώτη ημέρα του Νοεμβρίου.



Σχήμα 4.38 Ακτινοβολία για την τελευταία ημέρα του Οκτωβρίου και την πρώτη ημέρα του Νοεμβρίου.

Month	October	November
total radiation	4342.3	3811.6
hours of sunshine	10	10
radiation/hour of sunshine	434.23	381.16
radiation/sec	0.164710317	0.105877778
qsmax	610.6008091	539.4893328
qsref	502.3041	428.4848
kt	0.54	0.49
ta	19.6	15.4
tsup	21.1	16.9
October/November radiation	1.139232868	

Από τα αποτελέσματα εκτέλεσης του αλγορίθμου Matlab προκύπτει η μετάβαση από τον ένα μήνα στον επόμενο έγινε ομαλά. Η συνολική ακτινοβολία του Οκτωβρίου είναι 4342W, ενώ για τον Νοέμβριο 3811W. Η διάρκεια της ηλιοφάνειας και για τις δύο ημέρες που εξετάστηκαν είναι κατά προσέγγιση 10 ώρες. Η μέγιστη ακτινοβολία που παρουσιάστηκε κατά το ηλιακό μεσημέρι ήταν 502W για τον Οκτώβρη ενώ 482W για το Νοέμβρη. Συνολικά, η ακτινοβολία στο συλλέκτη το μήνα Οκτώβρη ήταν κατά 13% περισσότερη από ότι το Νοέμβρη. Σύμφωνα με τα στοιχεία

του TEE προέκυψε συντελεστής αιθριότητας 0.54 και 0.49, θερμοκρασία περιβάλλοντος 19.6 και 15.4 βαθμούς κελσίου, ενώ η θερμοκρασία του δικτύου ήταν 21.1 και 16.9 βαθμούς κελσίου. Ακόμη έγινε εμφανές πώς κατά τη λειτουργία του συστήματος η θερμοκρασία εξόδου από το συλλέκτη T_{fout} είναι πάντα μεγαλύτερη από τη θερμοκρασία εισόδου T_{fin} , ενώ η θερμοκρασία εισόδου T_{lin} στην κατανάλωση είναι μεγαλύτερη από τη θερμοκρασία εξόδου T_{lout} .

Μία βασική παράμετρος η οποία πρέπει να αναφερθεί λόγω της σημασίας της στην βελτιστοποίηση είναι το ποσοστό κάλυψης f , για το παράδειγμα που εκτελέστηκε η τιμή του ήταν $f=0,85$. Σε μία προσπάθεια βελτιστοποίησης του συστήματος για το ίδιο παράδειγμα ενεργοποιώντας τη λειτουργία της διαστρωμάτωσης και διατηρώντας όλες τις υπόλοιπες παραμέτρους ίδιες, προκύπτει $f=0,889$, δηλαδή μία αύξηση του ποσοστού κάλυψης της τάξης του 4,5%. Εκτελώντας ξανά το παράδειγμα, χωρίς τη λειτουργία της διαστρωμάτωσης μπορούμε να διαπιστώσουμε εύκολα πως η αύξηση αυτή ισοδυναμεί με αύξηση της επιφάνειας του συλλέκτη σε $A=32 \text{ m}^2$ με $f=0.898$. Όλα παραπάνω ενισχύουν την ανάγκη ύπαρξης της διαστρωμάτωσης για ένα σύστημα αλλά και αποδεικνύουν την ευκολία που προσδίδει στην βελτιστοποίηση το ποσοστό κάλυψης f .

Τέλος, με την εκτέλεση του αλγορίθμου έγινε προφανές πώς η απόκλιση στην ακτινοβολία και τις θερμοκρασίες δεν συμβαίνουν από την μία ημέρα στην άλλη με την αλλαγή του μήνα αλλά σταδιακά με την πάροδο του, πράγμα το οποίο αποτελεί και μία αδυναμία-σφάλμα στις εκτιμήσεις του αλγορίθμου προσομοίωσης.

Κεφάλαιο 5 Συμπεράσματα

Στην παρούσα διπλωματική εργασία μελετήθηκαν τα ηλιακά θερμικά συστήματα για θέρμανση και ζεστό νερό χρήσης (ZNX). Αρχικά, παρουσιάστηκε η μεθοδολογία και οι μαθηματικές σχέσεις που περιγράφουν τα στοιχεία των ηλιοθερμικών συστημάτων καθώς και τον τρόπο λειτουργίας τους. Πραγματοποιήθηκε περιγραφή του ηλιακού συλλέκτη, της δεξαμενής αποθήκευσης και του συστήματος ελέγχου για την κάλυψη των αναγκών σε φορτία θέρμανσης και ZNX και αναπτύχθηκε πηγαίος κώδικας σε Matlab με στόχο την προσομοίωση των ηλιακών συστημάτων μέσω της χρήσης ενός παράθυρου διεπαφής χρήσης. Τέλος έγινε σύγκριση του κώδικα Matlab μέσω αξιόπιστων πηγών, όπως ο κώδικας Fortran του Εργαστηρίου Φυσικών Χημικών Διεργασιών του τμήματος Μηχανολόγων Μηχανικών του Πανεπιστημίου Θεσσαλίας, το στούντιο προσομοίωσης Trnsys και αποτελέσματα από το διαδικτυακό χώρο του Κοινού Κέντρου Ερευνών της Ευρωπαϊκής Επιτροπής.

Τα αποτελέσματα του αλγορίθμου για τις διάφορες τοποθεσίες της Ελλάδας και τις λειτουργίες της διαστρωμάτωσης, των βαθμομερών, του ζεστού νερού χρήσης, του εναλλάκτη και του δυναμικού φορτίου ήταν σύμφωνα με την βιβλιογραφία. Οι μικρές αποκλίσεις οφείλονται κυρίως στον τρόπο υπολογισμού της ακτινοβολίας και την έναρξη - λήξη της ηλιοφάνειας. Είναι προφανές τόσο από την μεταπτυχιακή εργασία όσο και από την αρχή λειτουργίας των ΗΘΣ ότι η ακτινοβολία αποτελεί κύριο παράγοντα για τα αποτελέσματα της προσομοίωσης.

Ο υπολογισμός της ακτινοβολίας αποδείχθηκε το δυσκολότερο τμήμα της εργασίας και για το λόγο αυτό επιστρατεύτηκαν δύο μέθοδοι σε συνδυασμό με πίνακες και στοιχεία από το TEE. Κατά την πιστοποίηση του αλγορίθμου οι μέθοδοι παρουσίασαν ελάχιστες αποκλίσεις μεταξύ τους. Διαπιστώθηκε ότι πλησιάζουν τις εκτιμήσεις του Trnsys, όμως υπήρχαν αρκετές διαφορές κατά τη σύγκρισή τους με τα στοιχεία της Ευρωπαϊκής Επιτροπής Ερευνών. Η μέθοδος υπολογισμού του Trnsys αν και ήταν αρκετά κοντά στην παρούσα προσέγγισή του κώδικα Matlab δεν ακολουθούσε ημιτονοειδή κατανομή, αλλά προσομοίωνε την ακτινοβολία σε πραγματικές καιρικές συνθήκες. Η μέθοδος που εφαρμόζεται στο Κοινό Κέντρο Ερευνών της Ευρωπαϊκής Επιτροπής ακολουθεί ημιτονοειδή κατανομή, όπως και ο αλγόριθμός μας, αλλά οι τιμές είναι αρκετά μεγαλύτερες. Η μέθοδος υπολογισμού της βιβλιογραφίας εμφάνισε λίγο υψηλότερες τιμές από τον αλγόριθμό Matlab, αλλά όχι

τόσο υψηλές όσο αυτές του Κοινού Κέντρου Ερευνών και αστοχούσε στις πραγματικές ώρες έναρξης και λήξης της ημέρας. Η ώρα του ξημερώματος και ηλιοβασιλέματος για τις διάφορες ημέρες του αλγορίθμου Matlab ήταν ορθές και σύμφωνες με όλες τις άλλες εκτιμήσεις όπως αποδείξαμε.

Επιπλέον καθοριστικές παράμετροι για τα αποτελέσματα ήταν η κλίση του ηλιακού συλλέκτη που φάνηκε να επηρεάζει άμεσα την απόδοση του συστήματος και την πρόσληψη ακτινοβολίας κατά τη διάρκεια της ημέρας. Η διαστρωμάτωση ήταν ένας ακόμη παράγοντας που αύξησε σημαντικά την απόδοση του συστήματος. Εξίσου σημαντική ήταν η γωνία τοποθέτησης και ο προσανατολισμός που μετέβαλλαν το κλάσμα κάλυψης. Η αύξηση της επιφάνειας του συλλέκτη φάνηκε να οδηγεί επίσης σε αύξηση του ποσοστού κάλυψης. Τέλος, αρκετά σημαντικός είναι και ο συντελεστής αντανάκλασης ο οποίος έχει άμεση σχέση με την ακτινοβολία στο συλλέκτη.

Με την εφαρμογή των πηγαίων κωδικών Matlab και Fortran δημιουργήθηκαν διαγράμματα με τις κατανομές των θερμοκρασιών στη δεξαμενή αποθήκευσης και στην είσοδο και έξοδο του ηλιακού συλλέκτη. Έγινε εμφανές ότι οι θερμοκρασίες παρουσιάζουν διαφορές ως προς τις τιμές τους ανάλογα με τη περίοδο που μελετήθηκε με τις υψηλότερες να παρατηρούνται κατά τους θερινούς μήνες όπου υπάρχει υψηλή ηλιακή ακτινοβολία. Η κατανομή των τιμών ήταν πάντα ανάλογη με την κατανομή της ακτινοβολίας. Οι κατανομές των θερμοκρασιών παρουσιάζουν ημιτονοειδή μορφή κάτι το οποίο οφείλεται στον τρόπο υπολογισμού της προσπίπτουσας ηλιακής ακτινοβολίας κατά την εφαρμογή του πηγαίου κώδικα και με τους δύο τρόπους ακτινοβολίας που προτάθηκαν. Προφανώς η ημιτονοειδής κατανομή δεν ανταποκρίνεται απόλυτα στις πραγματικές συνθήκες, αφού η ακτινοβολία αυξομειώνεται στα διάφορα στάδια της ημέρας ανάλογα με τις τοπικές καιρικές συνθήκες που επικρατούν.

Η πιστοποίηση του υπολογισμού θερμοκρασίας νερού παροχής δικτύου εμφανίστηκε να λειτουργεί ομαλά συγκρίνοντας τις δύο περιοχές της Αθήνας και οι μικρές διαφορές που παρατηρήθηκαν οφείλονται στην απόσταση μεταξύ των δύο περιοχών και στο γεγονός ότι η προσομοίωση πολύ συχνά παρουσιάζει αποκλίσεις σε σχέση με τις πραγματικές μετρήσεις. Αξιοσημείωτη παρατήρηση αποτέλεσε η διακύμανση της θερμοκρασίας με την πάροδο του χρόνου και η εξάρτηση της από το βάθος και τη σύσταση του εδάφους. Ένας ακόμη παράγοντας που επηρεάζει ήταν η θερμοκρασία του περιβάλλοντος.

Το λογισμικό Matlab που αναπτύχθηκε είναι εύκολα προσαρμοζόμενο στα χαρακτηριστικά ποικίλων εφαρμογών θέρμανσης και ζεστού νερού χρήσης και ιδιαίτερα φιλικό προς τον χρήστη. Δύναται να χρησιμοποιηθεί ευρέως για εκπαιδευτικούς σκοπούς, όπως επίσης και για την συστηματική μελέτη, σχεδιασμό και βελτιστοποίηση.

Υπάρχουν μερικές βελτιώσεις στο πηγαίο κώδικα που θα μπορούσαν να αποτελέσουν αντικείμενο περαιτέρω εργασίας. Ένα βασικό στοιχείο αποτελεί το χρονικό βήμα προσομοίωσης. Οι υπολογισμοί πραγματοποιούνται με χρονικό βήμα 1 δευτερόλεπτο το οποίο είναι σταθερό για κάθε εφαρμογή και θα μπορούσε να τροποποιηθεί ώστε να επιλέγεται από τον χρήστη ανάλογα με τον χρόνο προσομοίωσης που επιθυμεί να μελετήσει, διότι η μεγάλη υπολογιστική ισχύς που απαιτείται τώρα βαραίνει το σύστημα και καθυστερεί την προσομοίωση, ειδικά όταν αυτή εκτελείται για πολλές ημέρες. Λόγω του όγκου των παραμέτρων που διέπουν τα ΗΘΣ, κατά τη συγγραφή και μεταγλώττιση του κώδικα χρησιμοποιήθηκαν πολλές μεταβλητές και ο κώδικας έγινε πολύπλοκος και δύσκολος στη διαχείριση. Αυτό θα μπορούσε να αποφευχθεί χωρίζοντας το πρόγραμμα σε υποπρογράμματα. Η μετάβαση από μήνα σε μήνα συμβαίνει με την αλλαγή της ημερομηνίας. Αυτό σημαίνει ότι χαρακτηριστικές τιμές όπως η θερμοκρασία του περιβάλλοντος μεταβάλλονται απότομα. Σε μελλοντική εργασία θα μπορούσε να προβλεφθεί μια ομαλή κατανομή, όπως πραγματικά συμβαίνει. Επίσης, με σχετικά λίγες προσθήκες θα μπορούσε να γίνει προσομοίωση συστημάτων με αέρα ή άλλα φέροντα ρευστά, και να προσομοιωθούν συλλέκτες διαφορετικής γεωμετρίας. Για τον υπολογισμό της θερμοκρασίας νερού παροχής δικτύου θα μπορούσε να γίνει προσθήκη στοιχείων για περιοχές της Ελλάδας ώστε να επιτευχθεί καλύτερη ακρίβεια κατά τον υπολογισμό της. Τέλος, για τον υπολογισμό της προσπίπτουσας ακτινοβολίας στον συλλέκτη είναι δυνατό να προστεθούν περισσότερες μέθοδοι προσομοίωσης, ενώ ο πηγαίος κώδικας μπορεί να τροποποιηθεί ώστε να λαμβάνει τιμές από βάσεις δεδομένων ώστε να υπάρχει μεγαλύτερη ακρίβεια των αποτελεσμάτων.

Αναφορές

- [1] Pablo-Romero M. D., Sánchez-Braza A. and Lerma E., “Solar Thermal Energy Use in EU-27 Countries: Evolution and promotion”, *Alternative Energies*, 34, 241-266, 2013.
- [2] <http://ec.europa.eu/eurostat/web/energy/data/main-tables>
- [3] Ευρωπαϊκή Επιτροπή, “Πλαίσιο για τις πολιτικές που αφορούν το κλίμα και την ενέργεια με χρονικό ορίζοντα το έτος 2030”, *Πράσινη Βίβλος*, Βρυξέλες, 2013.
- [4] Kalogirou S. A., *Solar Energy Engineering: Processes and Systems*, Elsevier, 2014.
- [5] www.estif.org.
- [6] Ivancic A., Mugnier D., Stryi-Hipp G. and Weiss W., “Solar Heating and Cooling Technology Roadmap”, Brussels, 2014.
- [7] Rehman H., Hirvonen J. and Sirén K., “Design of a simple control strategy for a community-size solar”, *Energy Procedia*, 91 486-495, 2016.
- [8] Yu N., Wang R. and Wang L., “Sorpton thermal storage for solar energy”, *Progress in Energy and Combustion Science*, 39 (5), 514-589, 2013.
- [9] Howell J. R., Bannerot R.B. and Vliet G.C., *Solar Thermal Energy Systems-Analysis and Design*, McGraw Hill Book Co., 1982.
- [10] Καπλάνης Σ., *Ηπιες Μορφές Ενέργειας ΙΙ-Ηλιακή Μηχανική*, Ίων, 2004.
- [11] <http://www.trnsys.com/>
- [12] <http://www.4m.gr/>
- [13] <http://www.velasolaris.com/>
- [14] Duffie J. and Beckman W, *Solar Engineering of Thermal Processes*, John Wiley & Sons Inc., 1991.
- [15] Okafor I.F. and Akubue G., ”Solar thermal water heating system designing using F-chart Method”, *International Journal of Scientific & Engineering Research*, Volume 3, Issue 9, September, 2012.

- [16] Fanney A. H. and Klein S.A., “Performance of solar domestic hot water systems at the National Bureau of Standards—measurements and predictions”, *Journal of Solar Energy Engineering* 105.3, 311-321, 1983.
- [17] Lof G., *Active solar systems Solar Heat Technologies*, The MIT Press, June, 1993.
- [18] <http://web.tee.gr/>
- [19] <https://www.mathworks.com/products/matlab.html>
- [20] <http://re.jrc.ec.europa.eu/pvgis/apps4/pvest.php>
- [21] http://re.jrc.ec.europa.eu/pvgis/apps4/databasehelp_en.html

Παραρτήματα

Α. Πηγαίος κώδικας ΗΘΣ για θέρμανση χώρων και ZNX

```
function SolsysF(handles)

%switches%

%for first way of calculating radiation 1, for second 0
irv=get(handles.Gui_data,'Value');
irv=get(handles.Real_env,'Value');
%for domestic hot water 1, otherwise 0%
idhw=get(handles.DHW,'Value');
%for load variation 1, otherwise 0%
sload=get(handles.Time_load,'Value');
%for heat exchanger 1, otherwise 0%
iheat=get(handles.Heat_exch,'Value');
%for stratified storage 1, mixed 0, otherwise -1% change name at gui
global itypest;
itypest=get(handles.Stratified,'Value');
%switch for degree days
idd=get(handles.Degrees_days,'Value'); %default is 0 where degree
days arent considered, for 1 they are calculated

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Simulation Parameters%

%variables to calculate range from gui

%day to begin the simulation
dday=get(handles.Day_input,'Value');

%month for the simulation
month=get(handles.Month_input,'Value');

%number of days to run the simulation
range=get(handles.Range_input,'Value');

%city for the simulation
city=get(handles.City,'Value');

%residence heat transfer coefficient
uresidence=get(handles.Uresidence,'Value');

%area of residence
aresidence=get(handles.Aresidence,'Value');

%matrix of loads in case choice dynamic load is selected
a_matrix=get(handles.Time_dependent_load,'Data');
ml u=a matrix/3600;
```

```

%vectors include day per year, month index and sun duration per
month
iupdate=[0 31 59 90 120 151 181 212 243 273 304 334 365];
imonth= ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k',
'l'];
sun_duration=[4.5 5.1 6 8 9.8 11.4 12.1 11.5 9.4 7 5.5 4.4 4.5];
hr = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
0];

for jj=1:62
    if (city==jj)
        icity=num2str(jj);
    end
end

%indexes for traversing the excel file
index=strcat(imonth(month),icity);
day=iupdate(month)+dday-1;
nday=day;

%soil heat transfer coefficient for water calculations
i_diff=get(handles.Soiltype,'Value');
%i_diff=4;
if i_diff==1 % ΓΡΑΝΙΤΗΣ
    diff=0.49*0.000001;
elseif i_diff==2 %ΑΣΒΕΣΤΟΛΙΘΟΣ
    diff=0.8*0.000001;
elseif i_diff==3 %ΜΑΡΜΑΡΟ
    diff=1.39*0.000001;
elseif i_diff==4 %ΨΑΜΜΑΜΙΤΗΣ
    diff=1.06*0.000001;
elseif i_diff==5 %ΑΡΤΙΛΙΟΣ
    diff=0.139*0.000001;
end

%Depth of water network
d_depth=get(handles.Input_dta,'Value');
%Coldest day for the soil
soil_day=get(handles.Input_tam,'Value');

%update variables from real values
%referance sunlight radiation

tam = xlsread('TEE.xls','s2',index)+273
dta = xlsread('TEE.xls','s8',index)

month=month+1;
DD =
xlsread('TEE.xls','s3',index)*uresidence*aresidence/(iupdate(month)
-iupdate(month-1)); %degree days
qsref = xlsread('TEE.xls','s4',index)*1000/(iupdate(month)-
iupdate(month-1))/sun_duration(month-1) %fixme
month = month -1;
kt = xlsread('TEE.xls','s5',index) %kt coefficient
tsup = xlsread('TEE.xls','s6',index)+273 %supply temperature from
water network
if (tsup==273)

```

```

    tsup = (xlsread('TEE.xls','s2',index)-
xlsread('TEE.xls','s7',index)*exp(-
d_depth*((pi/(365*diff))^0.5))*cos((2*pi/365)*(nday-soil_day-
(d_depth/2)*((365/(pi*diff))^0.5))))+273 %estimation of temperature
from water network
end

PSist=30;

%local meridian based on selected location
indexmeri=strcat('b',icity);
PSilocal=xlsread('TEE.xls','s1',indexmeri);

%local lattitude based on selected location
indexlatti=strcat('a',icity);
phi=xlsread('TEE.xls','s1',indexlatti);

%converts minutes to seconds
conversion=1;

%timestep for the simulation in minutes, current timestep is 1
second
timestep=1/86400;

%vector of results
vec=zeros(30);
ve=zeros(30);
%index for results vector
cell=0;

%counter for generic purposes
loopNr=-1;
count_timestep=0;
omega_correction=180;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Collector data input%

%insert collector surface a [m^2]%
a=get(handles.depthSlider,'Value');

%insert collector value b%
u=get(handles.Input_b,'Value');

%insert collector property ta_eff%
ta_eff=get(handles.Input_taeff,'Value');

%insert collector heat removal parameter fr%
fr=get(handles.Input_fr,'Value');

%insert specific heat of working fluid [J/kg*k]%
c_v=1;
if c_v==1 % Water collector
    c=4200;
elseif c_v==2 %air collector
    c=1;

```



```

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Load data input%

%insert load ql%
ql=get(handles.Input_ql, 'Value');

    if (idd==1)
        ql=DD;
    end
    if (idhw==1 && sload==0)
        qlload=get(handles.Input_ql, 'Value'); %must be fixed
    end

%insert load mass flow ml [kg/s]%
ml=get(handles.Input_ml, 'Value');

%insert load minimum temperature tl,min [k]%
tlmin=get(handles.Input_tlmin, 'Value');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Storage data input%

%insert storage surface ast [m^2]%
ast=get(handles.Input_ast, 'Value');

%insert storage heat transfer coefficient ust [W/m^2*k]%
ust=get(handles.Input_ust, 'Value');

%insert storage capacity mst*cp {J/k}%
mst=get(handles.mstslider, 'Value');
mst=mst*1000;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%*****Inputs and variables*****%

%temperature of collector (average)%
ta=273+8 ;

%effective thermal energy%
qu=4200;

```

```

%Outlet temperature of fluid%
tfout=305;

%inlet and Outlet temperature difference%
dtl=8;

%Pump on/off%
dc=0;
dl=0;

%total quantities%
sum_l=0;
sum_s=0;
sum_us=0;

%insert collector mass flow rate mc [kg/s] from rule of thumb%
mc=a/60;

%insert initial storage temperature tst,initial [k]%
tst=300;

%insert maximum storage temperature tstmax allowed [k]%
tstmax=373;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%economic data input%

%P initial investment cost P & ieff%

%insert cost of collector per m^2
colcost=300*a;

%insert cost of installation
instcost=2000;

%insert years for investment
ni=10;

%Winnings gw
gw=600;

%interest i & cost of fuel j
i=0.03;
j=0.05;

%Annual interest
ai=4.93;

%Annual rise for fuel prices
jp=-4.96; %Petrol
je=4.92; %Electricity

%Cost of solar collector per m^2 [Euro/m^2]
csc=300;

%Cost of installation and rest of the materials [Euro]

```

```

cin=2000;

%Years of depreciation
years=10;

%Cost of electricity[Euro/kwh]
ce=0.09460;

%Cost of petrol[Euro/kwh]
ce=0.09460;
cp=0.0744;

%Ieffective for economic results
iefe=(ai*0.01-je*0.01)/(1+je*0.01);
iefp=(ai*0.01-jp*0.01)/(1+jp*0.01);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%heat exchanger coefficient%
eps=0.6;

%Water density%
rho=1000*(1-(tstmax+288.9414)/(508929.2*(tstmax+68.12963))*(tstmax-
3.9863)^2);

%Storage volume
vstorage=0.32;

%Storage inner diameter
rindia=3;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Angle and qs calculation parameters%

%Angle b
beta=get(handles.angle_b, 'Value');
%Angle gamma
gmr=get(handles.angle_g, 'Value');
%----- POP UP MENU -----%

%reflection coefficient%
i_rr=get(handles.Reflectivity, 'Value');

%i_rr=get(handles.collector, 'Value');
if i_rr==1 % ΦΡΕΣΚΟ ΧΙΟΝΙ
rr=0.95;
elseif i_rr==2 % ΒΡΩΜΙΚΟ ΧΙΟΝΙ
rr=0.7;
elseif i_rr==3 % ΠΑΓΟΣ
rr=0.4;
elseif i_rr==4 % ΗΕΡΕΜΗ ΘΑΛΑΣΣΑ
rr=0.3;
elseif i_rr==5 % ΑΜΜΟΣ
rr=0.45;
elseif i_rr==6 % ΓΡΑΣΙΑΙ
rr=0.25;
elseif i_rr==7 % ΔΑΣΟΣ

```

```

rr=0.2;
elseif i_rr==8 % ΤΣΙΜΕΝΤΟ
rr=0.35;
elseif i_rr==9 % ΑΣΦΑΛΤΟΣ
rr=0.2;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%plot variables

global TST;
TST=zeros(1,250*range);
TST(:)=300;
global TST_str;
TST_str=zeros(1,250*range);
TST_str(:)=300;
global QU;
QU=zeros(1,250*range);
global N_A;
N_A=zeros(1,250*range);
global QS;
QS=zeros(1,250*range);
xmin=0*range;
xmax=250*range;
global x;
x=linspace(xmin,xmax,250*range);

qst=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Stratified water tank%

%Number of layers of storage tank%
layers=5;

%initialize variables%
if (itypest==0)
    for i = 1:(layers)
        tstrat(i)=300;
        gm(i)=0;
        fm(i)=0;
        cld(i)=0;
        ldd(i)=0;
        tsttmp(i)=283;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Main Algorithm %%%%%%%%%

%hot water main supply
tfin=289;

%minimum temperature
tlin=293;

%outlet temperature of fluid
tfout=300;

```

```

%change of time +-1 hour at 26/3 and 29/10
if (nday>85 && nday<302)
    correction=-1;
else
    correction=0;
end

qsmax=qmax(nday,kt,correction,omega_correction,phi,gmr,beta,rr,irv)

    if (ast<0) %Storage area calculation if Ast is not given and
the volume storage is given
        vstorage=mst/(rho*c);
        lstorage=(4*vstorage*(rindia^2)/pi)^0.33;
        dst=lstorage/rindia;
        ast=pi*(dst/2+lstorage)
    end

b=u*tam/qsmax;

    if (idhw==1) %domestic hot water%
        tlout=tsup;
    else
        tlout=300;
    end

gq=mc*c*tam/qsmax/a;%gamma coefficient for heat exchanger

wait=waitbar(0, 'Calculating ...', 'Name', 'Time marching');

while (nday<(day+range))

%This section updates variables based on real values in case the
month has changed
    for ii=12:-1:1
        if mod(nday,iupdate(ii))<=timestep
            if (nday > iupdate(ii));
                index=strcat(imonth(ii),icity);
                tam = xlsread('TEE.xls','s2',index)+273
                dta = xlsread('TEE.xls','s8',index)
                month=month+1;
                DD =
xlsread('TEE.xls','s3',index)*uresidence*aresidence/(iupdate(month)
-iupdate(month-1));
                if (idd==1)
                    ql=DD;
                end
                qsref =
xlsread('TEE.xls','s4',index)*1000/(iupdate(month)-iupdate(month-
1))/sun_duration(month) %fixme
                month=month-1;
                kt = xlsread('TEE.xls','s5',index)
                tsup = xlsread('TEE.xls','s6',index)+273
                if (tsup==273)
                    tsup = (xlsread('TEE.xls','s2',index)-
xlsread('TEE.xls','s7',index)*exp(-
d_depth*((pi/(365*diff))^0.5))*cos((2*pi/365)*(nday-soil_day-
(d_depth/2)*((365/(pi*diff))^0.5))))+273;

```

```

end
qsmax=qmax(nday,kt,correction,omega_correction,phi,gmr,beta,rr,irv)
    %h=qsmax;
end
end
end
%end of section

%This section calculates solar and local time

%trajectory coefficient%
bq=(360*(floor(nday-1)))/365;

%time correction factor based on trajectory coefficient
etraj=0.0172+0.3278*cosd(bq)*7.3456*sind(bq)-3.3468*(cosd(bq))^2-
9.3544*(sind(bq)^2);

%change of time +-1 hour at 26/3 and 29/10
if (nday>85 && nday<302)
    correction=-1;
else
    correction=0;
end

%calculation of local time%
tlocal= mod(nday,floor(nday))*24;

%calculation of solar time%
tsolar=tlocal+(4*(PSist-PSilocal)+etraj)/60;

%calculates solar radiation based on collector angles and location

%delta calculations%
delta=23.45*sind(360*((284+floor(nday))/365));

%wss calculations%
if (delta*phi<0)
wss=acosd(-tand(phi)*tand(delta));
else
wss=acosd(-tand(phi-beta)*tand(delta));
%wss=acosd(-tand(phi)*tand(delta));
end

%sunshine duration%
sunshine=wss/15;

%omega calculation
if (mod(count_timestep,1)==0)
    omega_correction=180;
end

omega=(15*(tsolar+correction))+omega_correction-360;

%as calculation%
as=(asind(sind(delta)*sind(phi)+cosd(delta)*cosd(phi)*cosd(omega)));
;

%gamma s angle%

```

```

gms=asind(cosd(delta)*sind(omega)/cosd(as));

%theta angle%
theta=acosd(sind(as)*cosd(beta)+cosd(as)*sind(beta)*cosd(gms-gmr));

%solar stable
isc=1367;

%extraterrestrial solar radiation
io=isc*(1+0.034*cosd((360*(nday))/365));
ho=io*sind(as);
if ho>0
if (irv==0)

    h=kt*ho;
    if (kt<= 0.85 && kt>=0.30)
        ib=-520+1800*kt;
    elseif (kt<0.3)
        ib=0;
    else
        ib=0;
    end

    hb=ib*sind(as);
    hd=h-hb;

else

    h=kt*ho;
    if (kt<0.35)
        hd=(1-0.249*kt)*h;
    elseif (kt>=0.35 && kt<=0.75)
        hd=(1.557-1.84*kt)*h;
    elseif (kt<0.85)
        hd=0.177*h;
    else
        hd=0;
        h=0;
    end
    ib=(h-hd)/sind(as);
    hb=ib*sind(as);
end

%calculates the actual solar radiation%
qs=(cosd(theta)/sind(as))*hb+hd*((cosd(beta/2))^2)+h*rr*((sind(beta/2))^2);
else
    qs=0;
end

%decide if it is day or night
if qs>0 %&& tsolar > 12-sunshine/2 && tsolar < 12+sunshine/2
    day_night=1;
else
    day_night=0;
end

qs=qs*day_night;

```

```

%if we have heat exchanger, then fr'/fr is calculated%
if (iheat==1)
    if (mc*c>ml*c && ml>0) %find minimum mass flow rate%
        fr_h=fr/(1+((ast*fr*u)/(mc*c))*((mc*c)/(eps*ml*c))-
1));
    else
        fr_h=fr/(1+((ast*fr*u)/(mc*c))*((mc*c)/(eps*mc*c))-
1));
    end
end

%calculates the load in case it is time dependent
if (sload==1 && idhw==1)
    for i=1:24
        if (hr(i)==(floor(tlocal)))
            mload=ml_u(i);
            qload=mload*(tlmin-tsup)*c;
        end
    end
elseif (sload==1 && idhw~=1)
    for i=1:24
        if (hr(i)==(floor(tlocal)))
            ml=ml_u(i);
            ql=ml*(tlmin-tlout)*c;
        end
    end
end

%different initialization in case we have dhw
if (idhw==1)
    if (itypest==0)
        tlin=tstrat(1);
    end
    if (tlin>tlmin)
        ml=qload/((tlin-tsup)*c);
        ql=ml*(tlin-tsup)*c;
    else
        ml=qload/((tlmin-tsup)*c);
        ql=ml*(tlmin-tsup)*c;
    end
end

%Just a reminder of what each temperature refers to

%ta= temperature of collector
%tam= mean temperature
%dta=temperature variation
%tfin = inlet temperature of fluid
%tfout= outlet temperature of fluid
%tstag= temperature of storage
%tst= initial storage temperature

if (loopNr==0)
tst=300;
end

if (loopNr==-1)
qs=0;
end

```



```

%ambient temperature variation during the day
ta=tam+dta*sin(-pi*(floor(nday)-nday));

%initialization for stratified storage
if (itypest==1)
    tfin=tst;
elseif (itypest==0)
    tfin=tstrat(layers);
    tlin=tstrat(1);
else
    tfin=tst;
end

%temperature of storage
tstag=ta+ta_eff*qs/b/qsmax*tam;

if (iheat==1) %tfout is different if we have heat exchanger%
    tfout=tfin-
    tlost*(fr_h*b/gq)+fr_h*tam/gq*(ta_eff*qs/qsref+b*ta/tam);
else
    tfout=tfin*(1-fr*b/gq)+fr*tam/gq*(ta_eff*qs/qsref+b*ta/tam);
end

if (ml==0) %when the flow rate of the load is zero, then the
temperature difference is also zero in order to avoid NAN
    dtl=0;
else
    dtl=ql/ml/c;
end

if (tstag-tfin>6)
    if (tst>tstmax)
        dc=0;
        if (itypest==0)
            if (tsttmp(1)<tstmax)
                dc=1;
            end
        end
    else
        if (tfout-tfin>2)
            dc=1;
        else
            dc=0;
        end
    end
else
    dc=0;
end

if (dtl>0)
    if (idhw==1) %if the system is dhw then the control about the
load is different
        if (itypest==1) %Mixed tank
            if (tst<tsup)
                dl=0;
            else
                dl=1;
            end
        else %Stratified tank
            if (tlin<tsup)

```

```

        dl=0;
    else
        dl=1;
    end
end
else
    if (itypest==1) %Mixed tank
        if (tst<tlmin)
            dl=0;
        else
            dl=1;
        end
    else
        if (tstrat(1)<tlmin)
            dl=0;
        else
            dl=1;
        end
    end
end
else
    dl=0;
end

%calculates usefull solar radiation%
qu=dc*mc*c*(tfout-tfin);

%if the system is dhw then the load outlet is different
if (idhw==1)
    tlout=tsup;
elseif (idhw==0)
    tlout=tlin-dtl;
end

%This area calculates what happens when the storage is mixed or
stratified
if (itypest==1)
    tlin=tst;
    qst=dc*qu-dl*ql-ust*ast*(tst-ta);
    tst=tst+(qst)/mst;
elseif (itypest==0)
    tlin=tstrat(1);
    frlcst=mc*c*dc/(mst/layers);
    frldst=ml*c*dl/(mst/layers);

    for i=1:layers

        sum_fm=0;
        sum_gm=0;
        fm(i)=0;
        gm(i)=0;

        if (i>1)

            if (tfout>tstrat(i) && tfout<tstrat(i-1))
                fm(i)=1;
            else
                fm(i)=0;
            end
        end
    end
end

```

```

        for j=1:(i-1)
            sum_fm=sum_fm+fm(j);
        end

        cld(i)=(tstrat(i-1)-tstrat(i))*sum_fm;

    else

        if (tfout>tstrat(i))
            fm(i)=1;
        else
            fm(i)=0;
        end
        sum_fm=0;
        cld(i)=0;

    end

    if (i<layers)
        if (tlout<tstrat(i) && tlout>tstrat(i+1))
            gm(i)=1;
        else
            gm(i)=0;
        end

        for j=(i+1):layers
            sum_gm=sum_gm+gm(j);
        end

        ldd(i)=(tstrat(i+1)-tstrat(i))*sum_gm;

    else

        if (tlout<tstrat(i))
            gm(i)=1;
        else
            gm(i)=0;
        end

        sum_gm=0;
        ldd(i)=0;

    end

    tsttmp(i)=tstrat(i)+frlcst*(fm(i)*(tfout-
tstrat(i))+cld(i))+frldst*(gm(i)*(tlout-tstrat(i))+ldd(i))-
ust*(ast/layers)*(tstrat(i)-ta)/(mst/layers);
    qst=qst+mst*(tsttmp(i)-tstrat(i))/layers;
    tstrat(i)=tsttmp(i);
end
    tst=tst+qst/mst;
end

sum_l= sum_l + ql;
sum_s=sum_s+qs*a;
sum_us=sum_us+qu;

%n calculation

```

```

if (qs>0)
    n=qu/(qs*a);
else
    n=0;
end

%fraction calculation
if (sum_us/sum_l<1)
    f=((sum_us/sum_l));
else
    f=1;
end

qlyear=sum_l*365*24/1000;

%case 1
%P,i calculation
ieff=(i-j)/(1+j);
P= gw * (((1+ieff)^ni)-1)/(ieff*((1+ieff)^ni));
rrq=P-(colcost+instcost);

%case 2
pinv=csc*a+cin;
ge=f*qlyear*ce;
gp=f*qlyear*cp;

pp=gp*((1+iefp)^years)-1/(iefp*((1+iefp)^years));
pe=ge*((1+iefe)^years)-1/(iefe*((1+iefe)^years));

loopNr=loopNr+1;

if mod(loopNr,360)==0
    cell=cell+1;

    TST(1,cell)=tst;
    QU(1,cell)=qu/1000;
    N_A(1,cell)=n*100;
    QS(1,cell)=qs/1000;
    if (itypest==0)
        TST_str(1,cell)=tstrat(1);
    end

    vec(cell,1)=tlocal;      %a
    vec(cell,2)=tst;        %b
    vec(cell,3)=tfin;       %c
    vec(cell,4)=tfout;      %d
    vec(cell,5)=qst;        %e
    vec(cell,6)=qu;         %f
    vec(cell,7)=dtl*ml*c;   %g
    vec(cell,8)=n;          %h
    vec(cell,9)=ql;         %i
    vec(cell,10)=qs;        %j
    vec(cell,11)=tlin;      %k
    vec(cell,12)=tlout;     %l
    vec(cell,13)=dc;        %m
    vec(cell,14)=dl;        %n
    vec(cell,15)=ta;        %o
    vec(cell,16)=ml;        %p
    vec(cell,17)=mc;        %q

```

```

vec(cell,28)=omega;
vec(cell,29)=rrq;

if (itypest==0)
vec(cell,18)=tstrat(1);
vec(cell,19)=tstrat(2);
vec(cell,20)=tstrat(3);
vec(cell,21)=tstrat(4);
vec(cell,22)=tstrat(5);
vec(cell,23)=tsttmp(1);
vec(cell,24)=tsttmp(2);
vec(cell,25)=tsttmp(3);
vec(cell,26)=tsttmp(4);
vec(cell,27)=tsttmp(5);
end

end

%time increment
nday=nday+timestep;
count_timestep=count_timestep+timestep;
waitbar(nday/(day+range), wait);
end

close(wait)
varname={'tlocal','tst','tfin','tfout','qst','qu','dtl*ml*c','n','q
l','qs','tlin','tlout','dc','dl','ta','ml','mc','tsrtat1','tstrat2'
,'tstrat3','tstrat4','tstrat5','tsttmp1','tsttmp2','tsttmp3','tsttm
p4','tsttmp5','omega','rrq'};
xlRange = 'A1:AC1';
xlswrite ('SOLSYS',varname,xlRange);
xlRange = 'A2:AC60000';
xlswrite ('SOLSYS',vec,xlRange);
xlswrite ('radiation',ve);
set(handles.fText,'String',num2str(f));

hax=handles.Tst_plot;
plot(hax,x,TST,x,QU);
xlabel(hax,'t (hours x 10)');
ylabel(hax,'Tst (K)');
grid(hax,'on');
end

function
qsmax=qmax(ndaymax,ktmax,correctionmax,omega_correctionmax,phimax,g
mrmax,betamax,rrmax,irvmax)

%delta angle for the day
deltamax=23.45*sind(360*((284+floor(ndaymax))/365));

%max omega angle%
omegamax=180+omega_correctionmax; %set it up for multiple days

%max as calculation%

```

```

asmax=(asind(sind(deltamax)*sind(phimax)+cosd(deltamax)*cosd(phimax)
)*cosd(omegamax));

%max gamma s angle%
gsmmax=asind(cosd(deltamax)*sind(omegamax)/cosd(asmax));

%max theta angle%
thetamax=acosd(sind(asmax)*cosd(betamax)+cosd(asmax)*sind(betamax)*
cosd(gsmmax-gmrmax));

%solar stable
isc=1367;

%max extraterrestrial solar radiation
iomax=isc*(1+0.034*cosd((360*(ndaymax))/365));
homax=iomax*sind(asmax);
if (irvmax==0)

    hmax=ktmax*homax;

    if (ktmax<= 0.85 && ktmax>=0.30)
        ibmax=-520+1800*ktmax;
    elseif (ktmax<0.3)
        ibmax=0;
    else
        ibmax=0;
    end

    hbmax=ibmax*sind(asmax);
    hdmax=hmax-hbmax;

else

    hmax=ktmax*homax;

    if (ktmax<0.35)
        hdmax=(1-0.249*ktmax)*hmax;
    elseif (ktmax>=0.35 && ktmax<=0.75)
        hdmax=(1.557-1.84*ktmax)*hmax;
    elseif (kt<0.85)
        hdmax=0.177*hmax;
    else
        hdmax=0;
        hmax=0;
    end
    ibmax=(hmax-hdmax)/sind(asmax);

end

%calculates the maximum solar radiation%
qsmmax=(ibmax*cosd(thetamax)+(hdmax)*(cosd(betamax/2)^2)+hmax*rrmax*
(sind(betamax/2)^2));

end

```

B. Πηγαίος κώδικας γραφικού περιβάλλοντος ΗΘΣ

```
function varargout = Solsys(varargin)
% SOLSYS M-file for Solsys.fig
%     SOLSYS, by itself, creates a new SOLSYS or raises the
existing
%     singleton*.
%
%     H = SOLSYS returns the handle to a new SOLSYS or the handle
to
%     the existing singleton*.
%
%     SOLSYS('CALLBACK', hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in SOLSYS.M with the given input
arguments.
%
%     SOLSYS('Property','Value',...) creates a new SOLSYS or
raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Solsys_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Solsys_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Solsys

% Last Modified by GUIDE v2.5 05-Jan-2018 02:00:25

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Solsys_OpeningFcn, ...
                  'gui_OutputFcn',  @Solsys_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Solsys is made visible.
function Solsys_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to Solsys (see VARARGIN)

% Choose default command line output for Solsys
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Solsys wait for user response (see UIRESUME)

%SolsysF(handles) %Only if we want the program to start when we
start the
%gui

% --- Outputs from this function are returned to the command line.
function varargout = Solsys_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function depthText_Callback(hObject, eventdata, handles)
% hObject handle to depthText (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of depthText as
text
% str2double(get(hObject,'String')) returns contents of
depthText as a double
astr=get(handles.depthText,'String');
a=str2double(astr);
set(handles.depthSlider,'Value',a);

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function depthText_CreateFcn(hObject, eventdata, handles)
% hObject handle to depthText (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function depthSlider_Callback(hObject, eventdata, handles)
% hObject handle to depthSlider (see GCBO)

```



```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine
range of slider
a=get(handles.depthSlider,'Value');
astr=num2str(a);
set(handles.depthText,'String',astr)
%SolsysF(handles);

% --- Executes on slider movement.
function mstslider_Callback(hObject, eventdata, handles)
% hObject handle to mstslider (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine
range of slider
mst=get(handles.mstslider,'Value');
mststr=num2str(mst);
set(handles.textmst,'String',mststr)

% --- Executes during object creation, after setting all
properties.
function mstslider_CreateFcn(hObject, eventdata, handles)
% hObject handle to mstslider (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function textmst_Callback(hObject, eventdata, handles)
% hObject handle to textmst (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of textmst as text
%         str2double(get(hObject,'String')) returns contents of
textmst as a double
mststr=get(handles.textmst,'String');
mst=str2double(mststr);
set(handles.mstslider,'Value',mst)
% --- Executes during object creation, after setting all
properties.
function textmst_CreateFcn(hObject, eventdata, handles)
% hObject handle to textmst (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

```

```

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%SolsysF(handles);
% --- Executes during object creation, after setting all
properties.
function depthSlider_CreateFcn(hObject, eventdata, handles)
% hObject    handle to depthSlider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes during object creation, after setting all
properties.
function Input_mst_slid_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Input_mst_slid (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Input_mst_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Input_mst (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Input_b_Callback(hObject, eventdata, handles)
% hObject    handle to Input_b (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Input_b as text
% str2double(get(hObject,'String')) returns contents of
Input_b as a double
bstr=get(handles.Input_b,'String');
b=str2double(bstr);
set(handles.Input_b,'Value',b);
%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Input_b_CreateFcn(hObject, eventdata, handles)
% hObject handle to Input_b (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function Input_taeff_Callback(hObject, eventdata, handles)
% hObject handle to Input_taeff (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Input_taeff as
text
% str2double(get(hObject,'String')) returns contents of
Input_taeff as a double
taeffstr=get(handles.Input_taeff,'String');
taeff=str2double(taeffstr);
set(handles.Input_taeff,'Value',taeff);

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Input_taeff_CreateFcn(hObject, eventdata, handles)
% hObject handle to Input_taeff (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function Input_q_Callback(hObject, eventdata, handles)
% hObject handle to Input_q (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Input_q as text
% str2double(get(hObject,'String')) returns contents of
Input_q as a double
qstr=get(handles.Input_q,'String');
q=str2double(qstr);
set(handles.Input_q,'Value',q);

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Input_q_CreateFcn(hObject, eventdata, handles)
% hObject handle to Input_q (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function Input_fr_Callback(hObject, eventdata, handles)
% hObject handle to Input_fr (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Input_fr as text
% str2double(get(hObject,'String')) returns contents of
Input_fr as a double
frstr=get(handles.Input_fr,'String');
fr=str2double(frstr);
set(handles.Input_fr,'Value',fr);

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Input_fr_CreateFcn(hObject, eventdata, handles)
% hObject handle to Input_fr (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in collector.
function collector_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to collector (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
collector contents as cell array
%          contents{get(hObject,'Value')} returns selected item from
collector
c_v = get(handles.collector,'Value');

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function collector_CreateFcn(hObject, eventdata, handles)
% hObject    handle to collector (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Input_dta_Callback(hObject, eventdata, handles)
% hObject    handle to Input_dta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Input_dta as
text
%          str2double(get(hObject,'String')) returns contents of
Input_dta as a double
dtastr=get(handles.Input_dta,'String');
dta=str2double(dtastr);
set(handles.Input_dta,'Value',dta);

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Input_dta_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Input_dta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function Input_tam_Callback(hObject, eventdata, handles)
% hObject      handle to Input_tam (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Input_tam as
text
%           str2double(get(hObject,'String')) returns contents of
Input_tam as a double
tamstr=get(handles.Input_tam,'String');
tam=str2double(tamstr);
set(handles.Input_tam,'Value',tam);

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Input_tam_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Input_tam (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Input_trise_Callback(hObject, eventdata, handles)
% hObject      handle to Input_trise (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Input_trise as
text
%           str2double(get(hObject,'String')) returns contents of
Input_trise as a double
trisestr=get(handles.Input_trise,'String');
trise=str2double(trisestr);
set(handles.Input_trise,'Value',trise);

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Input_trise_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Input_trise (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');

```

```

end

function Input_td_Callback(hObject, eventdata, handles)
% hObject    handle to Input_td (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Input_td as text
%         str2double(get(hObject,'String')) returns contents of
Input_td as a double
tdstr=get(handles.Input_td,'String');
td=str2double(tdstr);
set(handles.Input_td,'Value',td);

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Input_td_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Input_td (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Input_maxi_Callback(hObject, eventdata, handles)
% hObject    handle to Input_maxi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Input_maxi as
text
%         str2double(get(hObject,'String')) returns contents of
Input_maxi as a double
maxistr=get(handles.Input_maxi,'String');
maxi=str2double(maxistr);
set(handles.Input_maxi,'Value',maxi);

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Input_maxi_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Input_maxi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Input_ust_Callback(hObject, eventdata, handles)
% hObject    handle to Input_ust (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Input_ust as
text
%         str2double(get(hObject,'String')) returns contents of
Input_ust as a double
uststr=get(handles.Input_ust,'String');
ust=str2double(uststr);
set(handles.Input_ust,'Value',ust);

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Input_ust_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Input_ust (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Input_ast_Callback(hObject, eventdata, handles)
% hObject    handle to Input_ast (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Input_ast as
text
%         str2double(get(hObject,'String')) returns contents of
Input_ast as a double
aststr=get(handles.Input_ast,'String');
ast=str2double(aststr);
set(handles.Input_ast,'Value',ast);

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Input_ast_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Input_ast (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```



```

% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Input_tlmin_Callback(hObject, eventdata, handles)
% hObject    handle to Input_tlmin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Input_tlmin as
text
%       str2double(get(hObject,'String')) returns contents of
Input_tlmin as a double
tlminstr=get(handles.Input_tlmin,'String');
tlmin=str2double(tlminstr);
set(handles.Input_tlmin,'Value',tlmin);

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Input_tlmin_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Input_tlmin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Input_ml_Callback(hObject, eventdata, handles)
% hObject    handle to Input_ml (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Input_ml as text
%       str2double(get(hObject,'String')) returns contents of
Input_ml as a double
mlstr=get(handles.Input_ml,'String');
ml=str2double(mlstr);
set(handles.Input_ml,'Value',ml);

%SolsysF(handles);

```

```

% --- Executes during object creation, after setting all
properties.
function Input_ml_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Input_ml (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function Input_ql_Callback(hObject, eventdata, handles)
% hObject    handle to Input_ql (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of Input_ql as text
%       str2double(get(hObject, 'String')) returns contents of
Input_ql as a double
qlstr=get(handles.Input_ql, 'String');
ql=str2double(qlstr);
set(handles.Input_ql, 'Value', ql);

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Input_ql_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Input_ql (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in Start.
function Start_Callback(hObject, eventdata, handles)
% hObject    handle to Start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%uiresume(handles.figure1);
SolsysF(handles);
%PlotF(handles);

function Year_input_Callback(hObject, eventdata, handles)
% hObject    handle to Year_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of Year_input as
text
%         str2double(get(hObject,'String')) returns contents of
Year_input as a double

% --- Executes during object creation, after setting all
properties.
function Year_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Year_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Month_input_Callback(hObject, eventdata, handles)
% hObject    handle to Month_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Month_input as
text
%         str2double(get(hObject,'String')) returns contents of
Month_input as a double
month=get(handles.Month_input,'String');
monthstr=str2double(month);
set(handles.Month_input,'Value',monthstr)

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Month_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Month_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Range_input_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to Range_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Range_input as
text
%          str2double(get(hObject,'String')) returns contents of
Range_input as a double
range=get(handles.Range_input,'String');
rangestr=str2double(range);
set(handles.Range_input,'Value',rangestr)

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function Range_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Range_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in City.
function City_Callback(hObject, eventdata, handles)
% hObject    handle to City (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns City
contents as cell array
%          contents{get(hObject,'Value')} returns selected item from
City
city = get(handles.City,'Value');

%SolsysF(handles);

% --- Executes during object creation, after setting all
properties.
function City_CreateFcn(hObject, eventdata, handles)
% hObject    handle to City (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popmenu controls usually have a white background on
Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function Day_input_Callback(hObject, eventdata, handles)
% hObject      handle to Day_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
day=get(handles.Day_input,'String');
daystr=str2double(day);
set(handles.Day_input,'Value',daystr);
%SolsysF(handles);
% Hints: get(hObject,'String') returns contents of Day_input as
text
%          str2double(get(hObject,'String')) returns contents of
Day_input as a double

% --- Executes during object creation, after setting all
properties.
function Day_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Day_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in DHW.
function DHW_Callback(hObject, eventdata, handles)
% hObject      handle to DHW (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%DHW_check=get(handles.DHW, 'Value');
%SolsysF(handles);
% Hint: get(hObject,'Value') returns toggle state of DHW

% --- Executes on button press in Real_env.
function Real_env_Callback(hObject, eventdata, handles)
% hObject      handle to Real_env (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Real_env_check=get(handles.Real_env, 'Value');

%SolsysF(handles);
% Hint: get(hObject,'Value') returns toggle state of Real_env

% --- Executes on button press in Heat_exch.
function Heat_exch_Callback(hObject, eventdata, handles)
% hObject      handle to Heat_exch (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)
%Heat_exch=get(handles.Heat_exch, 'Value');

%SolsysF(handles);
% Hint: get(hObject,'Value') returns toggle state of Heat_exch

% --- Executes on button press in Time_load.
function Time_load_Callback(hObject, eventdata, handles)
% hObject      handle to Time_load (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Time_load=get(handles.Time_load, 'Value');

%SolsysF(handles);
% Hint: get(hObject,'Value') returns toggle state of Time_load

% --- Executes on button press in Stratified.
function Stratified_Callback(hObject, eventdata, handles)
% hObject      handle to Stratified (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
Stratified_check=get(handles.Stratified, 'Value');

%SolsysF(handles);
% Hint: get(hObject,'Value') returns toggle state of Stratified

% --- Executes on button press in Gui_data.
function Gui_data_Callback(hObject, eventdata, handles)
% hObject      handle to Gui_data (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of Gui_data
%Gui_check=get(handles.Gui_data, 'Value');

%SolsysF(handles);

% --- Executes on selection change in Plotting.
function Plotting_Callback(hObject, eventdata, handles)
% hObject      handle to Plotting (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns Plotting
contents as cell array
%          contents{get(hObject,'Value')} returns selected item from
Plotting
each_plot = get(handles.Plotting, 'Value');
%data=guidata(gcf)
hax=handles.Tst_plot;
global x;
global TST;
global N_A;
global QU;
global QS;

```

```

global TST_str;
global itypest;

if (each_plot==1)
    if (itypest==1)
        hax=handles.Tst_plot;
        plot(hax,x,TST);
        %axis(hax,[xmin, xmax, 250, 400]);
        xlabel(hax,'t (hours X 10)');
        ylabel(hax,'Tst(K)');
        grid(hax,'on');
    elseif (itypest==0)
        hax=handles.Tst_plot;
        plot(hax,x,TST_str);
        %axis(hax,[xmin, xmax, 250, 400]);
        xlabel(hax,'t (hours X 10)');
        ylabel(hax,'Tst(K)');
        grid(hax,'on');
    end
elseif (each_plot==2)
    hax=handles.Tst_plot;
    plot(hax,x,QU);
    %axis(hax2,[xmin, xmax, 0, 100]);
    xlabel(hax,'t (hours X 10)');
    ylabel(hax,'Qu(kW)');
    grid(hax,'on');
elseif (each_plot==3)
    %hax3=handles.n_plot;
    plot(hax,x,N_A);
    %axis(hax3,[xmin, xmax, 0, 100]);
    xlabel(hax,'t (hours X 10)');
    ylabel(hax,'n(%)');
    grid(hax,'on');
elseif (each_plot==4)
    %hax3=handles.n_plot;
    plot(hax,x,QS);
    %axis(hax3,[xmin, xmax, 0, 100]);
    xlabel(hax,'t (hours X 10)');
    ylabel(hax,'Qs(kW)');
    grid(hax,'on');
end

% --- Executes during object creation, after setting all
properties.
function Plotting_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Plotting (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Degrees days.

```

```

function Degrees_days_Callback(hObject, eventdata, handles)
% hObject    handle to Degrees_days (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of Degrees_days

function Uresidence_Callback(hObject, eventdata, handles)
% hObject    handle to Uresidence (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Uresidence as
text
%         str2double(get(hObject,'String')) returns contents of
Uresidence as a double
Uresidencst=get(handles.Uresidence,'String');
Uresidencetr=str2double(Uresidencst);
set(handles.Uresidence,'Value',Uresidencetr)

% --- Executes during object creation, after setting all
properties.
function Uresidence_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Uresidence (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Aresidence_Callback(hObject, eventdata, handles)
% hObject    handle to Aresidence (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Aresidence as
text
%         str2double(get(hObject,'String')) returns contents of
Aresidence as a double
Aresidencst=get(handles.Aresidence,'String');
Aresidencetr=str2double(Aresidencst);
set(handles.Aresidence,'Value',Aresidencetr)

% --- Executes during object creation, after setting all
properties.
function Aresidence_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Aresidence (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```



```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in Soiltype.
function Soiltype_Callback(hObject, eventdata, handles)
% hObject    handle to Soiltype (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns Soiltype
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
Soiltype
soiltype= get(handles.Soiltype,'Value');

% --- Executes during object creation, after setting all
properties.
function Soiltype_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Soiltype (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in Reflectivity.
function Reflectivity_Callback(hObject, eventdata, handles)
% hObject    handle to Reflectivity (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
Reflectivity contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
Reflectivity
refl= get(handles.Reflectivity,'Value');

% --- Executes during object creation, after setting all
properties.
function Reflectivity_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Reflectivity (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

```

% Hint: popmenu controls usually have a white background on
Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function Time_dependent_load_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Time_dependent_load (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
% set the tables 'Data' property to a cell array of empty matrices.
% The size of the cell array determines the number of rows and
columns in the table.
Time_data=[5.9 14.4 20.0 23.8 20.3 12.5 10.6 13.8 8.4 6.3 6.0 10.0
20.2 32.0 26.4 20.7 14.5 14.2 0.3 5.9 0.0 0.0 0.0 0.0].';
set(hObject, 'Data', Time_data);
set(hObject, 'RowName', [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24], 'ColumnName', {'Flow rate in kg/h'});

% --- Executes when entered data in editable cell(s) in
Time_dependent_load.
function Time_dependent_load_CellEditCallback(hObject, eventdata,
handles)
% hObject    handle to Time_dependent_load (see GCBO)
% eventdata  structure with the following fields (see
MATLAB.UI.CONTROL.TABLE)
%   Indices: row and column indices of the cell(s) edited
%   PreviousData: previous data for the cell(s) edited
%   EditData: string(s) entered by the user
%   NewData: EditData or its converted form set on the Data
property. Empty if Data was not changed
%   Error: error string when failed to convert EditData to
appropriate value for Data
% handles    structure with handles and user data (see GUIDATA)
iRowChanged=eventdata.Indices(1);
iColChanged=eventdata.Indices(2);
oldValue=eventdata.PreviousData;
newValue=eventdata.NewData;
disp(['Data at (', num2str(iRowChanged), ', ', ...
num2str(iColChanged), ')', ' changed from ', ...
num2str(oldValue), ' to ', num2str(newValue)]);

function angle_b_Callback(hObject, eventdata, handles)
% hObject    handle to angle_b (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of angle_b as text
%        str2double(get(hObject,'String')) returns contents of
angle_b as a double

```

```

angle_bst=get(handles.angle_b, 'String');
angle_btr=str2double(angle_bst);
set(handles.angle_b, 'Value', angle_btr)

% --- Executes during object creation, after setting all
properties.
function angle_b_CreateFcn(hObject, eventdata, handles)
% hObject    handle to angle_b (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function angle_g_Callback(hObject, eventdata, handles)
% hObject    handle to angle_g (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of angle_g as text
%       str2double(get(hObject, 'String')) returns contents of
angle_g as a double
angle_gst=get(handles.angle_g, 'String');
angle_gtr=str2double(angle_gst);
set(handles.angle_g, 'Value', angle_gtr)

% --- Executes during object creation, after setting all
properties.
function angle_g_CreateFcn(hObject, eventdata, handles)
% hObject    handle to angle_g (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

Γ. Πίνακες του ΤΕΕ

Πίνακας Γ.1 Τοπολογικά δεδομένα για διάφορες τοποθεσίες της Ελλάδας.

ΜΗΝΑΣ	Τοπολογικά δεδομένα		
	ΓΕΩΓΡΑΦΙΚΟ ΠΛΑΤΟΣ	ΓΕΩΓΡΑΦΙΚΟ ΜΗΚΟΣ	ΥΨΟΜΕΤΡΟ
Αθήνα(Ελληνικό)	370 54'	230 45'	15
Αθήνα(Φιλαδέλφεια)	380 03'	230 40'	138
Αγρίνιο	380 37'	210 23'	25
Αγχίαλος	390 13'	220 48'	15,3
Αλεξανδρούπολη	400 51'	250 56'	3,5
Αλίαρτος	380 23'	230 06'	110
Ανδραβίδα	370 55'	210 17'	15,1
Άραξος	380 09'	210 25'	11,5
Άργος	370 36'	220 47'	11,2
Αργοστόλι	380 11'	200 29'	22
Άρτα	390 10'	210 00'	10,5
Δράμα	410 09'	240 09'	104
Έδεσσα	400 58'	220 03'	30
Ζάκυνθος	370 47'	200 54'	7,9
Ηράκλειο	350 20'	250 11'	39,3
Θεσσαλονίκη	400 31'	220 58'	4,8
Ιεράπετρα	350 00'	250 44'	10
Ιωάννινα	390 42'	200 49'	484
Καλαμάτα	370 04'	220 00'	11,1
Καρδίτσα	390 22'	200 48'	111,1
Καρπενήσι	380 54'	210 47'	1001
Κάρυστος	380 01'	240 25'	10
Καστοριά	400 27'	210 17'	660,9
Κέρκυρα	390 37'	190 55'	4
Κοζάνη	400 18'	210 47'	625
Κομοτηνή	410 07'	250 24'	30
Κόνιτσα	400 03'	200 45'	542
Κόρινθος			
Κύθηρα	360 17'	230 10'	316,6
Κως	360 47'	270 04'	129
Λαμία	380 51'	220 24'	17,4
Λάρισα	390 39'	220 27'	73,6
Λευκάδα	380 50'	200 43'	1
Λήμνος	390 55'	250 14'	4,6
Μεθώνη	360 50'	210 42'	33
Μήλος	360 43'	240 27'	182
Μυτιλήνη	390 04'	260 36'	4
Νάξος	370 06'	250 23'	9,8
Ξάνθη	410 08'	240 53'	43
Πάρος	370 01'	250 08'	33,5
Πάτρα	380 15'	210 44'	1
Πολύγυρος	400 23'	230 26'	545
Πύργος	370 40'	210 18'	12
Ρέθυμνο	350 21'	240 31'	7
Ρόδος	360 24'	280 07'	11,5
Σάμος	370 42'	260 55'	7,3
Σέρρες	410 05'	230 34'	34,5
Σητεία	350 12'	260 06'	115,6
Σκύρος	380 54'	240 33'	17,9
Σούδα	350 33'	240 07'	151,6
Σπάρτη	370 04'	220 25'	212
Σύρος	370 25'	240 57'	72
Τανάγρα	380 19'	230 33'	140,1
Τρίκαλα(Ημαθίας)	390 33'	210 46'	114
Τρίκαλα(Θεσσαλίας)	400 36'	220 33'	0,8
Τρίπολη	370 32'	220 24'	650,9
Τυμπάκιο	350 00'	240 46'	6,7
Φλώρινα	400 48'	210 26'	617
Χαλκίδα	380 28'	230 36'	5
Χανιά	350 29'	240 07'	150
Χίος	380 28'	260 08'	5
Καβάλα	400 54'	240 36'	5,4

Πίνακας Γ.2 Μέση μηνιαία θερμοκρασία εικοσιτετραώρου για διάφορες τοποθεσίες της Ελλάδας.

ΜΗΝΑΣ	Μέση μηνιαία θερμοκρασία 24ώρου [οC]											
	ΙΑΝ	ΦΕΒ	ΜΑΡ	ΑΠΡ	ΜΑΙ	ΙΟΥΝ	ΙΟΥΛ	ΑΥΓ	ΣΕΠ	ΟΚΤ	ΝΟΕ	ΔΕΚ
Αθήνα (Ελληνικό)	10,3	10,6	12,3	16	20,7	25,4	28,1	28	24,3	19,6	15,4	12
Αθήνα (Φιλαδέλφεια)	10,3	10,6	12,3	16	20,7	25,4	28,1	28	24,3	19,6	15,4	12
Αργίριο	8,3	9,2	11,5	15,2	20,4	24,8	27,2	27	23	18	13,2	9,6
Αρχαίος	6,6	7,7	10	14,3	19,6	24,6	26,9	26,3	22,2	17	12,1	8,1
Αλεξανδρούπολη	5	5,9	8,3	13,2	18,4	23,2	25,9	25,6	21,1	15,7	10,8	7
Αλιάρτος	7,1	8,2	10,6	15,2	20,6	25,7	27,2	26,2	22,6	16,9	12	8,6
Ανδραβίδα	9,4	9,9	11,8	14,8	19,5	23,7	25,9	26	22,9	18,3	14,1	10,9
Άραξος	10,2	10,5	12,2	15,2	19,8	24,1	26,6	26,8	23,4	19	14,7	11,6
Άργος	8,1	8,4	10,6	14,7	20	24,9	27,3	26,5	22,6	17,8	12,9	9,5
Αργοστόλι	11,4	11,5	12,8	15,3	19,5	23,5	25,7	26,1	23,4	19,8	15,8	12,8
Άρτα	8	9	11,5	14,6	19,6	23,6	25,7	25,6	22	17,8	12,5	9,1
Δράμα	4,7	6,5	10	14,7	20,2	24,7	26,7	25,9	22,3	16,4	9,9	6
Έδεσσα	4,5	6,2	9,6	13,4	18,6	23,8	25,2	24,5	20,7	15,6	9,4	5,8
Ζάκυνθος	10,5	10,5	11,8	14,8	19,7	24,3	27,2	27,1	23,6	19,1	14,8	11,7
Ηράκλειο	12,1	12,2	13,5	16,5	20,3	24,4	26,2	26,1	23,6	20,1	16,7	13,7
Θεσσαλονίκη	5,3	6,8	9,8	14,3	19,7	24,5	26,8	26,2	21,9	16,3	11,1	6,9
Ιεράπετρα	12,9	12,9	14,2	17	20,9	25,4	27,8	27,7	24,9	21	17,5	14,5
Ιωάννινα	4,7	6	8,8	12,4	17,5	22	24,9	24,5	20,1	15	9,7	5,8
Καλαμάτα	10,2	10,6	12,3	15,2	19,8	24,2	26,5	26,3	23,2	19	14,8	11,6
Καρδίτσα	4,5	6,9	10,4	13,9	18	24,2	26,3	25,6	22,1	16,1	10,1	4,3
Καρπενήσι	3,8	3,1	5,4	10,6	14,7	18,9	21,6	20,9	17,6	12,4	6,8	4,8
Κάρυστος	10,4	10,3	12,4	15,7	19,5	24,2	26,8	26,4	23,6	19,4	14,9	11,7
Καστοριά	2,2	3,4	6,9	11,5	16,4	21,4	24	23,2	18,9	13,4	7,2	3
Κέρκυρα	9,7	10,3	12	15	19,8	24	26,5	26,5	22,7	18,5	14,3	11,1
Κοζάνη	2,3	3,7	6,9	11,6	16,8	21,5	24,1	23,6	19,3	13,5	8	3,9
Κομοτηνή	4,8	6,2	8,6	13,1	18,4	23	25,5	25	20,6	15,2	10,8	7
Κόνιτσα	5,2	6,5	9,5	12,2	17,2	21,7	24,4	24	20,3	15,4	9,8	6,4
Κόρινθος	8,8	9,3	11,5	15,4	20,7	25,8	28,3	27,8	23,4	18,6	13,4	10,1
Κύθηρα	10,9	10,9	11,9	14,6	18,9	23,2	25,7	25,7	22,9	19,1	15,8	12,7
Κως	11	10,5	12,1	15,4	19,5	23,8	25,9	25,4	23,2	19,4	15	12,4
Λαμία	7,1	8,1	10,7	15	20,2	25,4	27	26	22,5	17	11,9	8,2
Λάρισα	5,2	6,8	9,5	14	19,7	25,2	27,3	26,3	21,9	16,3	10,9	6,5
Λευκάδα	10,2	10,6	12,7	15,2	19,4	23,1	25,4	25,5	23	19,3	14,6	11,5
Λήμιος	7,4	7,8	9,7	13,8	18,5	23,6	25,9	25,1	21,5	16,8	12,3	9
Μεθώνη	11,3	11,5	12,9	15,4	19	22,6	24,8	25,7	23,6	19,8	16	12,9
Μήλος	10,7	10,8	11,9	15	19,4	23,6	25,2	24,9	22,3	18,8	15,3	12,4
Μυτιλήνη	9,5	9,9	11,6	15,6	20,2	24,7	26,6	26,1	22,9	18,5	14,3	11,3
Νάξος	12,1	12,2	13,3	16,1	19,5	23,3	24,9	24,8	22,8	19,6	16,3	13,6
Ξάνθη	5,6	6,8	9,6	14,3	19,8	24,1	26,6	26	22,4	16,5	11	6,9
Πάρος	11,2	11,2	12,9	16,2	19,8	24	25,5	25	22,8	19,1	15,2	12,3
Πάτρα	10	10,6	12,5	15,6	20,1	24,1	26,4	26,7	23,5	19	14,5	11,4
Πολύγυρος	4,9	4,7	8,7	12,4	16,3	22,4	24	24,1	21,3	15,1	10,7	6,8
Πύργος	9,6	10,1	12,2	15,2	19,7	23,9	26,4	26,3	23	18,7	14,1	11
Ρέθυμνο	12,8	12,9	14,2	17,1	20,7	24,9	26,9	26,8	24,2	20,6	17,3	14,5
Ρόδος	12	12,2	13,7	16,6	20,6	24,8	26,9	27,1	24,7	20,9	16,7	13,5
Σάμος	10,4	10,2	12,2	16,1	20,8	25,7	28,6	28,2	24,4	19,6	14,7	12
Σέρρες	4	6,3	9,7	14,4	19,7	24,4	26,5	25,6	21,7	15,7	9,4	4,8
Σητεία	12,2	12,3	13,6	16,6	20,3	24,2	25,9	25,7	23,6	20,2	16,8	13,8
Σκύρος	9,9	10,2	11,5	15	19,3	23,7	25,5	25,1	22,2	18,3	14,6	11,6
Σούδα	10,9	11	12,6	15,8	20,1	24,6	26,6	26,1	23,1	19,3	15,6	12,6
Σπάρτη	9,5	10,1	12,3	15,6	20,9	25,9	28,2	27,4	24,2	19,3	14	10,7
Σύρος	11,8	11,8	13,1	16,4	20,3	25	26,7	26,5	24,1	20,1	15,9	12,8
Τανάγρα	7,5	8,2	10,2	14,5	20	25,3	27,5	26,7	22,4	17,2	12,8	9,4
Τρίκαλα (Ημαθίας)	4,7	6,1	9,6	14,4	19,7	24,3	25,9	25	21,1	15,8	9,7	5,5
Τρίκαλα (Θεσσαλίας)	5,4	6,9	10,6	15	20,6	25,6	27,4	26,1	22,6	16,4	10,2	6,4
Τρίπολη	5,1	5,8	7,9	11,7	17	22	24,5	24,1	20	14,6	10,1	6,7
Τυμνάκιο	11,7	11,8	13,4	16,4	20,7	24,9	27,8	27,5	24,4	20,3	16,4	13,2
Φλώρινα	0,5	2,7	6,7	11,6	16,8	21	23,1	22,5	18,4	12,6	7	2,2
Χαλκίδα	9,1	9,1	11,8	16,1	20,7	25,8	27,8	27,5	24,5	19,7	13,9	10,5
Χανιά	11,6	11,8	13,2	16,3	20,1	24,5	26,5	26,1	23,3	19,4	16,1	13,1
Χίος	9,6	9,7	11,6	15,1	19,6	24,1	26,4	25,9	22,7	18,1	13,6	11,1
Καβάλα	5,5	6,2	8,7	13,7	18,6	23,4	25,9	25,3	21	15,6	10,4	6,6

Πίνακας Γ.3 Μέση μηνιαία ηλιακή ακτινοβολία στο οριζόντιο επίπεδο για διάφορες τοποθεσίες της Ελλάδας.

ΜΗΝΑΣ	Μέση μηνιαία ολική ηλιακή ακτινοβολία στο οριζόντιο επίπεδο [kWh/(m ² .mo)]											
	ΙΑΝ	ΦΕΒ	ΜΑΡ	ΑΠΡ	ΜΑΙ	ΙΟΥΝ	ΙΟΥΛ	ΑΥΓ	ΣΕΠ	ΟΚΤ	ΝΟΕ	ΔΕΚ
Αθήνα(Ελληνικό)	63	79	117,7	154,3	195,4	214	222,4	202,7	152,6	109	70,7	55,7
Αθήνα(Φιλαδέλφεια)	63,3	77,7	118,9	152,7	190,4	207,4	214,5	198,6	156	111,1	68,1	54,4
Αργίριο	63,5	78,3	119,4	148,4	189,9	214,1	224,2	200,3	151,3	109,8	69,8	55,1
Αγχίαλος	61,3	74,3	112,5	149,2	189,7	212,7	217,4	195,1	146,8	98,8	63,1	51,5
Αλεξανδρούπολη	50,7	68,9	107,3	141,8	182,8	205,8	211,6	192,3	144,2	99,4	57,8	43,7
Αλιάρτος	51	70	114	158	206	216	220	204	153	102	66	49
Ανδραβίδα	68,4	83,4	128,4	159,5	200,2	220,6	228,4	205,5	156,1	115,6	75,2	60,1
Άραξος	62,6	78,6	119,7	155,6	196,1	210,9	217,8	197,6	152,4	110,2	69,4	54,6
Άργος	68,7	83,6	127,7	159,5	202,5	220,6	229	206,4	157,2	115,5	74,8	59,2
Αργοστόλι	65	80	124,9	157,3	204,3	219,4	226,1	203,1	155,6	112,6	72,6	56,1
Άρτα	65,5	79,7	120,4	149,1	190,2	211,2	218,1	196,4	150,6	110	69,5	56,2
Δράμα												
Έδεσσα												
Ζάκυνθος	64,2	77,6	110,1	158,8	190,8	200,1	218,5	203,8	154	104,3	65,4	52,8
Ηράκλειο	65,6	81,6	125	166,5	207,3	222,4	227,1	207	163	117,3	78,6	61,2
Θεσσαλονίκη	52,6	67,5	103,2	140,7	179,1	198,6	209,5	184,7	136,7	91,4	56,6	45,5
Ιεράπετρα	73	89	137	174	210	220	224	205	165	125	89	69
Ιωάννινα	51,8	66,4	105,2	134,9	178,3	202,1	212	190,3	136,5	96,1	57,6	45,1
Καλαμάτα	68,2	82,3	126,1	156,2	198,7	216	222	200,9	154,9	114,5	75,2	59,3
Καρδίτσα												
Καρπενήσι												
Κάρυστος												
Καστοριά	57,6	71,3	111,2	141,1	173,6	201,8	206,3	185,5	138,5	97	60	47,7
Κέρκυρα	57,7	73,5	116,7	149,9	195,4	213,6	221	197,8	148,2	103,1	64,4	50,7
Κοζάνη												
Κομοτηνή	50	65	105	145	188	209	215	193	145	99	58	45
Κόνιτσα												
Κόρινθος	65,4	82,8	123,4	157,9	201,7	218,3	223,2	201,9	154,2	111,9	72	55,2
Κύθηρα												
Κως												
Λαμία	59,4	73,1	113,9	150,5	188,8	210,3	214,1	193,4	145,5	100,3	65,3	52,1
Λάρισα	55,1	71,4	112,1	151,1	190,9	210,8	215,8	194,3	145,9	97,8	61,2	47,8
Λευκάδα												
Λήμνος	51,1	69,6	112,3	154,3	199,5	215,3	220,9	198,5	150,8	104,6	61,3	46
Μεθώνη	62	78	125	155	207	215	220	199	157	116	77	57
Μήλος	56	67	120	175	213	223	226	205	164	112	77	52
Μυτιλήνη	52	70	113	156	209	219	223	201	156	109	67	50
Νάξος	60,3	77	122,6	161,2	204,7	220,4	224,5	204,8	159,1	115,9	73,7	55,6
Ξάνθη												
Πάρος	60	80	125	168	211	220	223	202	160	117	75	58
Πάτρα	55	72	124	147	200	215	218	197	153	107	66	53
Πολύγυρος												
Πύργος	68,4	83,1	127,5	157,9	200,4	215,6	223,8	202,1	155	115,9	75,5	59,3
Ρέθυμνο	62	81	119	164	211	218	223	204	160	106	81	58
Ρόδος	69,9	85,1	130,8	164	203	217,2	225,1	204,3	158,9	120,2	79,2	61,2
Σάμος	64,9	82,1	126,7	162,5	206,8	224,9	230,6	209,6	163,7	120,5	78,6	58,5
Σέρρες	50,8	68	105,7	141	180,5	202,8	209,7	187,7	140,8	94,7	56,5	43,7
Σητεία	66,5	83	128,4	165,2	207,4	223,2	227,1	207,5	163,7	119,3	80,4	61,9
Σκύρος	51,2	69,1	109,9	153,3	197,2	214,2	219,7	198,8	151,7	102,5	62,9	47,7
Σούδα	65	81,7	130,7	166,5	208,5	221,9	228,5	209,3	163,6	116,3	76,8	60,3
Σπάρτη												
Σύρος	58	80	121	172	212	219	225	204	160	119	74	57
Τανάγρα	59,1	74,2	112,7	151,9	194	215,4	222	201,3	153,1	104,5	64,7	51,2
Τρίκαλα(Ημαθίας)	57,3	72,2	105,6	140,2	178	202,9	206,4	185,8	138,6	94	59,7	49,1
Τρίκαλα(Θεσσαλίας)												
Τρίπολη												
Τυμπάκιο	73,4	90,5	137,5	169	207,8	222,9	228,7	209,8	166,3	127,2	85,9	67,7
Φλώρινα												
Χαλκίδα												
Χανιά	62	80	124	167	212	220	225	205	161	111	78	59
Χίος	55	72	119	161	210	220	225	203	159	116	71	53
Καβάλα	57,5	78	111,3	137,6	189,9	204	208,8	187,6	141,8	97,7	62,1	43,3

Πίνακας Γ.4 Μέσος μηνιαίος συντελεστής αιθριότητας για διάφορες τοποθεσίες της Ελλάδας.

ΜΗΝΑΣ	Μέσος μηνιαίος συντελεστής αιθριότητας κτ											
	ΙΑΝ	ΦΕΒ	ΜΑΡ	ΑΠΡ	ΜΑΙ	ΙΟΥΝ	ΙΟΥΛ	ΑΥΓ	ΣΕΠ	ΟΚΤ	ΝΟΕ	ΔΕΚ
Αθήνα(Ελληνικό)	0,44	0,47	0,48	0,53	0,57	0,62	0,63	0,64	0,6	0,54	0,49	0,44
Αθήνα(Φιλαδέλφεια)	0,45	0,46	0,49	0,52	0,56	0,6	0,61	0,63	0,61	0,55	0,47	0,43
Αργίριο	0,46	0,47	0,49	0,51	0,55	0,62	0,64	0,63	0,59	0,55	0,49	0,44
Αγχιάλος	0,45	0,46	0,47	0,51	0,55	0,61	0,62	0,62	0,58	0,51	0,46	0,43
Αλεξανδρούπολη	0,4	0,44	0,46	0,49	0,54	0,59	0,6	0,61	0,58	0,53	0,44	0,39
Αλιάρτος	0,37	0,42	0,47	0,54	0,6	0,62	0,63	0,64	0,6	0,51	0,46	0,39
Ανδραβίδα	0,48	0,49	0,53	0,54	0,58	0,63	0,65	0,65	0,61	0,57	0,52	0,47
Άραξος	0,44	0,47	0,49	0,53	0,57	0,61	0,62	0,62	0,6	0,55	0,48	0,43
Άργος	0,48	0,49	0,52	0,54	0,59	0,64	0,65	0,65	0,61	0,57	0,51	0,46
Αργοστόλι	0,46	0,48	0,51	0,54	0,6	0,63	0,65	0,64	0,61	0,56	0,51	0,44
Άρτα	0,48	0,49	0,5	0,51	0,56	0,61	0,62	0,62	0,6	0,56	0,5	0,46
Δράμα												
Έδεσσα												
Ζάκυνθος	0,45	0,46	0,45	0,54	0,56	0,58	0,62	0,64	0,6	0,52	0,45	0,41
Ηράκλειο	0,42	0,45	0,49	0,56	0,6	0,64	0,65	0,64	0,62	0,55	0,5	0,43
Θεσσαλονίκη	0,41	0,43	0,44	0,49	0,52	0,57	0,6	0,59	0,55	0,48	0,43	0,4
Ιεράπετρα	0,46	0,49	0,54	0,58	0,61	0,64	0,64	0,64	0,62	0,58	0,56	0,48
Ιωάννινα	0,39	0,41	0,44	0,47	0,52	0,58	0,61	0,6	0,54	0,5	0,42	0,38
Καλαμάτα	0,47	0,48	0,51	0,53	0,58	0,62	0,63	0,63	0,6	0,56	0,5	0,45
Καρδίτσα												
Καρπενήσι												
Κάρυστος												
Καστοριά	0,45	0,45	0,47	0,49	0,51	0,58	0,59	0,59	0,56	0,51	0,45	0,42
Κέρκυρα	0,43	0,46	0,49	0,52	0,57	0,61	0,63	0,63	0,59	0,53	0,47	0,43
Κοζάνη												
Κομοτηνή	0,4	0,42	0,45	0,51	0,55	0,6	0,61	0,62	0,59	0,53	0,45	0,41
Κόνιτσα	0,41	0,41	0,47	0,48	0,56	0,57	0,62	0,62	0,56	0,52	0,47	0,43
Κόρινθος	0,46	0,49	0,51	0,54	0,59	0,63	0,64	0,64	0,6	0,56	0,5	0,43
Κύθηρα	0,45	0,46	0,51	0,54	0,61	0,63	0,64	0,64	0,61	0,56	0,51	0,44
Κως												
Λαμία	0,43	0,44	0,47	0,52	0,55	0,6	0,61	0,61	0,57	0,51	0,47	0,43
Λάρισα	0,41	0,44	0,47	0,52	0,56	0,61	0,62	0,62	0,58	0,5	0,45	0,4
Λευκάδα												
Λήμνος	0,39	0,43	0,47	0,53	0,58	0,62	0,63	0,63	0,6	0,54	0,45	0,39
Μεθώνη	0,47	0,49	0,53	0,54	0,61	0,62	0,63	0,63	0,63	0,6	0,57	0,49
Μήλος	0,38	0,39	0,48	0,59	0,62	0,64	0,64	0,64	0,63	0,54	0,51	0,39
Μυτιλήνη	0,38	0,43	0,47	0,54	0,61	0,63	0,64	0,64	0,62	0,56	0,48	0,41
Νάξος	0,41	0,45	0,5	0,55	0,6	0,63	0,64	0,64	0,61	0,56	0,49	0,42
Ξάνθη												
Πάρος	0,41	0,46	0,51	0,57	0,61	0,63	0,64	0,63	0,62	0,57	0,5	0,44
Πάτρα	0,39	0,43	0,51	0,5	0,58	0,62	0,62	0,62	0,6	0,53	0,46	0,42
Πολύγυρος												
Πύργος	0,48	0,49	0,52	0,54	0,58	0,62	0,64	0,64	0,6	0,57	0,52	0,46
Ρέθυμνο	0,4	0,45	0,47	0,55	0,61	0,63	0,64	0,64	0,61	0,5	0,51	0,41
Ρόδος	0,47	0,48	0,52	0,55	0,59	0,63	0,64	0,64	0,61	0,58	0,52	0,45
Σάμος	0,45	0,48	0,52	0,55	0,6	0,65	0,66	0,66	0,64	0,6	0,54	0,45
Σέρρες	0,41	0,44	0,46	0,49	0,53	0,58	0,6	0,6	0,57	0,51	0,44	0,39
Σητεία	0,43	0,46	0,5	0,55	0,6	0,64	0,65	0,65	0,62	0,56	0,51	0,44
Σκύρος	0,37	0,42	0,46	0,53	0,58	0,62	0,63	0,63	0,6	0,52	0,45	0,39
Σούδα	0,42	0,46	0,52	0,56	0,61	0,64	0,65	0,65	0,62	0,55	0,49	0,43
Σπάρτη												
Σύρος	0,4	0,47	0,49	0,59	0,62	0,63	0,64	0,64	0,62	0,98	0,5	0,44
Τανάγρα	0,42	0,44	0,46	0,52	0,57	0,62	0,63	0,63	0,6	0,52	0,45	0,41
Τρίκαλα(Ημαθίας)	0,45	0,46	0,45	0,49	0,52	0,58	0,59	0,59	0,56	0,5	0,45	0,43
Τρίκαλα(Θεσσαλίας)												
Τρίτολη	0,46	0,46	0,49	0,52	0,55	0,6	0,61	0,61	0,6	0,54	0,49	0,45
Τυμπάκιο	0,47	0,5	0,54	0,57	0,6	0,64	0,65	0,65	0,63	0,59	0,54	0,47
Φλώρινα												
Χαλκίδα												
Χανιά	0,4	0,45	0,49	0,56	0,62	0,63	0,64	0,64	0,61	0,52	0,5	0,42
Χίος	0,39	0,43	0,49	0,55	0,61	0,63	0,64	0,64	0,62	0,58	0,5	0,42
Καβάλα	0,46	0,5	0,48	0,48	0,56	0,59	0,6	0,6	0,57	0,52	0,48	0,39

Πίνακας Γ.5 Μέση θερμοκρασία δικτύου παροχής νερού για διάφορες τοποθεσίες της Ελλάδας.

ΜΗΝΑΣ	Μέση θερμοκρασία δικτύου νερού (οC) σύμφωνα με ΕΛΟΤ 1291[24].											
	ΙΑΝ	ΦΕΒ	ΜΑΡ	ΑΠΡ	ΜΑΙ	ΙΟΥΝ	ΙΟΥΛ	ΑΥΓ	ΣΕΠ	ΟΚΤ	ΝΟΕ	ΔΕΚ
Αθήνα(Ελληνικό)	11,3	10,9	11,8	14,3	17,7	21,6	24,7	25,7	24,2	21,1	16,9	13,5
Αθήνα(Φιλαδέλφεια)	0	0	0	0	0	0	0	0	0	0	0	0
Αργίριο	0	0	0	0	0	0	0	0	0	0	0	0
Αγχιάλος	0	0	0	0	0	0	0	0	0	0	0	0
Αλεξανδρούπολη	0	0	0	0	0	0	0	0	0	0	0	0
Αλιάρτος	10,2	9,9	10,8	13,9	17,6	21,7	25,1	25,8	24,2	21	16,5	13
Ανδραβίδα	0	0	0	0	0	0	0	0	0	0	0	0
Άραξος	12,8	12,3	12,9	14,8	17,6	21,5	24,4	25,6	25	22,3	18,2	15
Άργος	0	0	0	0	0	0	0	0	0	0	0	0
Αργοστόλι	0	0	0	0	0	0	0	0	0	0	0	0
Άρτα	0	0	0	0	0	0	0	0	0	0	0	0
Δράμα	0	0	0	0	0	0	0	0	0	0	0	0
Έδεσσα	0	0	0	0	0	0	0	0	0	0	0	0
Ζάκυνθος	0	0	0	0	0	0	0	0	0	0	0	0
Ηράκλειο	14,7	14,2	14,8	17,2	20,6	24,5	27,3	28,2	27,2	24,7	20,9	17,2
Θεσσαλονίκη	8,2	7,9	9,2	12,8	16,8	20,2	21,5	22,8	22,1	19,4	15,7	11
Ιεράπετρα	15,4	14,7	15,3	17,5	21,2	25,1	27,7	28,7	27,4	24,8	21,2	17,7
Ιωάννινα	6,2	6,4	8	10,9	14,1	18	21,5	23,2	21,6	17,9	13,1	8,9
Καλαμάτα	12,5	12,2	12,9	15,1	18,3	21,8	24,7	25,7	24,9	21,9	18	14,4
Καρδίτσα	0	0	0	0	0	0	0	0	0	0	0	0
Καρπενήσι	0	0	0	0	0	0	0	0	0	0	0	0
Κάρυστος	0	0	0	0	0	0	0	0	0	0	0	0
Καστοριά	0	0	0	0	0	0	0	0	0	0	0	0
Κέρκυρα	11,3	11,1	12	14,2	17,7	20,5	22,6	23,6	22,7	20,3	16,9	13,3
Κοζάνη	0	0	0	0	0	0	0	0	0	0	0	0
Κομοτηνή	0	0	0	0	0	0	0	0	0	0	0	0
Κόνιτσα	0	0	0	0	0	0	0	0	0	0	0	0
Κόρινθος	13,1	12,7	13,4	15,8	19,1	23	26	27	26	22,9	18,9	15,5
Κύθηρα	0	0	0	0	0	0	0	0	0	0	0	0
Κως	0	0	0	0	0	0	0	0	0	0	0	0
Λαμία	11,3	11	11,5	14	16,5	19,3	21,8	22,8	22	20	16,5	13,4
Λάρισα	10	9,4	10,8	14,1	16,7	20,2	22,7	24,7	23,8	21,2	16,7	14
Λευκάδα	0	0	0	0	0	0	0	0	0	0	0	0
Λήμνος	10,3	9,2	10	13,6	17,7	22,7	25,9	26,2	24,8	21,4	16,6	13
Μεθώνη	0	0	0	0	0	0	0	0	0	0	0	0
Μήλος	0	0	0	0	0	0	0	0	0	0	0	0
Μυτιλήνη	13,2	12,7	13,1	15	18,3	22	24,6	25,4	24,4	21,9	18,4	15,3
Νάξος	12,5	12,2	12,9	15,1	18,3	21,8	24,7	25,7	24,9	21,9	18	14,4
Ξάνθη	0	0	0	0	0	0	0	0	0	0	0	0
Πάρος	0	0	0	0	0	0	0	0	0	0	0	0
Πάτρα	0	0	0	0	0	0	0	0	0	0	0	0
Πολύγυρος	0	0	0	0	0	0	0	0	0	0	0	0
Πύργος	0	0	0	0	0	0	0	0	0	0	0	0
Ρέθυμνο	0	0	0	0	0	0	0	0	0	0	0	0
Ρόδος	13,5	12,9	13,4	16	19,3	22,7	25,4	26	25,1	22,4	18,4	15,2
Σάμος	0	0	0	0	0	0	0	0	0	0	0	0
Σέρρες	8,3	8,4	9,7	12,8	16,6	20,4	22,8	23,8	22,6	19,7	15,3	11
Σητεία	0	0	0	0	0	0	0	0	0	0	0	0
Σκύρος	0	0	0	0	0	0	0	0	0	0	0	0
Σούδα	0	0	0	0	0	0	0	0	0	0	0	0
Σπάρτη	0	0	0	0	0	0	0	0	0	0	0	0
Σύρος	0	0	0	0	0	0	0	0	0	0	0	0
Τανάγρα	0	0	0	0	0	0	0	0	0	0	0	0
Τρίκαλα(Ημαθίας)	0	0	0	0	0	0	0	0	0	0	0	0
Τρίκαλα(Θεσσαλίας)	0	0	0	0	0	0	0	0	0	0	0	0
Τρίτολη	0	0	0	0	0	0	0	0	0	0	0	0
Τυμπάκιο	14,7	14,2	14,8	17,2	20,6	24,5	27,3	28,2	27,2	24,7	20,9	17,2
Φλώρινα	0	0	0	0	0	0	0	0	0	0	0	0
Χαλκίδα	0	0	0	0	0	0	0	0	0	0	0	0
Χανιά	14,2	13,4	13,7	15,4	17,8	20,9	23	24,1	23,6	21,7	19	16,1
Χίος	14,1	12,6	13,4	15,2	17,4	20,1	22,7	23,8	23,3	21,4	18,1	15,2
Καβάλα	0	0	0	0	0	0	0	0	0	0	0	0

Πίνακας Γ.6 Μέγιστη μηνιαία θερμοκρασία για διάφορες τοποθεσίες της Ελλάδας.

ΜΗΝΑΣ	Μέγιστη μηνιαία θερμοκρασία (οC) σύμφωνα με ΤΕΕ											
	ΙΑΝ	ΦΕΒ	ΜΑΡ	ΑΠΡ	ΜΑΙ	ΙΟΥΝ	ΙΟΥΛ	ΑΥΓ	ΣΕΠ	ΟΚΤ	ΝΟΕ	ΔΕΚ
Αθήνα(Ελληνικό)												
Αθήνα(Φιλαδέλφεια)	12,5	13,6	15,7	20,4	26,1	31,2	33,6	33,3	29,2	23,5	18,2	14,2
Αργίριο	13,6	14,5	16,9	20,6	26	30,5	33,4	33,7	29,8	24,5	18,9	14,8
Αγχάλως	11,1	12,3	14,4	18,9	24	29	31,1	30,8	27,1	21,8	16,8	12,5
Αλεξανδρούπολη	8,6	9,8	12,2	17,2	22,4	27,2	30,3	30,5	26,3	20,4	14,8	10,6
Αλιάρτος												
Ανδραβίδα	14	14,4	16,4	19,3	24	28,1	30,4	31,1	28,1	23,8	19,1	15,3
Άραξος												
Άργος	14,5	14,8	17	21,2	26,2	31,1	33,8	33,5	29,9	24,6	19	15,5
Αργοστόλι	14,3	14,3	15,8	18,3	22,6	26,5	28,8	29,5	26,8	23,1	18,8	15,5
Άρτα	13,8	14,5	16,9	19,9	25	28,8	31,4	31,8	28,5	24,3	18,4	14,5
Δράμα	8,2	10,5	14,5	19,6	25,2	29,9	31,9	31,1	28	21,3	13,6	9
Έδεσσα	9,4	10,5	13,6	17,5	23	28,3	29,8	29,4	25,9	20	13,9	10
Ζάκυνθος	14,4	14,3	15,8	18,8	24	28,6	31,9	31,9	28,5	23,6	18,8	15,4
Ηράκλειο												
Θεσσαλονίκη												
Ιεράπετρα												
Ιωάννινα												
Καλαμάτα												
Καρδίτσα	8,8	11,7	16,1	19,7	24,3	30,8	32,6	31,8	28,4	21,9	14,5	8,4
Καρπενήσι	7,8	6,9	9,3	15,5	19,5	23,8	26,4	26,2	23,2	17,5	11,3	8,7
Κάρυστος	13,7	13,5	15,9	19,4	23,3	28	30,4	30,1	27,6	23,2	18,3	14,8
Καστοριά												
Κέρκυρα												
Κοζάνη	6,1	8	11,4	16,3	21,7	26,5	29,3	29,2	25	18,8	12,7	8
Κομοτηνή	8,8	10,4	12,8	17,5	22,9	27,7	30,2	30,4	26,4	21	16	11,2
Κόνιτσα	10	11,4	14,6	17,4	22,7	27,6	30,8	30,7	27,1	21,1	14,6	10,5
Κόρινθος												
Κύθηρα	12,8	13	14,1	17	21,6	26,2	28,7	28,7	25,5	21,7	17,8	14,6
Κως	13,6	13,2	15	18,8	23,3	28	30,4	30	27,4	23,1	17,7	14,9
Λαμία												
Λάρισα												
Λευκάδα	13,4	13,8	15,8	18,3	22,7	26,5	29	29	26,4	22,8	17,8	14,6
Λήμνος												
Μεθώνη	14,7	14,8	16,1	18,4	22	25,5	27,7	28,8	26,9	23,6	19,6	16,3
Μήλος	13,1	13,3	14,7	18,3	23	27,3	28,4	28,1	25,5	21,8	18,1	14,8
Μυτιλήνη												
Νάξος												
Ξάνθη	9,4	10,4	13,4	18,5	23,8	28,4	31,1	30,7	27,2	21,1	14,5	10,4
Πάρος	14,9	15	17,1	20,8	24,2	28,4	29,6	29,5	27,7	23,9	19,2	16
Πάτρα	14,5	15	16,8	19,7	24,2	28	30,1	30,9	28,2	24	19,5	16,1
Πολύγυρος	10,9	8,7	13	16,5	20,6	26	27,5	27,1	24,8	19,7	15,2	12,8
Πύργος	14,7	15,1	17,2	20	24,7	29	31,9	32,1	29	24,8	19,4	15,8
Ρέθυμνο	15,5	15,7	17,2	20,4	24,2	28,2	29,9	29,8	27,5	24	20,4	17,2
Ρόδος												
Σάμος	13,4	13,4	15,7	19,7	24,7	29,8	32,7	32,5	28,6	23,4	18	14,9
Σέρρες												
Σητεία	15,2	15,4	16,9	20	23,6	27,3	28,6	28,5	26,6	23,6	20,2	16,9
Σκύρος	12,3	12,7	14	17,8	21,9	26,2	27,8	27,4	24,7	20,8	17,1	13,9
Σούδα	14,3	14,7	16,5	20	24,5	28,9	30,5	30,2	27,4	23,4	19,6	16,2
Σπάρτη	14,2	14,8	17,4	21,1	26,7	32	34,5	34	30,5	25,3	19	15,2
Σύρος	14,1	14,4	15,8	19,4	23,6	28,1	29,5	29,3	27	22,9	18,5	15,1
Τανάγρα	11,7	12,7	14,9	19,4	25	30,1	32	31,6	27,9	22,4	17,5	13,4
Τρίκαλα(Ημαθίας)	9,1	11,1	14,9	19,8	25,1	29,7	31,4	31,2	27,7	21,8	14,5	9,7
Τρίκαλα(Θεσσαλίας)	10	11,6	15,4	20	25,8	31,1	32,8	31,9	28,8	22,1	15,1	10,9
Τρίτολη	9,6	10,5	13	17,1	22,7	27,8	30,1	30,1	26,4	20,6	15,5	11,2
Τυμπάκιο												
Φλώρινα	4,6	7,3	11,8	16,7	22	26,2	28,8	28,7	25,2	19	12,1	6,3
Χαλκίδα	12,7	12,4	14,8	18,6	22,6	27,4	30,7	30,7	27	21	16	13,9
Χανιά												
Χίος												
Καβάλα	9,6	10,3	12,5	17,5	22,5	27,2	30	30,1	25,9	20,4	14,5	10,4

Πίνακας Γ.7 Μέγιστη μηνιαία διακύμανση θερμοκρασίας για διάφορες τοποθεσίες της Ελλάδας.

ΜΗΝΑΣ	Μέγιστη μηνιαία διακύμανση θερμοκρασίας (οC) σύμφωνα με ΤΕΕ											
	ΙΑΝ	ΦΕΒ	ΜΑΡ	ΑΠΡ	ΜΑΙ	ΙΟΥΝ	ΙΟΥΛ	ΑΥΓ	ΣΕΠ	ΟΚΤ	ΝΟΕ	ΔΕΚ
Αθήνα(Ελληνικό)												
Αθήνα(Φιλαδέλφεια)	4,4	6	6,8	8,8	10,8	11,6	11	10,6	9,8	7,8	5,6	4,4
Αργίριο	10,6	10,6	10,8	10,8	11,2	11,4	12,4	13,4	13,6	13	11,4	10,4
Αγχίαλος	9	9,2	8,8	9,2	8,8	8,8	8,4	9	9,8	9,6	9,4	8,8
Αλεξανδρούπολη	7,2	7,8	7,8	8	8	8	8,8	9,8	10,4	9,4	8	7,2
Αλιάρτος												
Ανδραβίδα	9,2	9	9,2	9	9	8,8	9	10,2	10,4	11	10	8,8
Άραξος												
Άργος	12,8	12,8	12,8	13	12,4	12,4	13	14	14,6	13,6	12,2	12
Αργοστόλι	5,8	5,6	6	6	6,2	6	6,2	6,8	6,8	6,6	6	5,4
Άρτα	11,6	11	10,8	10,6	10,8	10,4	11,4	12,4	13	13	11,8	10,8
Δράμα	7	8	9	9,8	10	10,4	10,4	10,4	11,4	9,8	7,4	6
Έδεσσα	9,8	8,6	8	8,2	8,8	9	9,2	9,8	10,4	8,8	9	8,4
Ζάκυνθος	7,8	7,6	8	8	8,6	8,6	9,4	9,6	9,8	9	8	7,4
Ηράκλειο												
Θεσσαλονίκη												
Ιεράπετρα												
Ιωάννινα												
Καλαμάτα												
Καρδίτσα	8,6	9,6	11,4	11,6	12,6	13,2	12,6	12,4	12,6	11,6	8,8	8,2
Καρπενήσι	8	7,6	7,8	9,8	9,6	9,8	9,6	10,6	11,2	10,2	9	7,8
Κάρυστος	6,6	6,4	7	7,4	7,6	7,6	7,2	7,4	8	7,6	6,8	6,2
Καστοριά												
Κέρκυρα												
Κοζάνη	7,6	8,6	9	9,4	9,8	10	10,4	11,2	11,4	10,6	9,4	8,2
Κομοτηνή	8	8,4	8,4	8,8	9	9,4	9,4	10,8	11,6	11,6	10,4	8,4
Κόνιτσα	9,6	9,8	10,2	10,4	11	11,8	12,8	13,4	13,6	11,4	9,6	8,2
Κόρινθος	17,6	18,6	23	30,8	41,4	51,6	56,6	55,6	46,8	37,2	26,8	20,2
Κύθηρα	3,8	4,2	4,4	4,8	5,4	6	6	6	5,2	5,2	4	3,8
Κως	5,2	5,4	5,8	6,8	7,6	8,4	9	9,2	8,4	7,4	5,4	5
Λαμία												
Λάρισα												
Λευκάδα	6,4	6,4	6,2	6,2	6,6	6,8	7,2	7	6,8	7	6,4	6,2
Λήμνος												
Μεθώνη	6,8	6,6	6,4	6	6	5,8	5,8	6,2	6,6	7,6	7,2	6,8
Μήλος	4,8	5	5,6	6,6	7,2	7,4	6,4	6,4	6,4	6	5,6	4,8
Μυτιλήνη												
Νάξος												
Ξάνθη	7,6	7,2	7,6	8,4	8	8,6	9	9,4	9,6	9,2	7	7
Πάρος	7,4	7,6	8,4	9,2	8,8	8,8	8,2	9	9,8	9,6	8	7,4
Πάτρα	9	8,8	8,6	8,2	8,2	7,8	7,4	8,4	9,4	10	10	9,4
Πολύγυρος	12	8	8,6	8,2	8,6	7,2	7	6	7	9,2	9	12
Πύργος	10,2	10	10	9,6	10	10,2	11	11,6	12	12,2	10,6	9,6
Ρέθυμνο	5,4	5,6	6	6,6	7	6,6	6	6	6,6	6,8	6,2	5,4
Ρόδος												
Σάμος	6	6,4	7	7,2	7,8	8,2	8,2	8,6	8,4	7,6	6,6	5,8
Σέρρες												
Σητεία	6	6,2	6,6	6,8	6,6	6,2	5,4	5,6	6	6,8	6,8	6,2
Σκύρος	4,8	5	5	5,6	5,2	5	4,6	4,6	5	5	5	4,6
Σούδα	6,8	7,4	7,8	8,4	8,8	8,6	7,8	551,8	8,6	8,2	8	7,2
Σπάρτη	9,4	9,4	10,2	11	11,6	12,2	12,6	13,2	12,6	12	10	9
Σύρος	4,6	5,2	5,4	6	6,6	6,2	5,6	5,6	5,8	5,6	5,2	4,6
Τανάγρα	8,4	9	9,4	9,8	10	9,6	9	9,8	11	10,4	9,4	8
Τρίκαλα(Ημαθίας)	8,8	10	10,6	10,8	10,8	10,8	11	12,4	13,2	12	9,6	8,4
Τρίκαλα(Θεσσαλίας)	9,2	9,4	9,6	10	10,4	11	10,8	11,6	12,4	11,4	9,8	9
Τρίτολη	9	9,4	10,2	10,8	11,4	11,6	11,2	12	12,8	12	10,8	9
Τυμπάκιο												
Φλώρινα	8,2	9,2	10,2	10,2	10,4	10,4	11,4	12,4	13,6	12,8	10,2	8,2
Χαλκίδα	7,2	6,6	6	5	3,8	3,2	5,8	6,4	5	2,6	4,2	6,8
Χανιά												
Χίος												
Καβάλα	8,2	8,2	7,6	7,6	7,8	7,6	8,2	9,6	9,8	9,6	8,2	7,6

Πίνακας Γ.8 Βαθμομέρες θέρμανσης με θερμοκρασία αναφοράς 18 βαθμούς κελσίου για διάφορες τοποθεσίες της Ελλάδας.

ΜΗΝΑΣ	Βαθμομέρες θέρμανσης DD με θερμοκρασία αναφοράς 18οC											
	ΙΑΝ	ΦΕΒ	ΜΑΡ	ΑΠΡ	ΜΑΙ	ΙΟΥΝ	ΙΟΥΛ	ΑΥΓ	ΣΕΠ	ΟΚΤ	ΝΟΕ	ΔΕΚ
Αθήνα (Ελληνικό)	239	207	177	60	0	0	0	0	0	0	78	186
Αθήνα (Φιλαδέλφεια)	239	207	177	60	0	0	0	0	0	0	78	186
Αργίριο	301	246	202	84	0	0	0	0	0	0	144	260
Αρχιάλο	353	288	248	111	0	0	0	0	0	31	177	307
Αλεξανδρούπολη	403	339	301	144	0	0	0	0	0	71	216	341
Αλιάρτο	338	274	229	84	0	0	0	0	0	34	180	291
Ανδραβίδα	267	227	192	96	0	0	0	0	0	0	117	220
Άραξο	242	210	180	84	0	0	0	0	0	0	99	198
Άργο	307	269	229	99	0	0	0	0	0	6	153	264
Αργοστόλι	205	182	161	81	0	0	0	0	0	0	66	161
Άρτα	310	252	202	102	0	0	0	0	0	6	165	276
Δράμα	412	322	248	99	0	0	0	0	0	50	243	372
Έδεσσα	419	330	260	138	0	0	0	0	0	74	258	378
Ζάκυνθο	233	210	192	96	0	0	0	0	0	0	96	195
Ηράκλειο	183	162	140	45	0	0	0	0	0	0	39	133
Θεσσαλονίκη	394	314	254	111	0	0	0	0	0	53	207	344
Ιεράπετρα	158	143	118	30	0	0	0	0	0	0	15	109
Ιωάννινα	412	336	285	168	16	0	0	0	0	93	249	378
Καλαμάτα	242	207	177	84	0	0	0	0	0	0	96	198
Καρδίτσα	419	311	236	123	0	0	0	0	0	59	237	425
Καρπενήσι	440	417	391	222	102	0	0	0	0	174	336	409
Κάρυστο	236	216	174	69	0	0	0	0	0	93	195	
Καστοριά	490	409	344	195	50	0	0	0	0	143	324	465
Κέρκυρα	257	216	186	90	0	0	0	0	0	111	214	
Κοζάνη	487	400	344	192	37	0	0	0	0	140	300	437
Κομοτηνή	409	330	291	147	0	0	0	0	0	87	216	341
Κόνιτσα	397	322	264	174	25	0	0	0	0	81	246	360
Κόρινθο	285	244	202	78	0	0	0	0	0	0	138	245
Κύθηρα	220	199	189	102	0	0	0	0	0	0	66	164
Κω	217	210	183	78	0	0	0	0	0	0	90	174
Λαμία	338	277	226	90	0	0	0	0	0	31	183	304
Λάρισα	397	314	264	120	0	0	0	0	0	53	213	357
Λευκάδα	242	207	164	84	0	0	0	0	0	102	202	
Λήμνο	329	286	257	126	0	0	0	0	0	37	171	279
Μεθώνη	208	182	158	78	0	0	0	0	0	0	60	158
Μήλο	226	202	189	90	0	0	0	0	0	0	81	174
Μυτιλήνη	264	227	198	72	0	0	0	0	0	0	111	208
Νάξο	183	162	146	57	0	0	0	0	0	0	51	136
Ξάνθη	384	314	260	111	0	0	0	0	0	47	210	344
Πάρο	211	190	158	54	0	0	0	0	0	0	84	177
Πάτρα	248	207	171	72	0	0	0	0	0	0	105	205
Πολύγυρο	406	372	288	168	53	0	0	0	0	90	219	347
Πύργο	260	221	180	84	0	0	0	0	0	0	117	217
Ρέθυμνο	161	143	118	27	0	0	0	0	0	0	21	109
Ρόδο	186	162	133	42	0	0	0	0	0	0	39	140
Σάμο	236	218	180	57	0	0	0	0	0	0	99	186
Σέρρες	434	328	257	108	0	0	0	0	0	71	258	409
Σητεία	180	160	136	42	0	0	0	0	0	0	36	130
Σκύρο	251	218	202	90	0	0	0	0	0	0	102	198
Σούδα	220	196	167	66	0	0	0	0	0	0	72	167
Σπάρτη	264	221	177	72	0	0	0	0	0	0	120	226
Σύρο	192	174	152	48	0	0	0	0	0	0	63	161
Τανάγρα	326	274	242	105	0	0	0	0	0	25	156	267
Τρίκαλα(Ημαθίας)	412	333	260	108	0	0	0	0	0	68	249	388
Τρίκαλα(Θεσσαλίας)	391	311	229	90	0	0	0	0	0	50	234	360
Τρίπολη	400	342	313	189	31	0	0	0	0	105	237	350
Τυμπάκιο	195	174	143	48	0	0	0	0	0	0	48	149
Φλώρινα	543	428	350	192	37	0	0	0	0	167	330	490
Χαλκίδα	276	249	192	57	0	0	0	0	0	0	123	233
Χανιά	198	174	149	51	0	0	0	0	0	0	57	152
Χίο	260	232	198	87	0	0	0	0	0	0	132	214
Καβάλα	388	330	288	129	0	0	0	0	0	74	228	353