

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΘΕΡΜΟΤΗΤΑΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Μείωση της θερμοκρασίας κυψέλης φωτοβολταϊκής μονάδας με τη χρήση υλικών αλλαγής φάσης. Προσομοίωση με τη μέθοδο πεπερασμένων διαφορών. »

«Using phase change materials in photovoltaic systems for cell temperature reduction: A finite difference simulation method. »

ΣΠΟΥΔΑΣΤΗΣ: Κλαδίσκος Παναγιώτης

ΕΠΙΒΛΕΠΟΥΣΑ: Δρ. Στέγγου-Σαγιά Αθηνά, Καθηγήτρια Ε.Μ.Π.

ΑΘΗΝΑ, 2015

ΕΥΧΑΡΙΣΤΙΕΣ

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας θα ήθελα να ευχαριστήσω την οικογένεια μου για την άνευ όρων στήριξη και συμπαράσταση που μου έδειξε όλα αυτά τα χρόνια. Πάνω απ' όλα, θα ήθελα να ευχαριστήσω θερμά την επιβλέπουσα καθηγήτρια, κα Στέγγου-Σαγιά Αθηνά, καθηγήτρια Ε.Μ.Π. Παρά τον περιορισμένο χρόνο της, πάντα εύρισκε χρόνο να με καθοδηγεί με ουσιαστικές συμβουλές και υποδείξεις.

ΣΥΝΟΨΗ

Οι κατασκευάστριες εταιρείες φωτοβολταϊκών μονάδων αξιολογούν τις μονάδες με βάση τη μέγιστη παραγόμενη ισχύ τους. Η ονομαστική ισχύς αντιστοιχεί σε κοινές συνθήκες αναφοράς (θερμοκρασία κυψέλης $T_{cell}=25\text{ }^{\circ}\text{C}$, ένταση ηλιακής ακτινοβολίας $G=1\text{ kW/m}^2$, και μάζα αέρα $AM=1.5$). Στην πραγματικότητα η ένταση της ηλιακής ακτινοβολίας είναι χαμηλότερη και η θερμοκρασία κυψέλης υψηλότερη, παράγοντες που οδηγούν σε μειωμένη παραγόμενη ισχύ. Στην παρούσα εργασία θα εξεταστεί η πιθανότητα μείωσης της θερμοκρασίας κυψέλης με μία από τις προτεινόμενες μεθόδους θερμικού ελέγχου, τη χρήση υλικών αλλαγής φάσης.

Η φωτοβολταϊκή μονάδα, τόσο αυτούσια όσο και σαν σύστημα σε άμεση επαφή με το εκάστοτε υλικό αλλαγής φάσης, θα μοντελοποιηθεί με τη μέθοδο πεπερασμένων διαφορών. Να σημειωθεί ότι, η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η Python (2.7.3), μία σχετικά νέα γλώσσα με πληθώρα εργαλείων και, πάνω από όλα, απλή σύνταξη.

ABSTRACT

Photovoltaic manufacturers rate their modules based on their maximum power output. This nominal power corresponds to the commonly accepted standard test conditions (STC, cell temperature $T_{cell}=25\text{ }^{\circ}\text{C}$, insolation $G=1\text{ kW/m}^2$, air mass $AM=1.5$). In reality, insolation is lower and cell temperature higher, both being factors that affect power generation in a negative fashion. In the present paper, the possibility of cell temperature reduction will be investigated using one of the proposed methods of thermal control, phase change materials.

The photovoltaic module, both as a unit and as a system in direct contact with a phase change material, will be simulated using a finite difference method. It should be noted that, Python (2.7.3) was used as a programming language, a relatively new language with a plethora of tools and, above all, simple syntax.

ΠΕΡΙΕΧΟΜΕΝΑ

<u>ΠΙΝΑΚΑΣ ΣΥΜΒΟΛΩΝ.....</u>	<u>6</u>
<u>A-ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ.....</u>	<u>8</u>
<u>A.1-ΘΕΩΡΙΑ ΦΩΤΟΒΟΛΤΑΪΚΩΝ.....</u>	<u>8</u>
<u> A.1.1-ΓΕΝΙΚΑ.....</u>	<u>8</u>
<u> A.1.2-ΛΕΙΤΟΥΡΓΙΑ ΦΩΤΟΒΟΛΤΑΪΚΩΝ.....</u>	<u>9</u>
<u> A.1.3-ΤΑΞΙΝΟΜΗΣΗ ΦΩΤΟΒΟΛΤΑΪΚΩΝ ΚΑΙ ΓΕΝΝΗΤΡΙΩΝ.....</u>	<u>11</u>
<u> A.1.4-ΕΠΙΔΡΑΣΗ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ ΚΑΙ ΚΛΙΣΗΣ ΣΤΗΝ ΠΡΟΣΠΙΠΤΟΥΣΑ</u> <u> ΑΚΤΙΝΟΒΟΛΙΑ.....</u>	<u>13</u>
<u> A.1.5-ΜΕΡΗ ΦΩΤΟΒΟΛΤΑΪΚΟΥ ΣΥΣΤΗΜΑΤΟΣ.....</u>	<u>16</u>
<u> A.1.6-ΤΥΠΟΙ ΦΩΤΟΒΟΛΤΑΪΚΩΝ ΣΥΣΤΗΜΑΤΩΝ.....</u>	<u>22</u>
<u> A.1.7-ΤΑΞΙΝΟΜΗΣΗ ΦΩΤΟΒΟΛΤΑΪΚΩΝ ΣΥΣΤΗΜΑΤΩΝ.....</u>	<u>26</u>
<u>A.2-ΘΕΩΡΙΑ ΥΛΙΚΩΝ ΑΛΛΑΓΗΣ ΦΑΣΗΣ.....</u>	<u>28</u>
<u> A.2.1-ΓΕΝΙΚΑ.....</u>	<u>27</u>
<u> A.2.2-ΚΑΤΗΓΟΡΙΕΣ ΥΛΙΚΩΝ ΑΛΛΑΓΗΣ ΦΑΣΗΣ.....</u>	<u>30</u>
<u> A.2.3-ΚΡΙΤΗΡΙΑ ΕΠΙΛΟΓΗΣ ΥΛΙΚΩΝ ΑΛΛΑΓΗΣ ΦΑΣΗΣ.....</u>	<u>34</u>
<u> A.2.4-ΕΦΑΡΜΟΓΕΣ ΥΛΙΚΩΝ ΑΛΛΑΓΗΣ ΦΑΣΗΣ.....</u>	<u>36</u>
<u>A.3-ΕΞΙΣΩΣΗ ΘΕΡΜΟΤΗΤΑΣ.....</u>	<u>37</u>
<u>A.4-ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ.....</u>	<u>38</u>
<u>A.5-ΥΠΟΛΟΓΙΣΜΟΣ ΣΤΙΓΜΙΑΙΑΣ ΟΛΙΚΗΣ ΗΛΙΑΚΗΣ ΑΚΤΙΝΟΒΟΛΙΑΣ</u> <u>ΚΕΚΛΙΜΕΝΟΥ ΕΠΙΠΕΔΟΥ.....</u>	<u>50</u>
<u>A.6-ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΥΛΙΚΩΝ ΑΛΛΑΓΗΣ ΦΑΣΗΣ.....</u>	<u>54</u>
<u>A.7-ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-ΥΛΙΚΟΥ</u> <u>ΑΛΛΑΓΗΣ ΦΑΣΗΣ.....</u>	<u>58</u>
<u>B-ΥΠΟΛΟΓΙΣΤΙΚΟ ΜΕΡΟΣ.....</u>	<u>66</u>
<u> B.1-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΑΕΡΑ.....</u>	<u>66</u>
<u> B.2-ΥΠΟΛΟΓΙΣΜΟΣ ΣΤΙΓΜΙΑΙΑΣ ΟΛΙΚΗΣ ΗΛΙΑΚΗΣ ΑΚΤΙΝΟΒΟΛΙΑΣ</u> <u> ΚΕΚΛΙΜΕΝΟΥ ΕΠΙΠΕΔΟΥ.....</u>	<u>67</u>
<u> B.3-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ.....</u>	<u>72</u>
<u> B.4-ΥΠΟΛΟΓΙΣΜΟΣ ΣΥΝΤΕΛΕΣΤΗ ΜΕΤΑΦΟΡΑΣ ΘΕΡΜΟΤΗΤΑΣ.....</u>	<u>81</u>
<u> B.5-ΣΥΓΚΡΙΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΕΝΕΡΓΟΥ ΘΕΡΜΟΧΩΡΗΤΙΚΟΤΗΤΑΣ ΓΙΑ ΤΑ ΥΛΙΚΑ</u> <u> ΑΛΛΑΓΗΣ ΦΑΣΗΣ RT20, RT27, SP25A8.....</u>	<u>82</u>
<u> B.6-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ</u> <u> ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-ΥΛΙΚΟΥ ΑΛΛΑΓΗΣ ΦΑΣΗΣ.....</u>	<u>83</u>
<u> B.6.1-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ</u> <u> ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-RT20.....</u>	<u>83</u>
<u> B.6.2-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ</u> <u> ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-RT27.....</u>	<u>92</u>
<u> B.6.3-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ</u> <u> ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-SP25A8.....</u>	<u>101</u>
<u> B.7-ΣΥΓΚΡΙΤΙΚΑ ΔΙΑΓΡΑΜΜΑΤΑ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ.....</u>	<u>110</u>
<u> B.8-ΣΥΜΠΕΡΑΣΜΑΤΑ.....</u>	<u>127</u>
<u>Γ-ΚΩΔΙΚΑΣ ΡΥΘΜΩΝ.....</u>	<u>129</u>
<u> Γ.1-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΑΕΡΑ.....</u>	<u>129</u>
<u> Γ.2-ΥΠΟΛΟΓΙΣΜΟΣ ΣΤΙΓΜΙΑΙΑΣ ΟΛΙΚΗΣ ΗΛΙΑΚΗΣ ΑΚΤΙΝΟΒΟΛΙΑΣ</u> <u> ΚΕΚΛΙΜΕΝΟΥ ΕΠΙΠΕΔΟΥ.....</u>	<u>132</u>

<u>Γ.3-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ.....</u>	<u>138</u>
<u>Γ.4-ΥΠΟΛΟΓΙΣΜΟΣ ΣΥΝΤΕΛΕΣΤΗ ΜΕΤΑΦΟΡΑΣ ΘΕΡΜΟΤΗΤΑΣ.....</u>	<u>144</u>
<u>Γ.5-ΣΥΓΚΡΙΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΕΝΕΡΓΟΥ ΘΕΡΜΟΧΩΡΗΤΙΚΟΤΗΤΑΣ ΓΙΑ ΤΑ ΥΛΙΚΑ ΑΛΛΑΓΗΣ ΦΑΣΗΣ RT20, RT27, SP25A8.....</u>	<u>147</u>
<u>Γ.6-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-ΥΛΙΚΟΥ ΑΛΛΑΓΗΣ ΦΑΣΗΣ.....</u>	<u>149</u>
<u>Γ.6.1-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-RT20.....</u>	<u>149</u>
<u>Γ.6.2-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-RT27.....</u>	<u>158</u>
<u>Γ.6.3-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-SP25A8.....</u>	<u>167</u>
<u>Γ.7-ΣΥΓΚΡΙΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ.....</u>	<u>176</u>
<u>Γ.8-ΡΟΥΤΙΝΕΣ.....</u>	<u>178</u>
<u>Γ.8.1-ΡΟΥΤΙΝΑ ΕΠΙΛΥΣΗΣ ΤΡΙΔΙΑΓΩΝΙΩΝ ΣΥΣΤΗΜΑΤΩΝ.....</u>	<u>178</u>
<u>Γ.8.2-ΡΟΥΤΙΝΑ ΥΠΟΛΟΓΙΣΜΟΥ ΕΝΕΡΓΟΥ ΘΕΡΜΟΧΩΡΗΤΙΚΟΤΗΤΑΣ ΓΙΑ ΤΟ ΥΛΙΚΟ RT20.....</u>	<u>179</u>
<u>Γ.8.3-ΡΟΥΤΙΝΑ ΥΠΟΛΟΓΙΣΜΟΥ ΕΝΕΡΓΟΥ ΘΕΡΜΟΧΩΡΗΤΙΚΟΤΗΤΑΣ ΓΙΑ ΤΟ ΥΛΙΚΟ RT27.....</u>	<u>180</u>
<u>Γ.8.4-ΡΟΥΤΙΝΑ ΥΠΟΛΟΓΙΣΜΟΥ ΕΝΕΡΓΟΥ ΘΕΡΜΟΧΩΡΗΤΙΚΟΤΗΤΑΣ ΓΙΑ ΤΟ ΥΛΙΚΟ SP25A8.....</u>	<u>181</u>
<u>Δ-ΒΙΒΛΙΟΓΡΑΦΙΑ.....</u>	<u>182</u>

ΠΙΝΑΚΑΣ ΣΥΜΒΟΛΩΝ

Λατινικά σύμβολα

C_p [kJ/kgK]: ειδική θερμοχωρητικότητα υπό σταθερή πίεση
 dx [m]: στοιχειώδες χωρικό βήμα
 dt [s]: στοιχειώδες χρονικό βήμα
 G [kW/m²]: στιγμιαία ολική ηλιακή ακτινοβολία οριζοντίου επιπέδου
 G_{bT} [kW/m²]: στιγμιαία άμεση ηλιακή ακτινοβολία κεκλιμένου επιπέδου
 G_{dT} [kW/m²]: στιγμιαία διάχυτη ηλιακή ακτινοβολία κεκλιμένου επιπέδου
 G_{ref} [W/m²]: Προσπίπτουσα ηλιακή ακτινοβολία αναφοράς
 G_{sc} [kW/m²]: ηλιακή σταθερά ($G_{sc}=1.3353$ kW/m²)
 G_T [kW/m²]: στιγμιαία ολική ηλιακή ακτινοβολία κεκλιμένου επιπέδου
 h [W/m²K]: ισοδύναμος συντελεστής μεταφοράς θερμότητας λόγω συναγωγής και ακτινοβολίας
 h_c [W/m²K]: ειδική συναγωγιμότητα
 h_r [W/m²K]: ισοδύναμος συντελεστής μεταφοράς θερμότητας λόγω ακτινοβολίας
 H [kJ/kg]: (λανθάνουσα) θερμότητα αλλαγής φάσης
 H_T [kJ/m²]: ωριαία ολική ηλιακή ακτινοβολία κεκλιμένου επιπέδου
 k [W/mK]: ειδική θερμική αγωγιμότητα
 L [m]: πάχος σώματος
 q [W/m²]: πυκνότητα παροχής θερμότητας
 Q [W]: ρεύμα θερμότητας
 Q_{cond} [W]: ρεύμα θερμότητας λόγω αγωγής
 Q_{conv} [W]: ρεύμα θερμότητας λόγω συναγωγής
 Q_{sol} [W]: ρεύμα θερμότητας λόγω ηλιακής ακτινοβολίας
 t [s]: χρόνος
 T [K ή °C]: θερμοκρασία
 T_{ref} [°C]: θερμοκρασία αναφοράς
 T_s, T_1 [K ή °C]: θερμοκρασία στην αρχή και στο τέλος της διαφασικής περιοχής, αντίστοιχα

Ελληνικά σύμβολα

α [m²/s]: θερμοκρασιακή αγωγιμότητα
 α_{glass} [-]: συντελεστής απορρόφησης γυαλιού
 β [°]: κλίση επιπέδου
 γ [°]: αζιμούθιο επιφάνειας
 δ [°]: ηλιακή απόκλιση
 ϵ [-]: συντελεστής εκπομπής σώματος
 θ [°]: γωνία πρόσπτωσης
 θ_z [°]: γωνία ζενίθ
 K_T [-]: μηνιαίος μέσος δείκτης διαύγειας
 ρ [kg/m³]: πυκνότητα
 σ [W/m²K⁴]: σταθερά Stefan-Boltzmann ($\sigma= 5.6697 \cdot 10^{-8}$ W/m²K⁴)
 Φ [W]: πηγή παραγωγής ή κατανάλωσης ενέργειας
 φ [°]: γεωγραφικό πλάτος
 ω [°]: ωριαία γωνία

Δείκτες

$i=0,1,2,\dots,n-1$: τρέχων χωρικός κόμβος

n : συνολικός αριθμός κόμβων

$p=0,1,2,\dots,m-1$: τρέχουσα χρονική στιγμή (ώρα)

m : συνολικός αριθμός χρονικών στιγμών

A-ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ

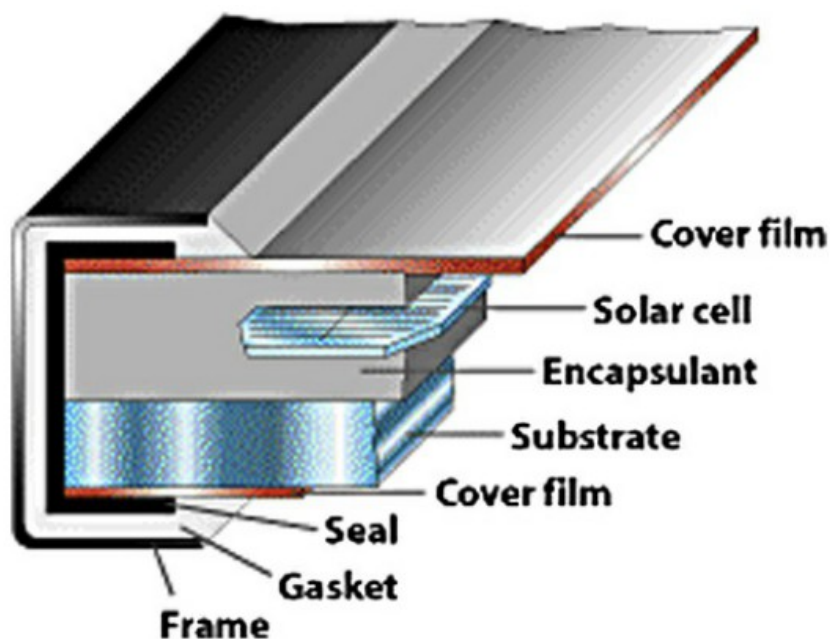
A.1-ΘΕΩΡΙΑ ΦΩΤΟΒΟΛΤΑΪΚΩΝ

A.1.1-ΓΕΝΙΚΑ [1], [2]

Η τεχνολογία φωτοβολταϊκών (photovoltaics-pv) δύναται να παρέχει ηλεκτρισμό προς όφελος της ανθρωπότητας και για μεγάλο εύρος εφαρμογών και περιβαλλοντικών συνθηκών. Τα φωτοβολταϊκά παρέχουν ηλεκτρικό ρεύμα ακόμα και σε απομακρυσμένες περιοχές και εγκαταστάσεις, όπου η πρόσβαση σε ένα κεντρικό ηλεκτροδοτικό δίκτυο είναι ανέφικτη.

Είναι η τεχνολογία με την οποία παράγεται ηλεκτρική ισχύς, συνεχούς ρεύματος (DC-direct current), όταν φωτόνια του ηλιακού φωτός προσπίπτουν σε ημιαγωγούς. Μια φωτοβολταϊκή μονάδα (photovoltaic module-pv module) απαρτίζεται από ηλιακές κυψέλες (solar cells). Μια τυπική φωτοβολταϊκή κυψέλη είναι συνήθως μικρού μεγέθους και παράγει 1-2 W ηλεκτρικής ισχύος. Για την ενίσχυση της παραγόμενης ισχύος οι κυψέλες συνδέονται σχηματίζοντας φωτοβολταϊκές μονάδες. Οι μονάδες αυτές, με τη σειρά τους, μπορούν να συνδεθούν, σχηματίζοντας συστοιχίες (pv arrays), ικανές για ακόμα υψηλότερη παραγόμενη ισχύ.

Τυπικές φωτοβολταϊκές συστοιχίες κάνουν χρήση μονάδων επίπεδης πλάκας (flat-plate pv module). Μια τυπική επίπεδη μονάδα περιλαμβάνει υπόστρωμα μετάλλου, γυαλιού ή πλαστικού για την στήριξη της κατασκευής στο πίσω μέρος, ένα περίβλημα (encapsulant material) για την προστασία των κυψελών και ένα διάφανο κάλυμμα από πολυμερές ή γυαλί. Η πλειονότητα των μονάδων κάνουν χρήση ενισχυμένου γυαλιού (tempered soda-lime glass), παρόμοιο με το ενισχυμένο γυαλί που χρησιμοποιείται στα παράθυρα. Χαρακτηρίζεται από υψηλή διαπερατότητα και η ανακλαστικότητα του γυαλιού μπορεί να μειωθεί μετά από επεξεργασία. Για την προστασία μονάδας, αυτή τοποθετείται σε μεταλλικό πλαίσιο (frame). Εγκάρσια τομή τυπικής μονάδας pv τύπου επίπεδης πλάκας παρουσιάζεται στο σχήμα A.1.1.1.



Σχήμα A.1.1.1: Τομή τυπικού επίπεδου pv panel [1].

Οι φωτοβολταϊκές συστοιχίες μπορεί να είναι πακτωμένες ή να τους επιτρέπεται να “ανιχνεύουν” τον ήλιο (μεταβλητή κλίση επιφάνειας για μέγιστη ηλιακή πρόσπτωση, tracking). Ανταποκρίνονται τόσο στην άμεση, όσο και στη διάχυτη ακτινοβολία, σε αντίθεση με άλλα ηλιακά συστήματα.

Ακόμα και σε μέρες με έντονη ηλιοφάνεια η διάχυτη ακτινοβολία αποτελεί το 10-20 % της ολικής για οριζόντιες επιφάνειες. Σε περιπτώσεις έντονης νέφωσης το ποσοστό της διάχυτης αυξάνεται κατά πολύ (50-100 %). Η απλούστερη συστοιχία pv αποτελείται από επίπεδες μονάδες, πακτωμένες στο έδαφος, με κλίση η οποία καθορίζεται από το γεωγραφικό πλάτος. Σημαντικό χαρακτηριστικό μιας πακτωμένης συστοιχίας είναι η απουσία κινητών-μηχανικών μερών, γεγονός που οδηγεί σε μειωμένο κόστος εγκατάστασης και συντήρησης και αυξημένη διάρκεια ζωής.

Στη συνέχεια παρατίθενται τα πλεονεκτήματα και τα μειονεκτήματα της φωτοβολταϊκής τεχνολογίας.

Πλεονεκτήματα:

- Η ενεργειακή πηγή είναι ευρέως προσβάσιμη και ουσιαστικά ανεξάντλητη.
 - Μηδενικές εκπομπές ρύπων ή παραγωγή αποβλήτων.
 - Χαμηλό κόστος λειτουργίας (δεν απαιτείται καύσιμο).
- Απουσία κινητών μερών, κάτι που οδηγεί σε μακροχρόνια λειτουργία.
 - Λειτουργία σε συνθήκες περιβάλλοντος.
- Υψηλή αξιοπιστία (κάποιοι κατασκευαστές εγγυώνται περισσότερο από 30 χρόνια λειτουργίας).
 - Προβλέψιμη ετήσια παραγωγή ισχύος.
 - Δυνατότητα ενσωμάτωσης σε νέα ή παλαιότερα κτίρια.
 - Ταχύτητα εγκατάστασης.
- Παραμετροποιήσιμη κατασκευή (μπορούν να προστεθούν μονάδες pv εκ των υστέρων).

Μειονεκτήματα:

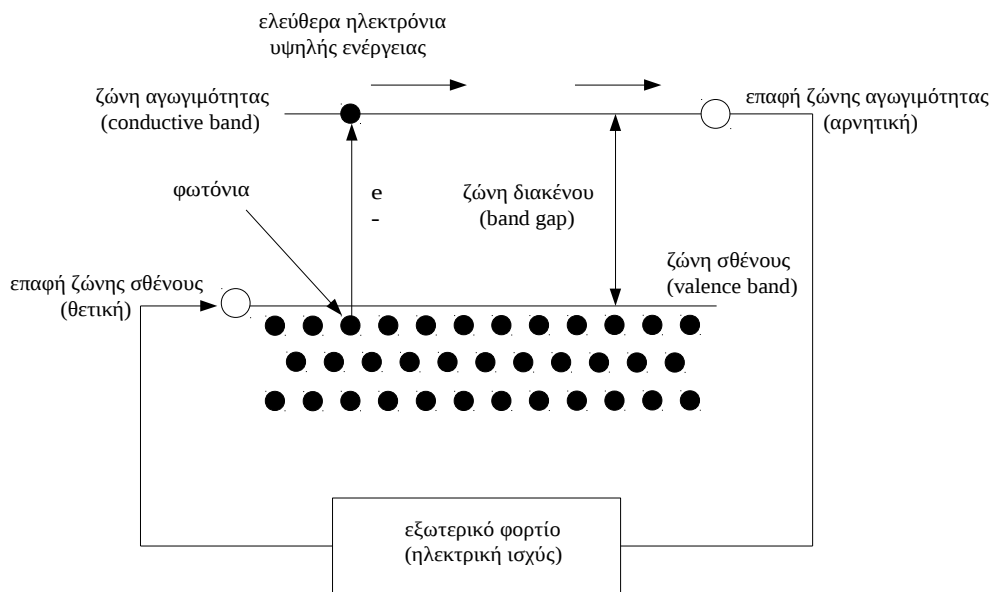
- Μεγάλες απαιτούμενες επιφάνειες.
- Υψηλό αρχικό κόστος εγκατάστασης.
- Απρόβλεπτη ωριαία ή ημερήσια ισχύς (νέφωση, μερική σκίαση κ.τ.λ.).
- Απουσία οικονομικά αποδοτικών μεθόδων αποθήκευσης ενέργειας.

A.1.2-ΛΕΙΤΟΥΡΓΙΑ ΦΩΤΟΒΟΛΤΑΪΚΩΝ [1], [2]

Οι φωτοβολταϊκές κυψέλες τυπικά κατασκευάζονται από ημιαγώγιμα υλικά, τα οποία φέρουν ασθενώς συνδεδεμένα ηλεκτρόνια, που βρίσκονται σε ενεργειακό επίπεδο γνωστό και ως ζώνη σθένους (valence band). Όταν ενέργεια, η οποία υπερβαίνει ένα συγκεκριμένο κατώφλι, γνωστό και ως ενέργεια ζώνης διακένου (bandgap energy), εφαρμόζεται σε ένα ηλεκτρόνιο ζώνης σθένους, οι δεσμοί σπάνε και το ηλεκτρόνιο απελευθερώνεται, ικανό να κινείται, πλέον, στη νέα ζώνη ενέργειας, τη ζώνη αγωγιμότητας (conduction band), στην οποία μπορεί να άγει ηλεκτρισμό διά μέσου του υλικού. Έτσι, τα ελεύθερα ηλεκτρόνια στη ζώνη αγωγιμότητας χωρίζονται, με τη ζώνη σθένους, από τη ζώνη διακένου. Η ενέργεια που αποκτά ή χάνει ένα ηλεκτρόνιο από τη φόρτιση ή αποφόρτιση του μετριέται σε eV (electronvolts, ηλεκτρονιοβόλτ). Η απαιτούμενη ενέργεια για την απελευθέρωση ενός ηλεκτρονίου παρέχεται από τα προσπίπτοντα, πάνω στο ημιαγώγιμο υλικό, φωτόνια (σωματίδια που αποτελούν το φως). Η ενέργεια ζώνης διακένου ή ενέργεια διακένου αποτελεί χαρακτηριστική παράμετρο κάθε ημιαγώγιμου υλικού. Καλή απορρόφηση της ηλιακής ενέργειας συνοδεύεται από ενέργεια διακένου της τάξης 1-2 eV.

Στο σχήμα A.1.2.1, παρουσιάζεται η σχέση μεταξύ ενέργειας (κάθετος άξονας) και χωρικών ορίων (οριζόντιος άξονας). Όταν η ηλιακή κυψέλη εκτίθεται σε φως επαρκούς ενέργειας, τα προσπίπτοντα φωτόνια απορροφώνται από τα άτομα, σπάζοντας τους δεσμούς των ηλεκτρονίων σθένους και ανεβάζοντάς τα σε υψηλότερη ενεργειακή στάθμη, τη ζώνη αγωγιμότητας. Εκεί, μια ειδική επαφή (n-type semiconductor, ημιαγώγος τύπου n) συλλέγει τα ηλεκτρόνια ζώνης αγωγιμότητας και τα οδηγεί σε εξωτερικό κύκλωμα. Τα ηλεκτρόνια χάνουν ενέργεια εκτελώντας

ωφέλιμο έργο εκτός της κυψέλης. Στη συνέχεια, επιστρέφονται στην ηλιακή κυψέλη μέσω δεύτερης ειδικής επιλεκτικής επαφής (p-type semiconductor, ημιαγωγός τύπου p), η οποία τα επαναφέρει στη ζώνη σθένους με ενέργεια ίση με την αρχική τους. Η ενέργεια στην οποία τα ηλεκτρόνια παραδίδονται στο εξωτερικό κύκλωμα είναι λιγότερη από το κατώφλι ενέργειας η οποία απαιτείται για τη διέγερση των ηλεκτρονίων (ενέργεια ζώνης διακένου). Είναι ανεξάρτητη από την ενέργεια του φωτονίου που τη δημιούργησε (υπό την προϋπόθεση ότι είναι υψηλότερη του κατωφλίου). Κατά συνέπεια, σε ένα υλικό με ενέργεια ζώνης διακένου 1 eV, ηλεκτρόνια διεγερμένα από φωτόνια ενέργειας 2 eV (κόκκινα) ή 3 eV (μπλε) θα έχουν και στις δύο περιπτώσεις τάση ελαφρώς μικρότερη του 1 V (και στις 2 περιπτώσεις τα ηλεκτρόνια παραδίδονται με ενέργεια περίπου 1 eV). Η ισχύς είναι ίση με το γινόμενο τάσης επί την ένταση ρεύματος, συνεπώς, η ισχύς μιας κυψέλης ισούται με το γινόμενο του αριθμού των ελεύθερων ηλεκτρονίων επί το φορτίο τους. Εντονότερο φως προκαλεί την απελευθέρωση περισσότερων ηλεκτρονίων, γεγονός που οδηγεί σε υψηλότερη παραγόμενη ισχύ.



Σχήμα A.1.2.1: Πορεία ηλεκτρονίου στο εσωτερικό μιας ηλιακής κυψέλης [2].

Το φως του ηλίου αποτελεί φάσμα φωτονίων που διανέμονται σε ένα εύρος ενέργειας. Τα φωτόνια που φέρουν ενέργεια μεγαλύτερη της ενέργειας ζώνης διακένου (ενεργειακό κατώφλι) μπορούν να διεγείρουν ηλεκτρόνια από τη ζώνη σθένους, στην ζώνη αγωγιμότητας, όπου μπορούν να βγουν από την μονάδα και να παράγουν ηλεκτρική ισχύ. Τα φωτόνια που φέρουν ανεπαρκή ποσά ενέργειας αποτυγχάνουν στη διέγερση των ηλεκτρονίων. Αντιθέτως, η ανεπαρκής αυτή ενέργεια ταξιδεύει δια μέσου της κυψέλης και απορροφάται, ως θερμότητα, στο πίσω μέρος της μονάδας. Ηλιακές κυψέλες απευθείας εκτεθειμένες στο ηλιακό φως μπορεί να έχουν έως και 20-30 °C υψηλότερη θερμοκρασία από τη θερμοκρασία περιβάλλοντος. Έτσι, οι κυψέλες παράγουν ηλεκτρικό ρεύμα χωρίς να λειτουργούν σε υψηλές θερμοκρασίες και δίχως κινητά μέρη, κάτι που δικαιολογεί το χαρακτηρισμό των μονάδων pn ως απλές και αξιόπιστες κατασκευές.

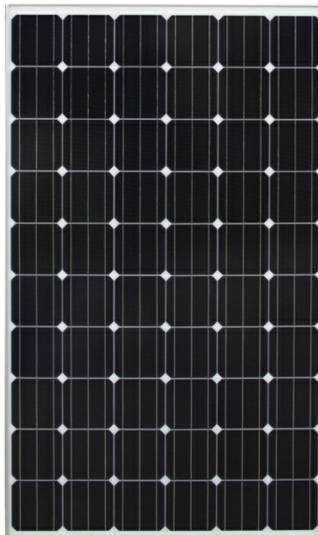
Στην καρδιά κάθε ηλιακής κυψέλης βρίσκεται η συμβολή pn (pn junction). Η μοντελοποίηση και κατανόηση απλοποιείται κατά πολύ χρησιμοποιώντας την έννοια της συμβολής pn. Η συμβολή pn προκύπτει από τη "νόθευση" (doping, επεξεργασία με στόχο τη μεταβολή-χειραγώγιση των ηλεκτρικών ιδιοτήτων ενός υλικού), διαδικασία που οδηγεί στη δημιουργία δύο επιλεκτικών επαφών. Η μία επαφή, επιλεκτική στη ζώνη σθένους, γίνεται η θετική πλευρά (p-side) και η άλλη, επιλεκτική στη ζώνη αγωγιμότητας (αρνητική πλευρά), γίνεται η αρνητική πλευρά (n-side).

Για πρακτικούς λόγους, συγκεκριμένος αριθμός ηλιακών κυψελών συνδέονται μεταξύ τους, σχηματίζοντας φωτοβολταϊκές μονάδες, που αποτελούν το τελικό προϊόν. Οι μονάδες pn παράγουν

συνεχές ρεύμα (DC) το οποίο, με τη σειρά του, μετατρέπεται στην πιο χρήσιμη μορφή του εναλλασσόμενου ρεύματος (AC), από ειδική ηλεκτρονική διάταξη, τον αναστροφέα. Ο αναστροφέας, οι επαναφορτιζόμενες μπαταρίες (όπου αυτό απαιτείται) και η μηχανική στήριξη για την έδραση και τον προσανατολισμό των μονάδων pv αποτελούν σημαντικά στοιχεία μιας φωτοβολταϊκής εγκατάστασης.

Οι περισσότερες φωτοβολταϊκές μονάδες στη σημερινή αγορά κατασκευάζονται από κυψέλες κρυσταλλικής σιλικόνης (crystalline silicone, c-Si). Περίπου το 10 % κατασκευάζονται με βάση τις ηλιακές κυψέλες λεπτού υμένα (thin-film solar cells), καλύπτοντας ένα ευρύ φάσμα τεχνολογιών (amorphous silicon (a-Si), copper indium gallium diselenide (CIGS, Cu(InGa)S₂), cadmium telluride (CdTe) κ.α.). Θεωρείται ότι οι κυψέλες λεπτού υμένα είναι πολλά υποσχόμενες στο θέμα μείωσης κόστους. Επίσης, υπάρχει και η διαρκώς αναπτυσσόμενη τεχνολογία των συγκεντρωτικών φωτοβολταϊκών (concentrator photovoltaics, CPV), όπου δαπανηρές αλλά αποδοτικές ηλιακές κυψέλες πολλαπλών συμβολών (multijunction solar cells, MJ) δέχονται ηλιακό φως υψηλής έντασης εστιασμένο από συγκεντρωτές (concentrators) κατασκευασμένους από κάτοπτρα ή φακούς.

Τέλος, υπό ανάπτυξη βρίσκεται και η τεχνολογία των οργανικών ηλιακών κυψελών. Η οργανική ηλιακή κυψέλη είναι ένας τύπος πολυμερούς ηλιακής κυψέλης που κάνει χρήση οργανικών ηλεκτρονικών (αγώγιμα οργανικά πολυμερή ή μικρά οργανικά μόρια).



Σχήμα Α.1.2.2: Μονοκρυσταλλική μονάδα PV (Suniva OPTimus 270 W, monocrystalline PV module) [1].

Α.1.3-ΤΑΞΙΝΟΜΗΣΗ ΦΩΤΟΒΟΛΤΑΪΚΩΝ ΜΟΝΑΔΩΝ ΚΑΙ ΓΕΝΝΗΤΡΙΩΝ [1], [2]

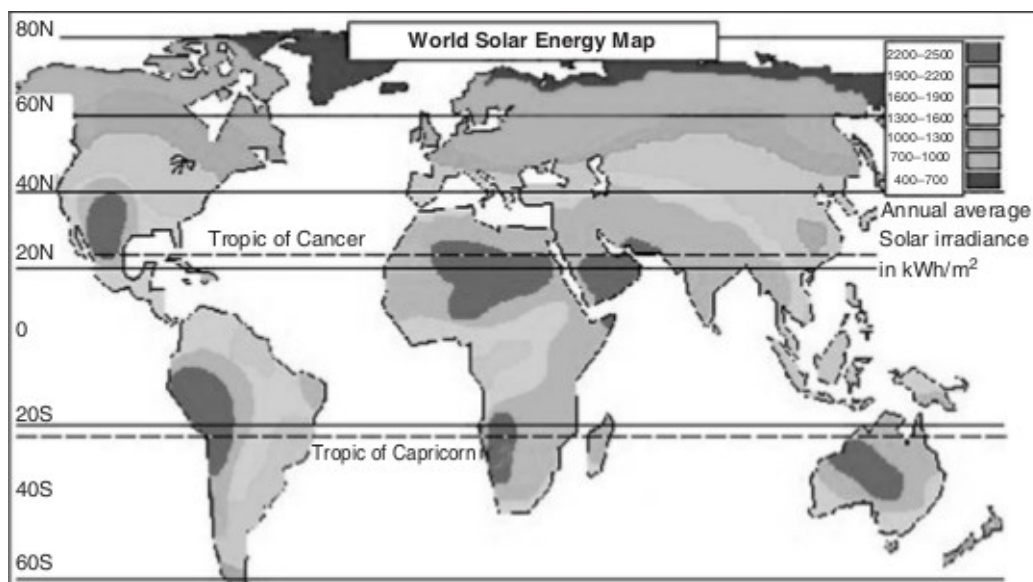
Μια γεννήτρια καυσίμου, ταξινομείται σε Watts (ή kW ή MW), είναι, δηλαδή, σχεδιασμένη να λειτουργεί παράγοντας αυτό το επίπεδο ισχύος, για όσο θα τροφοτεείται με καύσιμο, και να διαχέει την παραγόμενη θερμότητα. Εάν εξαναγκαστεί να λειτουργήσει σε επίπεδο μεγαλύτερο της χαρακτηριστικής ισχύος, θα καταναλώσει περισσότερο καύσιμο, θα υποστεί φθορές και, κατά συνέπεια, θα μειωθεί ο χρόνος ζωής της. Κάποιες γεννήτριες μπορούν να λειτουργήσουν σε χαμηλότερη ισχύ με μειωμένη απόδοση, ωστόσο οι περισσότερες συμπεριφέρονται απρόβλεπτα.

Αντίθετα, οι φωτοβολταϊκές μονάδες αξιολογούνται σε Watt μέγιστης ισχύος (W_p). Πρόκειται για την ισχύ που θα παρήγαγε η μονάδα με κατάλληλο συνδεδεμένο φορτίο υπό ηλιακή ακτινοβολία έντασης 1 kW/m^2 και θερμοκρασία κυψέλης σταθερή στους $25 \text{ }^\circ\text{C}$ (συνθήκες αναφοράς). Αντίστοιχα, μία φωτοβολταϊκή συστοιχία ταξινομείται με βάση το άθροισμα της μέγιστης παραγόμενης ισχύος κάθε μονάδας.

Αυτές οι συνθήκες αναφοράς ή STC (standard testing conditions), που εφαρμόζονται παγκοσμίως για τον υπολογισμό της μέγιστης ισχύος μιας ηλιακής κυψέλης σε συνθήκες εργαστηρίου, σπανίως λαμβάνουν χώρα στην πραγματικότητα. Γενικά, η προσπίπτουσα ηλιακή ακτινοβολία είναι χαμηλότερη και η θερμοκρασία κυψέλης υψηλότερη (σε ιδιαίτερα θερμές μέρες φτάνει έως και 30 °C πάνω από τη θερμοκρασία περιβάλλοντος), παράγοντες που ευθύνονται για την πτώση της παραγόμενης ισχύος για συγκεκριμένο συνδεδεμένο φορτίο. Σε κάποιες περιπτώσεις ακόμα και το φορτίο δεν ταιριάζει, οδηγώντας σε περαιτέρω μείωση της ισχύος. Έτσι, ενώ η παραγόμενη ισχύς είναι καθορισμένη σε συνθήκες αναφοράς (STC), κάτω από πραγματικές συνθήκες ποικίλει σημαντικά. Ενώ μια γεννήτρια diesel 10 kW, για όσο τροφοδοτείται με καύσιμο, παράγει ισχύ 10 kW, μια φωτοβολταϊκή συστοιχία 10 kW δύναται να παράγει 0-10 kW, ανάλογα με τη θερμοκρασία, την ένταση της ηλιοφάνειας και το συνδεδεμένο φορτίο.

Για ασφαλείς προβλέψεις, η ενέργεια σε kWh (και όχι η ισχύς) που παράγεται από μια φωτοβολταϊκή μονάδα κατά τη διάρκεια ενός έτους (ή μήνα ή ημέρας, αντίστοιχα) υπολογίζεται πολλαπλασιάζοντας τη χαρακτηριστική μέγιστη ισχύ σε kW_p (rated power) επί τον αριθμό των "ενεργών ωρών" που ηλιακή ακτινοβολία προσπίπτει στη μονάδα κατά τη διάρκεια ενός έτους επί τον συντελεστή απόδοσης (PR-performance ratio), ο οποίος λαμβάνει υπόψη τις απώλειες λόγω λειτουργίας σε πραγματικές συνθήκες, που αναφέρθηκαν προηγουμένως, συν τις απώλειες καλωδίωσης και αναστροφέα (ο βαθμός αποδοσής τους κυμαίνεται μεταξύ 0.90 και 0.97). Ο χρόνος που απαιτείται για συντήρηση, επίσης, λαμβάνεται υπόψη. Ο συντελεστής απόδοσης μιας καλοσχεδιασμένης εγκατάστασης κυμαίνεται μεταξύ 0.7 και 0.8 και σε θερμότερο κλίμα, πέφτει χαμηλότερα, καθώς οι επιδόσεις μειώνονται με την αύξηση της θερμοκρασίας κυψέλης.

Οι "ενεργές ώρες" (effective sun hours), λαμβάνοντας υπόψη ότι η ηλιακή ακτινοβολία αναφοράς είναι 1 kW/m², είναι ο συνολικός αριθμός ωρών στις οποίες η προσπίπτουσα ηλιακή ακτινοβολία σε επιφάνεια με την ίδια κλίση και προσανατολισμό με την μονάδα ρη είναι 1 kW/m². Εάν, για παράδειγμα, η μέση προσπίπτουσα ακτινοβολία κατά τη διάρκεια ενός 24-ώρου πάνω σε οριζόντια επιφάνεια είναι 4 kW/m², τότε οι "ενεργές ώρες" είναι 4. Στο σχήμα A.1.3.1 παρουσιάζεται η κατανομή της ετήσιας ηλιακής ακτινοβολίας, σε παγκόσμιο επίπεδο.



Σχήμα A.1.3.1: Παγκόσμια κατανομή ετήσιας ηλιακής ακτινοβολίας (kWh/m²) [2].

Σημαντική παράμετρος, σε εγκαταστάσεις με περιορισμένη επιφάνεια, αποτελεί και το μέγεθος της παραγόμενης ισχύος ανά μονάδα επιφάνειας. Εξίσου σημαντικός είναι και ο θερμοκρασιακός συντελεστής ισχύος (T_{coP}-temperature coefficient of power), ο οποίος εκφράζεται με την ποσοστιαία απώλεια ισχύος για κάθε μονάδα αυξανόμενης θερμοκρασίας (°C). Όσο περισσότερο

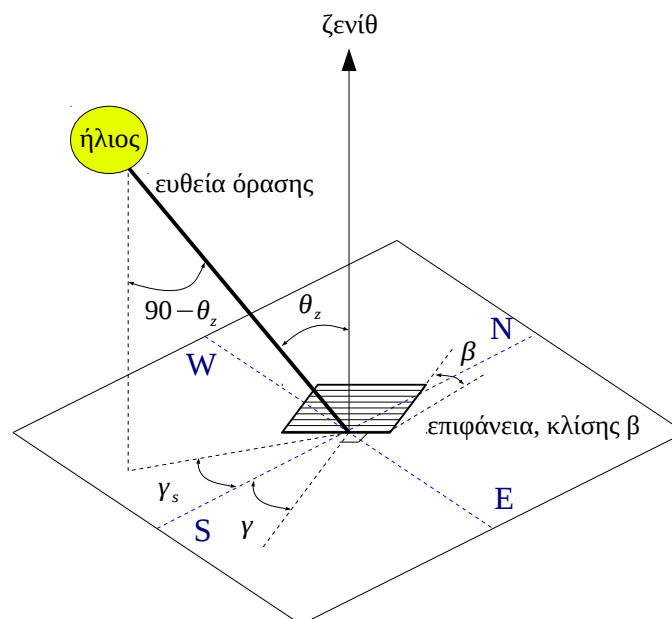
τείνει ο συντελεστής αυτός στο 0 (αρνητικός, όμως, πάντα), τόσο καλύτερες είναι και οι επιδόσεις της μονάδας κατά τη διάρκεια θερμών περιόδων. Για παράδειγμα, αν μια μονάδα έχει $T_{\text{cop}} -0.30 \%$, τότε η μονάδα θα απωλέσει το 6 % της παραγόμενης ισχύος με θερμοκρασιακή αύξηση $20 \text{ }^\circ\text{C}$ ($20 \text{ }^\circ\text{C} * 0.30 \text{ \%} / \text{ }^\circ\text{C} = 6 \text{ \%}$).

A.1.4-ΕΠΙΔΡΑΣΗ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ ΚΑΙ ΚΛΙΣΗΣ ΣΤΗΝ ΠΡΟΣΠΙΠΤΟΥΣΑ ΑΚΤΙΝΟΒΟΛΙΑ [3]

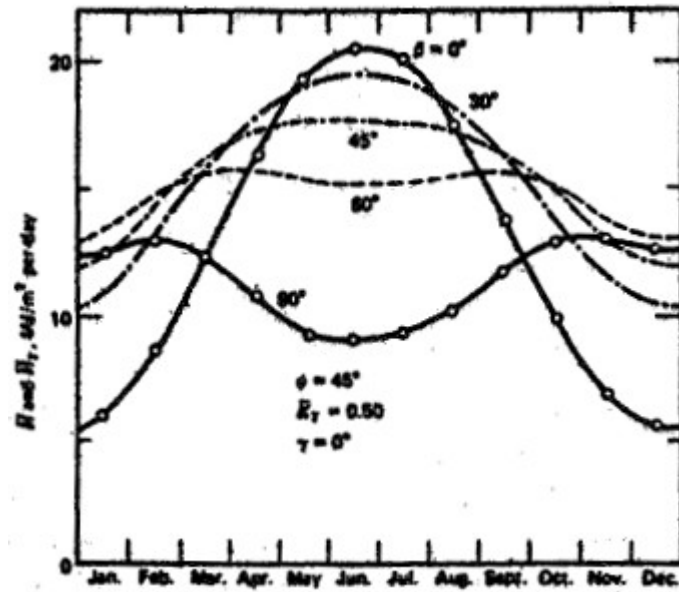
Η απόδοση μιας φωτοβολταϊκής μονάδας επηρεάζεται άμεσα από την ένταση της ηλιακής ακτινοβολίας. Η ένταση της ηλιακής ακτινοβολίας, με τη σειρά της, επηρεάζεται από την κλίση και τον προσανατολισμό της επιφάνειας της φωτοβολταϊκής μονάδας.

Μια κεκλιμένη επιφάνεια χαρακτηρίζεται από την κλίση της, β , ως προς το οριζόντιο επίπεδο και από το αζιμούθιο της, γ , γωνία που σχηματίζεται από την προβολή της καθέτου στην επιφάνεια πάνω στο οριζόντιο επίπεδο και από τη νότια κατεύθυνση ($\gamma=0^\circ, 90^\circ, 180^\circ$ και -90° για νότιο, δυτικό, βόρειο και ανατολικό προσανατολισμό αντίστοιχα). Είναι πολύ σημαντικό να είναι γνωστή η επίδραση των δύο αυτών παραμέτρων στην ποσότητα της προσπίπτουσας ενέργειας επί μηνιαίας, εποχικής ή ετήσιας βάσης.

Στο σχήμα A.1.4.2 δίνεται διάγραμμα μεταβολής της μέσης μηνιαίας ολικής ακτινοβολίας κεκλιμένου επιπέδου, H_T (kJ/m^2), συναρτήσει του μήνα και για διάφορες κλίσεις επιπέδου β και σταθερό αζιμούθιο ($\gamma=0^\circ$). Οι τιμές των παραμέτρων που υπεισέρχονται στους υπολογισμούς είναι: γεωγραφικό πλάτος $\varphi=45^\circ$, μηνιαίος μέσος δείκτης διαύγειας $K_T=0.5$ και ανακλαστικότητα εδάφους $\rho=0.2$. Είναι φανερό ότι άλλες κλίσεις δίνουν μεγαλύτερα ποσά ενέργειας κατά το χειμώνα και άλλες κατά το καλοκαίρι.



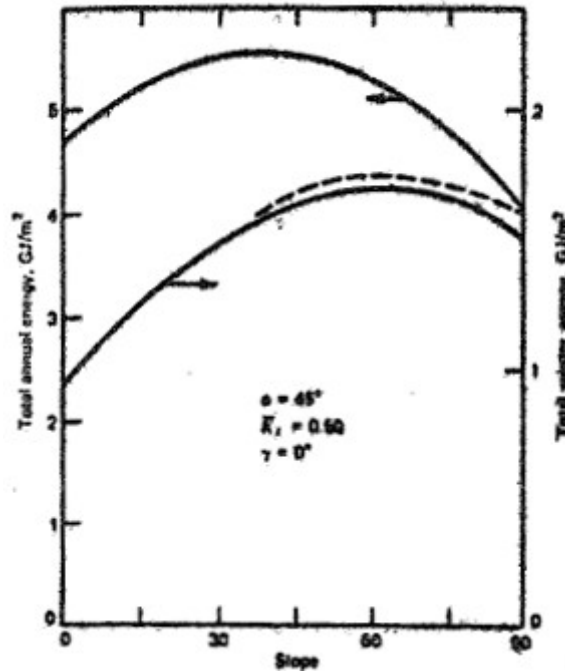
Σχήμα A.1.4.1: Κλίση επιπέδου β , αζιμούθιο επιφάνειας γ [3].



Σχήμα A.1.4.2: Μεταβολή της μέσης μηνιαίας ολικής ακτινοβολίας ημέρας κεκλιμένου επιπέδου συναρτήσει του μήνα του έτους για κεκλιμένο επίπεδο νότιου προσανατολισμού ($\gamma=0^\circ$) και διάφορες τιμές της κλίσης β [3].

Στο σχήμα A.1.4.3 δίνεται η προπίπτουσα ολική ετήσια ενέργεια (για τις ίδιες συνθήκες με το προηγούμενο σχήμα: $\gamma=0^\circ$, $\phi=45^\circ$, $K_T=0.5$, $\rho=0.2$) συναρτήσει της κλίσης β . Παρατηρείται μέγιστη ενέργεια για βέλτιστη τιμή της κλίσης $\beta=\phi$. Για μεταβολές της β κατά $\pm 20^\circ$ περί τη βέλτιστη τιμή, η μείωση της ετήσιας ενέργειας είναι μόνο 5%. Στο ίδιο σχήμα δίνεται και η ολική ενέργεια χειμώνα (Δεκέμβριος έως Μάρτιος) η οποία παρουσιάζει μέγιστο για βέλτιστη τιμή της κλίσης $\beta=\phi+15^\circ$, ενώ για μεταβολή της β κατά $\pm 15^\circ$ περί τη βέλτιστη τιμή, η μείωση της ενέργειας χειμώνα είναι μόνο 5%. Η διακεκομμένη τιμή αντιστοιχεί σε χιονισμένο έδαφος ($\rho=0.5$ για Ιανουάριο και Φεβρουάριο, $\rho=0.2$ για Δεκέμβριο και Μάρτιο).

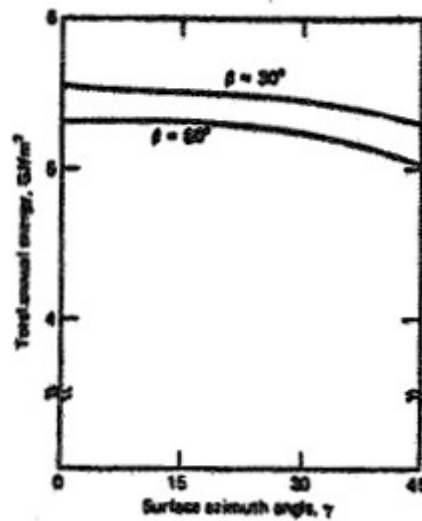
Στο σχήμα A.1.4.4 δίνεται η προσπίπτουσα ολική ετήσια ενέργεια συναρτήσει του αζιμούθιου γ της επιφάνειας για κλίσεις $\beta=30^\circ$ και $\beta=60^\circ$ (λοιπές συνθήκες όπως προηγουμένως, δηλαδή $\phi=45^\circ$, $K_T=0.5$, $\rho=0.2$). Παρατηρείται ότι ο βέλτιστος προσανατολισμός είναι ο νότιος ($\gamma=0^\circ$). Αποκλίσεις μέχρι και 20° ανατολικά ή δυτικά του νότου, ελάχιστα μειώνουν την προσπίπτουσα ολική ετήσια ενέργεια.



Σχήμα A.1.4.3: Μεταβολή της ετήσιας ολικής ενέργειας και της ολικής ενέργειας χειμώνα συναρτήσει της κλίσης επιπέδου με νότιο προσανατολισμό ($\gamma=0^\circ$) [3].

Από τα παραδείγματα των σχημάτων A.1.4.2, A.1.4.3 και A.1.4.4 εξάγονται οι παρακάτω χρήσιμοι κανόνες:

- (α) Για μέγιστη ετήσια ενέργεια απαιτείται κλίση επιφάνειας ίση με το γεωγραφικό πλάτος ($\beta=\phi$).
- (β) Για μέγιστη ενέργεια καλοκαιριού απαιτείται κλίση επιφάνειας κατά 15° μικρότερη του γεωγραφικού πλάτους ($\beta=\phi-15^\circ$).
- (γ) Για μέγιστη ενέργεια χειμώνα απαιτείται κλίση επιφάνειας κατά 15° μεγαλύτερη του γεωγραφικού πλάτους ($\beta=\phi+15^\circ$).
- (δ) Αποκλίσεις από τις παραπάνω τιμές προκαλούν μικρές μειώσεις της ενέργειας, πχ. απόκλιση του β κατά $\pm 15^\circ$ μειώνει την ενέργεια μόνο κατά 5 %.
- (ε) Ο βέλτιστος προσανατολισμός επιφάνειας είναι ο νότιος ($\gamma=0^\circ$) για το βόρειο ημισφαίριο και ο βόρειος ($\gamma=180^\circ$) για το νότιο ημισφαίριο, δηλαδή ο συλλέκτης πρέπει πάντα να είναι στραμμένος προς τον ισημερινό. Αποκλίσεις μέχρι και 20° προκαλούν μικρές μειώσεις της προσπίπτουσας ενέργειας.



Σχήμα Α.1.4.4: Μεταβολή της ετήσιας ολικής ενέργειας συναρτήσει του αζιμούθιου της επιφάνειας για κλίσεις 30° και 60° ($=45^\circ$, $K_T=0.5$, $\rho=0.2$) [3].

Α.1.5-ΜΕΡΗ ΦΩΤΟΒΟΛΤΑΪΚΟΥ ΣΥΣΤΗΜΑΤΟΣ [1], [2]

Τα φωτοβολταϊκά συστήματα μπορούν να είναι εξαιρετικά απλά, τα περισσότερα, ωστόσο, περιλαμβάνουν σειρά εξαρτημάτων, σχεδιασμένα να υποβοηθούν τις μονάδες pv στην παραγωγή της μέγιστης δυνατής ενέργειας και να συμβάλουν στην αξιόπιστη και ασφαλή μεταφορά της στο επιθυμητό φορτίο. Αν και συγκεκριμένα μέρη διαφέρουν από εγκατάσταση σε εγκατάσταση, κάποια εξαρτήματα είναι κοινά σε πληθώρα εφαρμογών, ειδικά στην περίπτωση των συστημάτων που είναι συνδεδεμένα σε ηλεκτρικό δίκτυο.

Φωτοβολταϊκές μονάδες

Ένας μικρός αριθμός τύπων φωτοβολταϊκών μονάδων κυριαρχεί στην αγορά σήμερα. Πάνω από το 99 % είναι τύπου επίπεδης πλάκας (flat-plate type), με τα συγκεντρωτικά φωτοβολταϊκά (concentrator pv) να αποτελούν τα υπόλοιπα. Οι κυψέλες πλακών σιλικόνης (silicon wafer-based solar cells) αποτελούν την κύρια συνισταμένη των μονάδων pv. Οι μονάδες αυτές παρουσιάζουν τις υψηλότερες αποδόσεις ανάμεσα στις εμπορικές και, τυπικά, κατασκευάζονται από ράβδους (ingots) μονοκρυσταλλικής (c-Si) ή πολυκρυσταλλικής σιλικόνης (multi-Si). Μαζί απαρτίζουν το 85 % των μονάδων που χρησιμοποιούνται παγκοσμίως, σε περίπου ίση αναλογία, με διάφορες μονάδες λεπτού υμένα από άμορφη σιλικόνη (a-Si) ή θειούχο κάδμιο/τελλουριούχο κάδμιο (cadmium sulfide CdS/cadmium telluride CdTe) να απαρτίζουν το υπόλοιπο 15 %. Η λωρίδα σιλικόνης (ribbon silicon) είναι παραλλαγή της πολυκρυσταλλικής σιλικόνης (die-drawn variant), η οποία κατασκευάζεται από έναν πολύ μικρό αριθμό εταιρειών.

Σημαντική αύξηση στην παραγωγή μονάδων c-Si και multi-Si έλαβε χώρα τα τελευταία χρόνια. Η αυξημένη εισροή μονάδων αυτού του τύπου από την Κίνα και άλλες χώρες χαμηλού κόστους παραγωγής οδήγησε, αναπόφευκτα, στη σημαντική μείωση κόστους και την αύξηση των επιδόσεων των συγκεκριμένων μονάδων.

Οι περισσότερες από τις υπόλοιπες, εμπορικά διαθέσιμες, μονάδες pv περιγράφονται ως φωτοβολταϊκά λεπτού υμένα (thin film), στις οποίες ένας αριθμός επιπρόσθετων υλικών, πέραν της σιλικόνης, χρησιμοποιείται. Η εισαγωγή της τεχνολογίας λεπτού υμένα ή άμορφης σιλικόνης έγινε στα μέσα του 1980. Γενικά, αποδίδουν λιγότερο από το μισό της απόδοσης των καλύτερων

μονάδων τύπου πλακών σιλικόνης (wafer Si type). Άλλα υλικά που χρησιμοποιούνται για την κατασκευή μονάδων pn είναι: τελουριούχο κάδμιο (CdTe, cadmium telluride), σεληνιούχο ίνδιο γάλλιο χαλκός (CIGS, copper indium gallium selenide), σεληνιούχο ίνδιο χαλκός (CIS, copper indium selenide) και αρσενικούχο κάδμιο (GaAs, gallium arsenide). Αξιοσημείωτο είναι ότι ο αριθμός των εγκατεστημένων μονάδων λεπτού υμένα διαρκώς αυξάνεται με έμφαση στα μεγάλα, επίγειας στήριξης, συστήματα. Αυτό οφείλεται στη σχετικά χαμηλή απόδοση των μονάδων, γεγονός που σημαίνει μεγαλύτερη απαιτούμενη επιφάνεια για συγκεκριμένη απαιτούμενη ισχύ. Οι μονάδες λεπτού υμένα σπανίως χρησιμοποιούνται σε εγκαταστάσεις όπου η επιφάνεια είναι περιορισμένη, όπως στην περίπτωση μιας οικιακής εγκατάστασης οροφής. Το κόστος τους έχει μειωθεί σημαντικά, κάτι που κάνει τη συγκεκριμένη τεχνολογία ελκυστική.

Τα σημαντικότερα τεχνολογικά χαρακτηριστικά για την αναγνώριση και διάκριση των μονάδων pn είναι οι ηλεκτρικές και φυσικές προδιαγραφές τους. Σημαντικά ηλεκτρικά χαρακτηριστικά αποτελούν η ονομαστική ισχύς, η τάση ανοικτού κυκλώματος V_{oc} , η ένταση βραχυκυκλώματος I_{sc} , η τάση σημείου μέγιστης ισχύος V_{mp} , η ένταση ρεύματος σημείου μέγιστης ισχύος I_{mp} , οι θερμοκρασιακοί συντελεστές και ο βαθμός απόδοσης. Τα σημαντικά φυσικά χαρακτηριστικά είναι οι διαστάσεις, το βάρος, το υλικό κάλυμματος, οι απαιτήσεις τοποθέτησης και η μέθοδος γείωσης.

Αναστροφείς

Τις μονάδες pn ακολουθούν, τόσο σε κόστος, όσο και σε σημασία, οι αναστροφείς (inverters). Για παραδείγμα, ένας αναστροφέας αντιπροσωπεύει, συνήθως, το 5-10 % του συνολικού κόστους μιας φωτοβολταϊκής εγκατάστασης εμπορικού τύπου, συνδεδεμένης σε ηλεκτρικό δίκτυο, και το 15-25 % του συνολικού κόστους ενός οικιακού συστήματος εκτός δικτύου. Ο αναστροφέας ηλεκτρικού ρεύματος είναι διάταξη η οποία μετατρέπει το παραγόμενο, από τη φωτοβολταϊκή συστοιχία, συνεχές ρεύμα σε εναλλασσόμενο ρεύμα, που χρησιμοποιείται στα περισσότερα σπίτια. Δέχεται σαν είσοδο το ρεύμα εξόδου της φωτοβολταϊκής συστοιχίας και το εξαναγκάζει σε ταλαντώσεις μέχρι η συχνότητα της ταλάντωσης να συμπέσει με τη συχνότητα ταλάντωσης του ηλεκτροδοτικού δικτύου. Η φωτοβολταϊκή συστοιχία συνδέεται στον αναστροφέα μέσω ενός διακόπτη αποσύνδεσης DC (DC disconnect switch). Παρομοίως, ένας διακόπτης αποσύνδεσης AC (AC disconnect switch) παρεμβάλλεται μεταξύ της εξόδου του αναστροφέα και του ηλεκτρικού δικτύου εναλλασσόμενου ρεύματος. Αυτοί οι διακόπτες επιτρέπουν την ασφαλή συντήρηση και επισκευή, απενεργοποιώντας τον αναστροφέα. Συχνά, οι διακόπτες αποσύνδεσης ενσωματώνονται στο πλαίσιο του αναστροφέα. Επίσης, υπάρχουν και διατάξεις προστασίας από υπερένταση (over-current) και στις δύο πλευρές (πλευρές AC και DC). Επιπλέον προστασία, συνήθως, περιλαμβάνεται στις πλευρές AC και DC για την αντιμετώπιση παροδικών υπερτάσεων π.χ. στην περίπτωση κεραυνού. Ένα βασικό μέρος του αναστροφέα είναι ο ελεγκτής. Πρόκειται για την υλοποίηση αλγορίθμων που ελέγχουν τα τρανζίστορ, παρέχει παρακολουθήση σημείου μέγιστης ισχύος (MPPT, maximum power point tracking), ικανοποιεί τις διαδραστικές απαιτήσεις του δικτύου, μετράει την είσοδο και την έξοδο του συστήματος και μεταφέρει πληροφορίες σε χρήστες και συστήματα παρακολούθησης/ελέγχου. Για συστήματα εκτός δικτύου, ο ελεγκτής περιλαμβάνει και τη λειτουργία φόρτισης της μπαταρίας. Οι περισσότεροι αναστροφείς έχουν βαθμό απόδοσης της τάξεως 85-96 %, ανάλογα με την εταιρεία κατασκευής και το μοντέλο. Οι απώλειες καλωδίωσης, διακοπών καθώς και μετατροπής του ρεύματος από DC σε AC θα πρέπει να λαμβάνονται υπόψη στον υπολογισμό του απαιτούμενου συνολικού αριθμού φωτοβολταϊκών μονάδων.

Ένας τυπικός γραμμικός αναστροφέας (γραμμικός γιατί συνδέεται σε σειριακή συστοιχία pn μονάδων), παρουσιάζεται στο σχήμα A.1.5.1. Μετατρέπει το συνεχές ρεύμα, έξοδο της φωτοβολταϊκής συστοιχίας, σε συμβατό με το ηλεκτροδοτικό δίκτυο εναλλασσόμενο ρεύμα. Σε διάταξη γραμμικού αναστροφέα, οι συνδεδεμένες κατά σειρά μονάδες παρέχουν συσσωρευμένο DC ρεύμα στον αναστροφέα προς μετατροπή σε AC ρεύμα και στη συνέχεια το ρεύμα αυτό

διοχετεύεται στο ηλεκτρικό δίκτυο.



Σχήμα A.1.5.1: Τυπικός γραμμικός αναστροφέας [1].

Το κυριότερο μειονέκτημα μιας διάταξης γραμμικού αναστροφέα είναι ότι περιορίζεται από τη χειρότερη, σε όρους επιδόσεων, μονάδα. Με άλλα λόγια, η μέγιστη απόδοση μετατροπής του αναστροφέα καθορίζεται από τη μονάδα που αποδίδει λιγότερο. Για παράδειγμα, εάν μια μονάδα είναι μερικώς επισκιασμένη και χάνει 40 % της απόδοσης της, κάθε μονάδα επίσης, περιορίζεται στο 40 % της δυνητικά παραγόμενης ισχύος εξόδου. Οι αναστροφείς χρησιμοποιούν μία τεχνική γνωστή σαν ανίχνευση σημείου μέγιστης ισχύος ή MPPT (maximum power point tracking), για τη βελτιστοποίηση της εξόδου του συστήματος PV, ρυθμίζοντας τα συνδεδεμένα φορτία.

Πληθώρα γωνιών και μη παραδοσιακών διατάξεων οροφών και σκεπών, αποτελούν πρόβλημα για συγκεκριμένα συστήματα γραμμικών αναστροφέων καθώς, για τη μέγιστη παραγόμενη ισχύ πρέπει όλες οι μονάδες να εκτίθενται στην ίδια ένταση ηλιακής ακτινοβολίας. Επομένως, όλες οι μονάδες θα πρέπει να έχουν την ίδια κλίση πρόσπτωσης και τον ίδιο προσανατολισμό. Επιπλέον, οι γραμμικοί αναστροφείς έχουν περιορισμένες διαβαθμίσεις ισχύος, κάτι που σημαίνει ότι, οι διαβαθμίσεις ισχύος των φωτοβολταϊκών μονάδων θα πρέπει να συμπίπτουν με τις διαβαθμίσεις ισχύος του γραμμικού αναστροφέα. Το γεγονός αυτό θέτει περιορισμό στην επέκταση της φωτοβολταϊκής συστοιχίας.

Ένας εναλλακτικός τύπος αναστροφέα είναι ο μικροαναστροφέας, ο οποίος μετατρέπει το συνεχές ρεύμα μιας PV μονάδας σε, συμβατό με το ηλεκτρικό δίκτυο, εναλλασσόμενο ρεύμα. Πρόκειται για μικρούς αναστροφείς ικανούς να χειρίζονται την ισχύ εξόδου μίας μόνο μονάδας. Κάθε φωτοβολταϊκή μονάδα έχει το δικό της μικροαναστροφέα. Συστοιχίες μονάδων συνδέονται, μεταξύ τους, παράλληλα και μέσω του αντίστοιχου μικροαναστροφέα, συνεισφέρουν το αντίστοιχο εναλλασσόμενο ρεύμα. Η διάταξη αυτή παρουσιάζει, σε σχέση με τη διάταξη συνεχούς ρεύματος ενός γραμμικού αναστροφέα, το πλεονέκτημα της απομόνωσης κάθε μονάδας. Μείωση ή ακόμη και πιθανή απώλεια της ισχύος εξόδου μιας μονάδας δεν επηρεάζει την απόδοση ολόκληρης της συστοιχίας. Η χρήση μικροαναστροφέων και ο επακόλουθος ανεξάρτητος έλεγχος κάθε μονάδας PV, ελαχιστοποιεί την επιρροή της σκίασης, νέφωσης και αστοχίας μιας μονάδας, στο μέγεθος της παραγόμενης ισχύος. Επίσης, κάθε μικροαναστροφέας μπορεί να διατηρεί βέλτιστη ισχύ εκτελώντας MPPT της δικής του ανεξάρτητης μονάδας. Τυπική διάταξη με μικροαναστροφείς

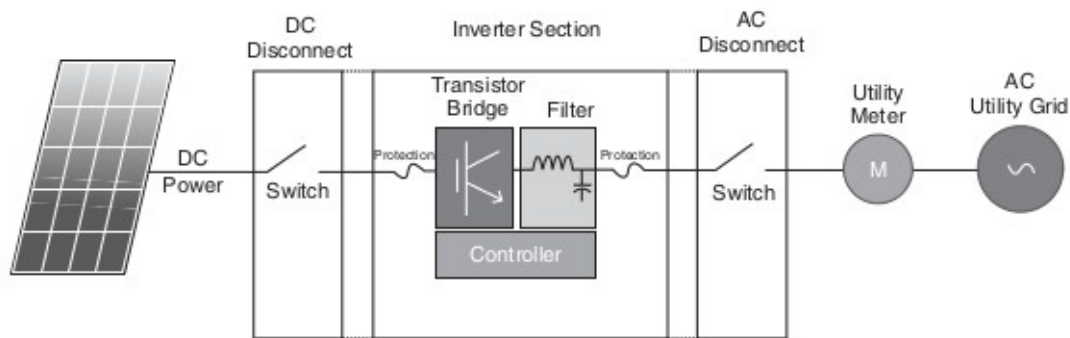
παρουσιάζεται στο σχήμα A.1.5.2. Αντίξοες καιρικές συνθήκες είναι πιθανότερο να επηρεάσουν πολλαπλούς μικροαναστροφείς, παρά έναν καλά τοποθετημένο γραμμικό αναστροφέα. Το κόστος μπορεί να είναι απαγορευτικό, ανάλογα με τον αριθμό των PV μονάδων. Αξίζει να σημειωθεί ότι, ένα σύστημα με μικροαναστροφείς είναι παραμετροποιήσιμο με την εξής έννοια: μπορούν να προστεθούν κι άλλες μονάδες αργότερα απλά επεκτείνοντας την καλωδίωση εναλλασσόμενου ρεύματος μέχρι και τις επόμενες μονάδες PV. Επίσης, συστήματα μικροαναστροφέων, μπορούν να ελέγχονται ανεξάρτητα κάνοντας τη συντήρησή τους απλούστερη. Συστήματα μικροαναστροφέων προτείνονται για συστήματα χαμηλής παραγόμενης ισχύος (κάτω των 3 kW) και για οροφές με "δύσκολες" γεωμετρίες και προσανατολισμούς.



Σχήμα A.1.5.2: Φωτοβολταϊκή συστοιχία με μικροαναστροφείς για κάθε μονάδα PV [2].

Σε ό,τι αφορά τα συστήματα που συνδέονται με το ηλεκτρικό δίκτυο, ο αναστροφέας πρέπει να ρυθμίζει το εναλλασσόμενο ρεύμα το οποίο ρέει προς το δίκτυο, καλύπτοντας τις απαιτήσεις του εκάστοτε δικτύου και ακολουθώντας τις προδιαγραφές που έχουν τεθεί.

Οι αναστροφείς φωτοβολταϊκών συστημάτων που δε συνδέονται σε δίκτυο απαιτείται να ρυθμίζουν την τάση εναλλασσόμενου ρεύματος που τροφοδοτούν στα τοπικά φορτία, κάτι που με τη σειρά του απαιτεί ταχεία και δυναμική απόκριση για τη σταθερότητα του συστήματος. Μια εξωτερική αποθηκευτική διάταξη (π.χ. μπαταρία), αποτελεί μια δευτερεύουσα μεταβατική πηγή ισχύος, ικανή να καλύπτει τις ανάγκες του συστήματος. Συχνά, ο έλεγχος της ροής ισχύος από και προς την αποθηκευτική διάταξη ρυθμίζεται από ένα ρυθμιστή φόρτισης (charge controller), διάταξη που μπορεί αποτελεί ξεχωριστή συσκευή ή να συμπεριλαμβάνεται στον αναστροφέα.



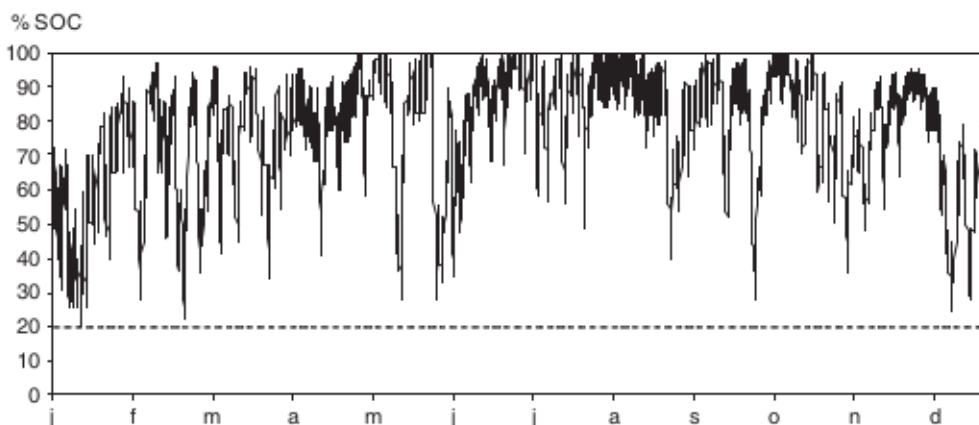
Σχήμα A.1.5.3: Σημαντικά μέρη αναστροφέα δικτύου [2].

Επιπρόσθετος εξοπλισμός

Ο επιπρόσθετος εξοπλισμός (BOS, balance of system) περιλαμβάνει καλωδίωση, ασφάλειες, γειώσεις, διακόπτες και μετρητές. Το υλικό αυτό θα πρέπει να είναι ανθεκτικό ώστε να αντέξει για πολλά χρόνια εκτεθειμένο στο εξωτερικό περιβάλλον, κάτω από συνθήκες υγρασίας, ακραίων θερμοκρασιών, διάβρωσης και υπερϊώδους ακτινοβολίας. Επίσης, θα πρέπει να είναι κατάλληλα επιλεγμένο για την επιθυμητή τάση λειτουργίας και το περιβάλλον στο οποίο καλείται να λειτουργήσει. Η καλωδίωση, που προορίζεται για φωτοβολταϊκά, συστήματα διατίθεται για πολλά επίπεδα τάσεων και θερμοκρασιών λειτουργίας.

Αποθηκευτικές διατάξεις

Η αποθήκευση ενέργειας θεωρείται άκρως απαραίτητη για όλα τα συστήματα εκτός δικτύου, κάτι που συνήθως σημαίνει αποθήκευση παραγόμενου ηλεκτρισμού, από φωτοβολταϊκό σύστημα, σε μπαταρία. Αν και μεγάλη ποικιλία τεχνολογιών μπαταρίας χρησιμοποιούνται ολοένα και περισσότερο (Pb-Sb, Li-ion, NiMH, NiCd, Zn-air και άλλες), οι τύποι μολύβδου-οξέως (lead-acid types) κυριαρχούν εδώ και πολύ καιρό σαν μορφή αποθήκευσης. Οι περισσότερες ηλιακές εφαρμογές κάνουν χρήση παραλλαγών βαθείας αποφόρτισης (deep discharge) της τεχνολογίας μπαταριών μολύβδου-οξέως, οι οποίες φέρουν παχύτερες πλάκες μολύβδου για αυξημένη διάρκεια ζωής και αντοχή. Αποτελεί κοινή σχεδιαστική πρακτική να εκτιμώνται, έτσι ώστε να μην αποφορτίζονται συχνά περισσότερο από 50 %. Επίσης, κοινή λειτουργική πρακτική αποτελεί η περιοδική φόρτιση. Κατάλληλα επιλεγμένες και συντηρημένες μπαταρίες μπορούν να λειτουργούν ικανοποιητικά για περισσότερο από 10 χρόνια.



Σχήμα A.1.5.4: Διάγραμμα κατάστασης μπαταρίας απομακρυσμένου συστήματος pv κατά τη διάρκεια ενός έτους [2].

Η αποθηκευτική διάταξη θα πρέπει να μπορεί να αποζημιώνει για τις περιόδους με φορτίο και ανεπαρκή ηλιακή ακτινοβολία (π.χ. κατά τη διάρκεια νύχτας ή συννεφιασμένου καιρού). Οι ημερήσιοι και εποχικοί κύκλοι φόρτισης-αποφόρτισης καθορίζουν το μέγεθος και το σχεδιασμό της μπαταρίας. Σε έναν τυπικό ημερήσιο κύκλο, η μπαταρία φορτίζεται τη μέρα και αποφορτίζεται τη νύχτα. Οι εποχικοί κύκλοι είναι αποτέλεσμα της μεταβλητής ηλιακής ακτινοβολίας και των μεταβατικών απαιτήσεων των φορτίων. Οι κύκλοι, αυτοί, αποτελούν τις κύριες παραμέτρους που επηρεάζουν τη διάρκεια ζωής μιας καλοσυντηρημένης μπαταρίας, αν και άλλοι παράμετροι, όπως η θερμοκρασία της μπαταρίας, η ένταση του ρεύματος και η τάση, παίζουν σημαντικό ρόλο. Στο σχήμα A.1.5.4 φαίνεται η οριακή ικανότητα να διατηρεί επαρκές φορτίο κατά τη διάρκεια του χειμώνα και η χαρακτηριστική ευκολία με την οποία αυτό επιτυγχάνεται κατά τη διάρκεια του καλοκαιριού. Τα εποχικά μέγιστα κυμαίνονται από 20 % το χειμώνα μέχρι 100 % το καλοκαίρι, με ετήσιο μέσο 70 %. Το χειμώνα ο ελεγκτής φόρτισης αποσυνδέει φορτία περιστασιακά για την αποφυγή βλάβης της μπαταρίας. Συνήθως, σε μηνιαία βάση, η μπαταρία θα είναι πλήρως φορτισμένη το 30 % των ημερών και κατά προτίμηση περισσότερο, για την εξασφάλιση της μακροζωίας της. Ανάλογα με το πόσο προσεκτικά ο χρήστης διαχειρίζεται τη φωτοβολταϊκή εγκατάσταση, η μπαταρία μπορεί να υπόκειται σε μεγάλες περιόδους ενδιάμεσης φόρτισης χωρίς να φτάνει ποτέ τη μέγιστη αποφόρτιση, ακόμα και κάτω από συνθήκες ανεπαρκούς ηλιακής ακτινοβολίας, κάτι που οδηγεί στην αύξηση της διάρκειας ζωής της.

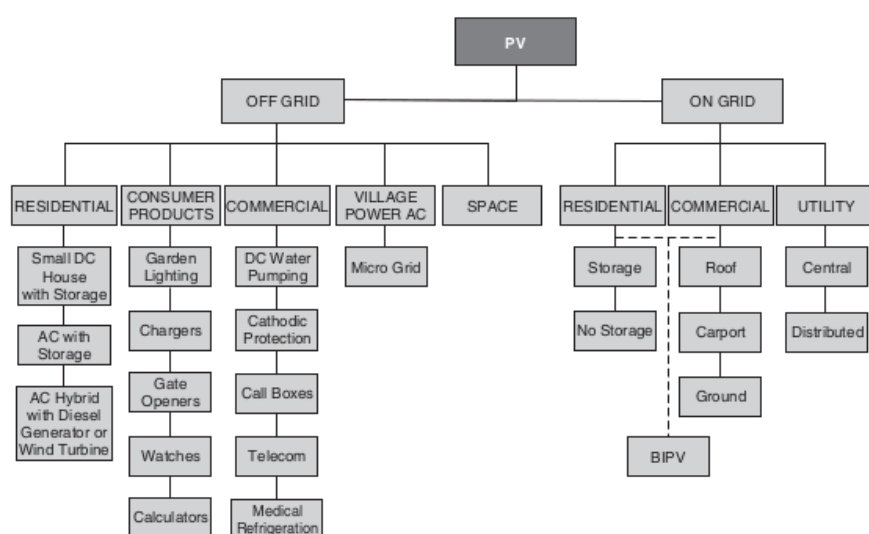
Ρυθμιστές φόρτισης

Τυπικά, ο σχεδιασμός φωτοβολταϊκών συστημάτων εκτός δικτύου, υψηλής αξιοπιστίας, οδηγεί σε υπερμεγέθεις εγκαταστάσεις, καθώς θα πρέπει να παρέχεται επαρκής ισχύς για μέγιστο φορτίο και κάτω από αντίξοες συνθήκες (συννεφιασμένη μέρα). Κατά συνέπεια, παράγει, πολύ περισσότερη ενέργεια από αυτή που απαιτείται πολλές φορές κατά τη διάρκεια ενός έτους. Χωρίς την ύπαρξη δικτύου που αποδέχεται αυτό το ενεργειακό περίσσειμα, θα πρέπει το σύστημα pv να αδρανοποιείται περιστασιακά για την αποφυγή της υπερφόρτισης των μπαταριών. Ο σκοπός του ρυθμιστή φόρτισης (charge controller) είναι, ουσιαστικά, η διαχείριση της κυκλοφορίας: οδηγεί ρεύμα στις μπαταρίες όταν χρειάζεται, επιτρέπει την ροή ρεύματος κατ' απαίτηση των φορτίων, διατηρεί την συστοιχία στο σημείο μέγιστης ισχύος (κάποιοι ρυθμιστές φόρτισης δεν παρέχουν αυτή τη δυνατότητα), και, πάνω απ' όλα, αποτρέπει την υπερφόρτιση της μπαταρίας είτε με αποσύνδεση είτε απενεργοποιώντας συγκεκριμένα εξαρτήματα.

A.1.6-ΤΥΠΟΙ ΦΩΤΟΒΟΛΤΑΪΚΩΝ ΣΥΣΤΗΜΑΤΩΝ [1], [2]

Τα φωτοβολταϊκά έχουν πληθώρα χρήσεων, με διάφορους βαθμούς πολυπλοκότητας συστημάτων. Ένα εύρος εφαρμογών φαίνεται στο σχήμα A.1.6.1. Οι εφαρμογές που προορίζονται εκτός δικτύου παρουσιάζουν κάποιες κοινές ιδιότητες με τις εφαρμογές δικτύου, τα συστήματά τους, ωστόσο, καλύπτουν διαφορετικές ανάγκες. Για παράδειγμα, συστήματα εντός και εκτός δικτύου μπορεί να χρησιμοποιούν την ίδια τεχνολογία μονάδας, να έχουν την ίδια εγκατάσταση, να προορίζονται για το ίδιο κλίμα και να παρέχουν την ίδια ισχύ σε έναν υποθετικό πελάτη. Το κόστος εγκατάστασης και συντήρησης ανά μονάδα ισχύος ενός συστήματος εντός δικτύου θα κοστίζει λιγότερο και θα λειτουργεί αποδοτικότερα από το αντίστοιχο σύστημα εκτός δικτύου.

Στην περίπτωση που δεν υπάρχει πρόσβαση σε ηλεκτρικό δίκτυο, είναι, συνήθως, οικονομικά απαγορευτική η επέκταση του δικτύου σε απομακρυσμένη περιοχή. Σε τέτοιες περιπτώσεις, παρά τη χαμηλότερη απόδοση τους και το σχετικά υψηλότερο κόστος, μονάδες PV εκτός δικτύου αποτελούν τη βέλτιστη λύση σε σχέση με παραδοσιακές λύσεις (π.χ. γεννήτριες ορυκτών καυσίμων, μπαταρίες). Ακολουθεί η περιγραφή κοινών φωτοβολταϊκών συστημάτων.

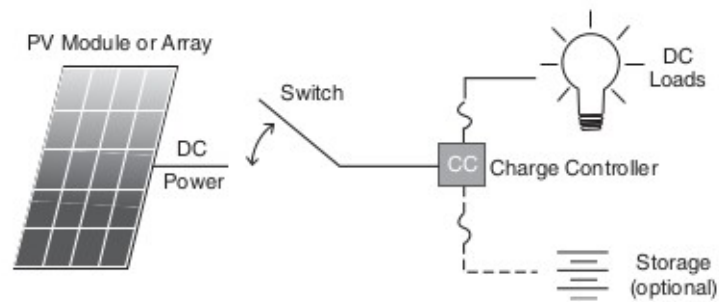


Σχήμα A.1.6.1: Πίνακας ταξινόμησης φωτοβολταϊκών συστημάτων [2].

Μικρό σύστημα συνεχούς ρεύματος εκτός δικτύου (small off-grid DC system).

Τα συστήματα συνεχούς ρεύματος περιλαμβάνουν εφαρμογές οι οποίες χρησιμοποιούν απευθείας την παραγόμενη, από φωτοβολταϊκές μονάδες, ισχύ συνεχούς ρεύματος σε φορτία συνεχούς ρεύματος.

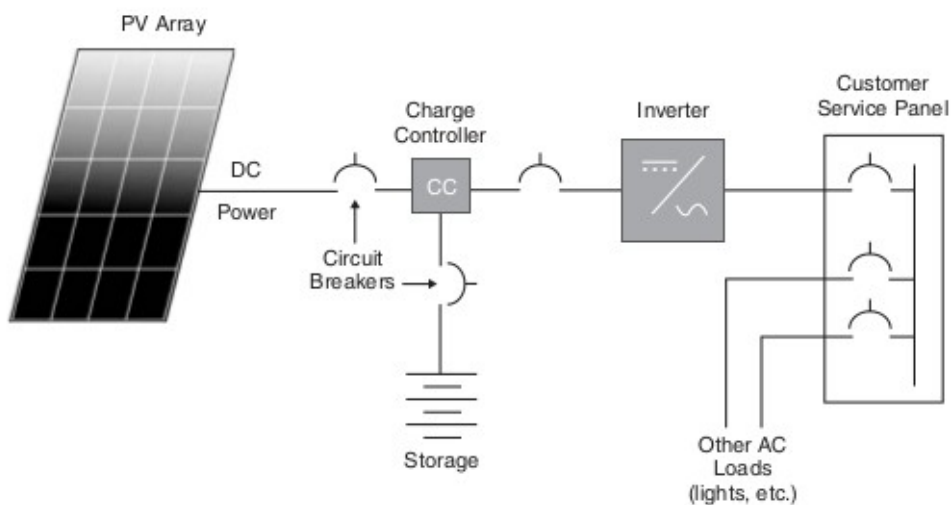
Αυτό περιλαμβάνει συστήματα που προορίζονται για διαστημικές χρήσεις (π.χ. δορυφόροι), φορητές ηλιακές συσκευές και μικρές εμπορικές εφαρμογές, οικιακά συστήματα, αντλίες νερού και άλλες μικρές εφαρμογές, μικρότερες του 1 kW σε μέγεθος και στις οποίες το φορτίο απαιτεί συνεχές ρεύμα για να λειτουργήσει. Απλοποιημένο διάγραμμα τέτοιου συστήματος, παρατίθεται στην συνέχεια (σχήμα A.1.6.2):



Σχήμα Α.1.6.2: Απλοποιημένο σχήμα συστήματος DC PV, εκτός δικτύου [2].

Σύστημα εναλλασσόμενου ρεύματος εκτός δικτύου (off-grid AC system).

Εκτός δικτύου συστήματα είναι αυτά στα οποία η ισχύς του φωτοβολταϊκού μετατρέπεται σε ισχύ εναλλασσόμενου ρεύματος, χωρίς να υπάρχει διαθέσιμο ηλεκτρικό δίκτυο. Τα φορτία σε αυτόν τον τύπο συστήματος λειτουργούν με εναλλασσόμενο ρεύμα. Ο αναστροφέας (inverter), σε αυτές τις περιπτώσεις, λειτουργεί ρυθμίζοντας την τάση AC σε όλα τα φορτία. Αποθήκευση ενέργειας (π.χ. μπαταρίες), συνήθως, λαμβάνει χώρα σε συστήματα εκτός δικτύου, για την αποκατάσταση της ισορροπίας μεταξύ της διακοπτόμενης φωτοβολταϊκής πηγής και των απαιτήσεων του φορτίου (π.χ. νυκτερινός φωτισμός). Κλασικό παράδειγμα αποτελεί το απομακρυσμένο σπίτι με σύστημα που απαρτίζεται από μονάδες PV στην οροφή του, έναν αναστροφέα εκτός δικτύου και μια αποθηκευτική διάταξη. Ακολουθεί απλοποιημένο σχήμα (σχήμα Α.1.6.3):

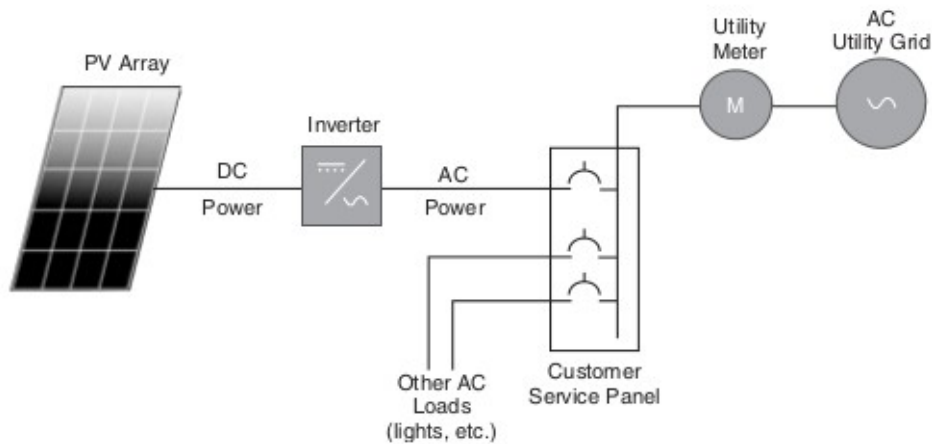


Σχήμα Α.1.6.3: Απλοποιημένο σχήμα συστήματος AC PV, εκτός δικτύου [2].

Συστήματα δικτύου (on-grid systems)

Συστήματα συνδεδεμένα στο δίκτυο, είναι αυτά που η παραγόμενη, από τα φωτοβολταϊκά, ισχύς συνεχούς ρεύματος μετατρέπεται σε ισχύ εναλλασσόμενου ρεύματος και, είτε χρησιμοποιείται απ'ευθείας είτε τροφοδοτείται στο δίκτυο. Η μετατροπή αυτή επιτυγχάνεται μέσω του αναστροφέα. Όπως σε κάθε εγκατάσταση παραγωγής ηλεκτρικής ισχύος θα πρέπει το εναλλασσόμενο ρεύμα να είναι συντονισμένο με το συνδεδεμένο ηλεκτρικό δίκτυο, κάτι που περιλαμβάνει την τάση και τη συχνότητα του εναλλασσόμενου ρεύματος. Έτσι, η παραγόμενη ισχύς επιμερίζεται στα φορτία του

δικτύου και όχι σε συγκεκριμένο εξοπλισμό. Ακολουθεί σχήμα του συστήματος (Σχήμα Α.1.6.4):



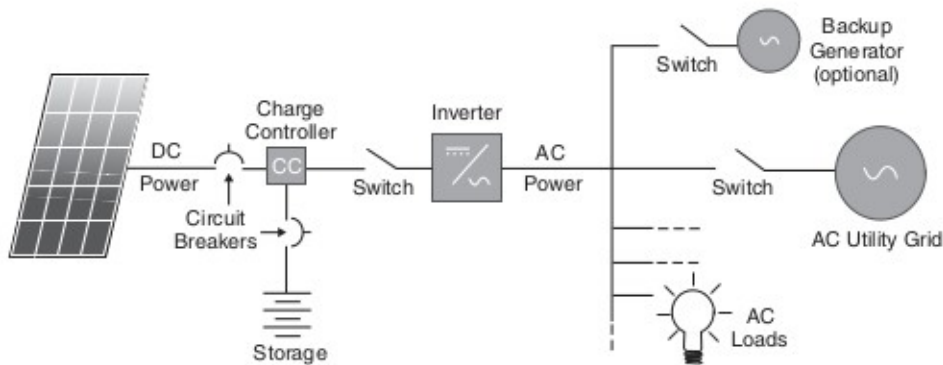
Σχήμα Α.1.6.4: Απλοποιημένο σχήμα συστήματος AC PV, εντός δικτύου [2].

Υβριδικά συστήματα

Ο όρος υβριδικό μπορεί να χρησιμοποιηθεί για το χαρακτηρισμό συστήματος PV το οποίο χρησιμοποιείται σε συνδυασμό με μια ή περισσότερες βοηθητικές πηγές ισχύος. Παραδοσιακά, αυτό σημαίνει δεύτερη πηγή ενέργειας όπως ο άνεμος ή οι υδροηλεκτρικοί στρόβιλοι, ωστόσο, πολλά σύγχρονα φωτοβολταϊκά συστήματα στρέφονται σε κατ' απαίτηση (on-demand) βοηθητικά συστήματα, όπως γεννήτριες ορυκτών καυσίμων ή ένα κεντρικό ηλεκτρικό δίκτυο. Τυπικό υβριδικό σύστημα παρουσιάζεται στο σχήμα Α.1.6.5.

Η αδιάκοπη παροχή ισχύος κατά τη διάρκεια της νύχτας ή ακόμα και υπό μη ευνοϊκές καιρικές συνθήκες εξασφαλίζεται με τη χρήση μπαταρίας. Τα υβριδικά συστήματα διαστασιολογούνται (sizing) έτσι ώστε ένα μεγάλο μέρος της καταναλισκόμενης ισχύος να είναι ηλιακής προέλευσης. Πλεόνασμα ενέργειας αποθηκεύεται σε μπαταρία. Μια γεννήτρια εκτάκτου ανάγκης (back-up generator), μπορεί να παρέχει ενέργεια στα συνδεδεμένα φορτία και συγχρόνως να φορτίζει τη μπαταρία, αυξάνοντας την αξιοπιστία και μειώνοντας την ανάγκη ενός μεγάλου συστήματος PV. Τα υβριδικά συστήματα συνδυάζουν τα καλύτερα χαρακτηριστικά κάθε ενεργειακής πηγής και παρέχουν ισότιμο ηλεκτρισμό με του ηλεκτρικού δικτύου.

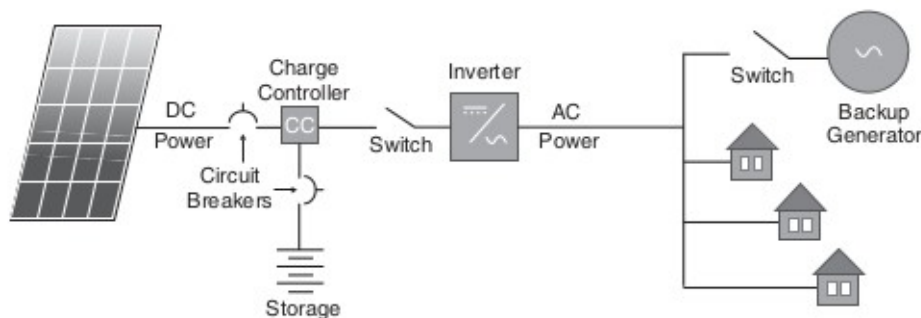
Όλα τα εξαρτήματα παραγωγής ηλεκτρισμού είναι συνδεδεμένα μέσω ξεχωριστών ελεγκτών φορτίων (charge controllers) και ανορθωτών (rectifiers) σε ένα δρομολογητή συνεχούς ρεύματος (DC bus), όπου η μπαταρία είναι συνδεδεμένη. Η μπαταρία παρέχει ισχύ στα συνδεδεμένα φορτία κατ' απαίτηση. Για την προστασία της, ο ελεγκτής φορτίων αποσυνδέει όλα τα φορτία, πριν η μπαταρία αποφορτιστεί πλήρως. Συγχρόνως, όταν η μπαταρία είναι χαμηλά, σε επίπεδο αποθηκευμένης ενέργειας, και τα φορτία απαιτούν περισσότερη ισχύ από ότι ο αναστροφέας μπορεί να παρέχει, η αναπληρωματική γεννήτρια μπορεί να ξεκινήσει αυτόματα, παρέχοντας απευθείας εναλλασσόμενο ρεύμα και φορτίζοντας, ταυτόχρονα, τη μπαταρία.



Σχήμα A.1.6.5: Απλοποιημένο σχήμα υβριδικού συστήματος pv [2].

Μικρο-πλέγματα

Φωτοβολταϊκά συστήματα εκτός δικτύου σταδιακά καθιερώνονται για την παροχή ισχύος σε απομακρυσμένα σπίτια ή μικρά ατομικά φορτία. Όταν πολλά απομακρυσμένα σπίτια σχηματίζουν οικισμό, προκύπτει η ανάγκη σχεδιασμού μεγάλης εγκατάστασης παραγωγής ισχύος, ικανή να παρέχει σε κάθε οικία το απαραίτητο εναλλασσόμενο ρεύμα (σχήμα A.1.6.6). Σε επίπεδο μεγάλης εγκατάστασης παραγωγής ισχύος το συνολικό κόστος εγκατάστασης και λειτουργίας είναι χαμηλότερο από το συνολικό κόστος του αθροίσματος των ατομικών εγκαταστάσεων. Τυπικά, τα μικρο-πλέγματα είναι της τάξης των 100 kW.



Σχήμα A.1.6.6: Απλοποιημένο σχήμα μικρο-πλέγματος [2].

Υποδειγματικά φωτοβολταϊκά συστήματα

Τα καλύτερα φωτοβολταϊκά συστήματα, πάντοτε επιδεικνύουν:

- 1.αποδοτικότητα
- 2.ασφάλεια
- 3.αξιοπιστία

Η απόδοση ενός συστήματος μπορεί να μετρηθεί από την ποσότητα ενέργειας που παράγεται και από το αν το σύστημα ανταποκρίνεται στους οικονομικούς στόχους που έχουν τεθεί. Για συστήματα εκτός δικτύου, ο βαθμός επιτυχίας είναι απλούστερος και στενά συνδεδεμένος με την αξιοπιστία. Συνοψίζεται στο αν παρέχει την απαιτούμενη ισχύ κάθε χρονική στιγμή, ανεξαρτήτως καιρού ή εποχής. Για συστήματα συνδεδεμένα στο δίκτυο, η παροχή της μέγιστης δυνατής ενέργειας στα τοπικά φορτία είναι συνήθως ο στόχος. Ο κατάλληλος σχεδιασμός, η εγκατάσταση και η συντήρηση αποτελούν κύριες παράμετρους για την επίτευξη ενός συστήματος υψηλής απόδοσης.

Η ασφάλεια, επίσης, παίζει ένα ουσιαστικό ρόλο στην επιτυχία των συστημάτων ρν. Είναι κρίσιμο το σύστημα να είναι συμβατό με το τοπικό ηλεκτρικό δίκτυο, να περιλαμβάνει εξαρτήματα κατασκευασμένα με τις εκάστοτε προδιαγραφές, να είναι σωστά γειωμένο και να μην αποτελεί κίνδυνο φωτιάς. Όσο η βιομηχανία των φωτοβολταϊκών επεκτείνεται, τόσο αυξάνονται και οι απαιτήσεις ασφαλείας.

Τέλος, τα σύγχρονα συστήματα ρν, προορίζονται για μακροχρόνια (άνω των 25 ετών), απρόσκοπτη λειτουργία πολύ κοντά στα αρχικά τους επίπεδα απόδοσης. Στην περίπτωση των συστημάτων εκτός δικτύου, η αξιοπιστία είναι κρίσιμότερη, καθώς το δίκτυο δε μπορεί να χρησιμοποιηθεί σαν ασφαλιστική δικλείδα. Μειωμένη αξιοπιστία επηρεάζει αρνητικά την απόδοση, μειώνοντας την ποσότητα του ηλεκτρικού ρεύματος που παράγεται και ταυτόχρονα αυξάνοντας το κόστος συντήρησης του συστήματος. Σε αντίθεση με τα περισσότερα ηλεκτρονικά εξαρτήματα, τα ρν υπόκεινται σε αντίξοες περιβαλλοντικές συνθήκες, όπου ακραίες θερμοκρασίες και υγρασία αυξάνουν τον ρυθμό αστοχίας και επιταχύνουν την υποβάθμιση των εξαρτημάτων.

A.1.7-ΤΑΞΙΝΟΜΗΣΗ ΦΩΤΟΒΟΛΤΑΪΚΩΝ ΣΥΣΤΗΜΑΤΩΝ [1], [2]

Υπάρχουν πολλοί τρόποι να ταξινομήσουμε/αξιολογήσουμε την ονομαστική ισχύ ενός φωτοβολταϊκού συστήματος (rating). Είναι κοινό οι αξιολογήσεις αυτές να παρατίθενται σε όρους ονομαστικής (ή αλλιώς μέγιστης) ισχύος συνεχούς ή εναλλασσόμενου ρεύματος. Συχνά δηλώνονται ως W_P , kW_P ή MW_P , συντομογραφίες αντιπροσωπευτικές της ισχύος DC, ή ως W_{AC} , kW_{AC} ή MW_{AC} , συντομογραφίες αντιπροσωπευτικές της ισχύος AC. Κανένας από αυτούς τους όρους δε στηρίζεται σε μετρήσεις υπό πραγματικές συνθήκες. Βασίζονται, απλά, στις ονομαστικές τιμές και προβλέπουν την καλύτερη δυνατή επίδοση.

Θα πρέπει να σημειωθεί ότι ο χαρακτηρισμός της ονομαστικής ισχύος μιας φωτοβολταϊκής μονάδας βασίζεται σε συνθήκες αναφοράς, γνωστές σαν STC (standard test conditions), οι οποίες είναι:

-ένταση ηλιακής ακτινοβολίας $G_{Tref}=1,000 \text{ W/m}^2$

-θερμοκρασία κυψέλης $T_{cell}=25 \text{ }^\circ\text{C}$

-μάζα αέρα $AM=1.5$ (air mass spectrum)

Πρόκειται για ένα συνδυασμό συνθηκών, εξαιρετικά σπάνιο να συμβεί στην πραγματικότητα.

Στην περίπτωση της αξιολόγησης με όρους DC, η ονομαστική ισχύς είναι ίση με το άθροισμα της ονομαστικής ισχύος κάθε μονάδας ενώ στην περίπτωση της αξιολόγησης με όρους AC, η ονομαστική ισχύς είναι ίση με το άθροισμα των ονομαστικών τιμών των αναστροφών. Οι χαρακτηρισμοί ισχύος με όρους AC, DC είναι εξίσου ανακριβείς όροι, προσφέρουν, ωστόσο, μια γενική εικόνα του μεγέθους του συστήματος. Οι ονομαστικές τιμές της ισχύος συστημάτων με όρους AC είναι εξ' ορισμού χαμηλότερες απ' ότι με όρους DC, λόγω αναπόφευκτων απωλειών κατά τη μετατροπή του συνεχούς ρεύματος σε εναλλασσόμενο. Πολλά συστήματα έχουν ονομαστικές τιμές ισχύος σε όρους DC 10-30 % μεγαλύτερες από τις αντίστοιχες με όρους AC. Πρόκειται για ένα εύλογο επίπεδο αύξησης, αν λάβουμε υπόψη τις απώλειες λόγω αντίστασης καλωδίων, θερμοκρασίας, σκόνης, το βαθμό απόδοσης αναστροφέα και άλλες αναπόφευκτες αιτίες. Αυτή η σημαντική διαφορά ανάμεσα στους δύο τρόπους αξιολόγησης/ταξινόμησης προκαλεί σύγχυση και παρερμηνεία όταν κόστος και ωφέλη είναι υπό εξέταση χωρίς να διευκρινιστεί αν αντιστοιχούν σε όρους συνεχούς ή εναλλασσόμενου ρεύματος.

Έχουν γίνει προσπάθειες για τον καθορισμό της ονομαστικής ισχύος συστημάτων AC με άλλους τρόπους. Μερικές μέθοδοι είναι αναλυτικές, κάποιες, όμως, βασίζονται σε δεδομένα μετρήσεων. Τα αναλυτικά χαρακτηριστικά AC τείνουν να ξεκινούν από τις ονομαστικές τιμές DC και περνώντας από μια σειρά συντελεστών απωλειών, καταλήγουν μειωμένες, αλλά πιο κοντά στην πραγματικότητα. Συχνά, οι επιλεγμένες συνθήκες είναι οι ονομαστικές STC, αλλά όχι πάντα. Εναλλακτική αποτελούν οι SRC (standard reporting conditions), οι οποίες, απλά, χρησιμοποιούν ρεαλιστικότερη θερμοκρασία κυψέλης, $45 \text{ }^\circ\text{C}$. Ανάλογα με τη χρησιμοποιούμενη τεχνολογία, οι

τιμές που αντιστοιχούν σε συνθήκες SRC είναι χαμηλότερες κατά 5-10 % των αντίστοιχων σε STC.

A.2-ΘΕΩΡΙΑ ΥΛΙΚΩΝ ΑΛΛΑΓΗΣ ΦΑΣΗΣ

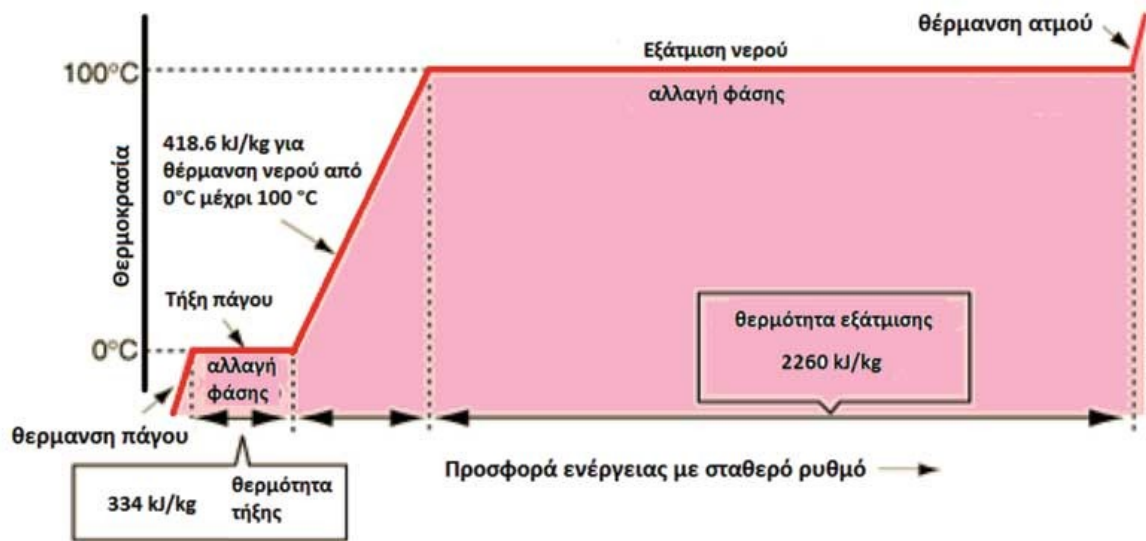
A.2.1-ΓΕΝΙΚΑ [4], [5], [6]

Τα υλικά αλλαγής φάσης είναι ουσίες με υψηλή λανθάνουσα θερμότητα τήξης (latent heat of fusion). Με άλλα λόγια, πρόκειται για ουσίες που απορροφούν ή απελευθερώνουν μεγάλα ποσά θερμότητας κατά την υγροποίηση ή τη στερεοποίηση τους υπό ορισμένη θερμοκρασία. Η μοναδική αυτή ικανότητα των PCM (phase change materials) τα κατηγοριοποιεί ως μονάδες αποθήκευσης λανθάνουσας θερμότητας (latent heat storage units).

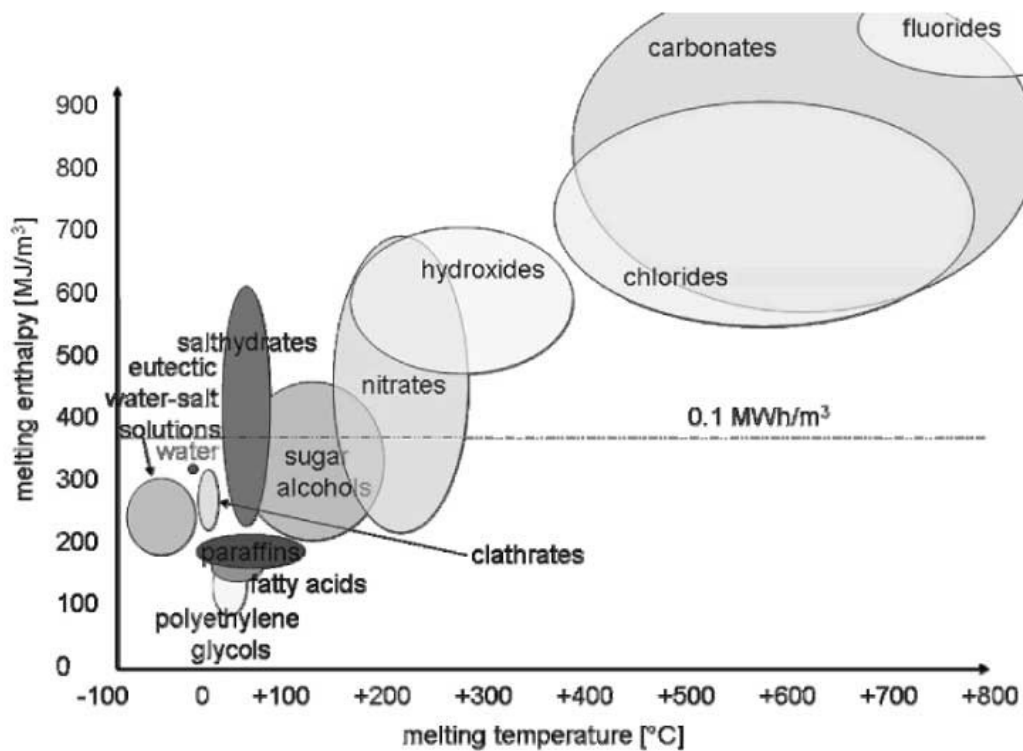
Η αποθήκευση λανθάνουσας θερμότητας μπορεί να επιτευχθεί μέσω αλλαγής φάσης μεταξύ στερεής και στερεής (solid-solid), στερεής και υγρής (solid-liquid), στερεής και αέριας (solid-gas) και υγρής και αέριας (liquid-gas). Ωστόσο, η μόνη αλλαγή φάσης των PCM που χρησιμοποιείται στην πράξη είναι μεταξύ στερεής και υγρής φάσης. Οι αλλαγές μεταξύ υγρής-αέριας φάσης δεν είναι πρακτικές λόγω των μεγάλων απαιτούμενων όγκων αποθήκευσης και των υψηλών πιέσεων κατά την αέρια φάση τους. Οι αλλαγές υγρής-αέριας φάσης παρουσιάζουν υψηλότερη λανθάνουσα θερμότητα αλλαγής φάσης από τις αλλαγές στερεής-υγρής. Οι αλλαγές στερεής-στερεής φάσης είναι τυπικά πολύ αργές και παρουσιάζουν χαμηλή θερμότητα αλλαγής φάσης.

Ο γνωστότερος εκπρόσωπος της κατηγορίας των υλικών αλλαγής φάσης είναι το νερό. Στους 0°C, προκειμένου να αλλάξει φάση, από στερεό (πάγος) σε υγρό, χρειάζεται να απορροφήσει ενέργεια ίση με 334 kJ/kg. Η ενέργεια αυτή ονομάζεται, όπως προαναφέρθηκε, λανθάνουσα θερμότητα τήξης γιατί δεν προκαλεί αύξηση της θερμοκρασίας του (δηλαδή, δεν είναι αντιληπτή με τις αισθήσεις) αλλά αποδίδεται στη μεταβολή των δεσμών και της κινητικής ενέργειας των μορίων του. Ομοίως, για την εξάτμιση του νερού, στους 100°C, δηλαδή για τη μετάβαση από την υγρή στην αέρια φάση, θα πρέπει να απορροφήσει, ως λανθάνουσα θερμότητα εξάτμισης, ενέργεια ίση με 2260 kJ/kg. Αντίστροφα, κατά την ψύξη του νερού, στις θερμοκρασίες αλλαγής φάσης, οι κατά περίπτωση λανθάνουσες θερμότητες απελευθερώνονται. Στο παρακάτω διάγραμμα A.2.1.1 φαίνεται ότι με συνεχή και σταθερό ρυθμό θέρμανσης μιας ποσότητας νερού επιτυγχάνουμε σταθερή αύξηση της θερμοκρασίας του, εκτός από τις θερμοκρασίες των 0 και 100°C, όπου η προσφορά θερμικής ενέργειας (σε ποσότητες 334 και 2260 kJ/kg αντίστοιχα) δεν προκαλεί αύξηση της θερμοκρασίας αλλά μεταβολή φάσης. Το ίδιο διάγραμμα ισχύει και κατά την αντίστροφη διαδικασία, της ψύξης, όπου πλέον κατά την αλλαγή φάσης (από ατμό σε υγρό και από υγρό σε πάγο) η αποθηκευμένη λανθάνουσα θερμότητα επιστρέφεται στο περιβάλλον.

Επίσης, ακολουθεί το διάγραμμα A.2.1.2, στο οποίο παρουσιάζονται οι διάφορες κατηγορίες PCM με τα αντίστοιχα λειτουργικά τους όρια (λανθάνουσα θερμότητα τήξης, θερμοκρασία τήξης). Εντός της ζώνης θερμικής άνεσης (20-30 °C) κάποια υλικά αποδεικνύονται πολύ αποδοτικά, αποθηκεύοντας 5-14 φορές περισσότερη θερμότητα ανά μονάδα όγκου σε σχέση με κοινά δομικά υλικά.



Διάγραμμα Α.2.1.1: Γραφική απεικόνιση της θερμότητας που απορροφάται/απελευθερώνεται κατά την αλλαγή φάσης του νερού [6].



Διάγραμμα Α.2.1.2: Γραφική απεικόνιση ομάδων PCM. Για κάθε ομάδα διακρίνονται τα θερμοκρασιακά όρια λειτουργίας και η λανθάνουσα θερμότητα τήξης [6].

A.2.2-ΚΑΤΗΓΟΡΙΕΣ ΥΛΙΚΩΝ ΑΛΛΑΓΗΣ ΦΑΣΗΣ [4], [5], [7]

Τα υλικά αλλαγής φάσης μπορούν να ταξινομηθούν στις παρακάτω κατηγορίες:

1. Παραφίνες (paraffins)
2. Οργανικά μη παραφινώδη (non-paraffin organics)
3. Ενυδρα άλατα (salt hydrates)
4. Μεταλλικά (metallics)
5. Εύτηκτα άλατα (fused salt eutectics)
6. Διάφορα (miscellaneous)
7. Στερεής μετάβασης (solid-solid)

Παραφίνες

Οι παραφίνες του τύπου C_nH_{2n+2} αποτελούν οικογένεια κορεσμένων υδρογονανθράκων με παρόμοιες ιδιότητες. Όλες της σειράς κάτω από πεντάνιο (pentane, C_5H_{12}) είναι αέρια σε κανονικές θερμοκρασίες. Αυτές μεταξύ C_5 και C_{12} είναι υγρά και οι υπόλοιπες κηρώδη στερεά (waxy solids).

Παραφίνες κανονικής (normal chain) ή ευθείας αλυσίδας (straight chain) συνήθως τήκονται σε υψηλότερες θερμοκρασίες από αυτές των διακλαδισμένων αλυσίδων (branched chain). Οι χημικές ιδιότητες καθορίζονται από τη δομή τους. Οι παραφίνες ευθείας αλυσίδας και οι συμμετρικής διακλαδισμένης αλυσίδας είναι οι σταθερότερες.

Οι παραφίνες με ζυγό αριθμό ατόμων άνθρακα χρησιμοποιούνται περισσότερο από αυτές με περιττό αριθμό λόγω διαθεσιμότητας, χαμηλότερου κόστους και υψηλότερης λανθάνουσας θερμότητας τήξης. Το σημείο τήξης τείνει να αυξάνεται με την αύξηση του μοριακού βάρους.

Γενικά, οι παραφίνες είναι τα πρώτα υλικά που λαμβάνονται υπόψη σε θέματα θερμικού ελέγχου. Είναι υλικά αξιόπιστα, ασφαλή, μη διαβρωτικά και προβλέψιμα. Από την άλλη είναι αρκετά εύφλεκτα και παρουσιάζουν εξαιρετικά χαμηλή θερμική αγωγιμότητα, γεγονός που περιορίζει την αποδοτικότητα τους. Μεταλλικά υλικά πληρώσης (metallic fillers) προτείνονται για την αύξηση της θερμικής αγωγιμότητας. Αλουμινένια πλέγματα μορφής κερήθρας (aluminum honeycomb) βελτιώνουν την απόδοση του συστήματος.

Οι ιδιότητες των παραφινών έχουν ως εξής:

- Υψηλή λανθάνουσα θερμότητα τήξης ανά μονάδα μάζας.
- Μεγάλος εύρος σημείου τήξης ($268-339\text{ }^\circ\text{K}/23-151\text{ }^\circ\text{F}/-5-66\text{ }^\circ\text{C}$).
- Εύφλεκτα υλικά
- Δεν είναι τοξικά.
- Δεν διαβρώνονται .
- Χημική αδράνεια και σταθερότητα κάτω από τους $773\text{ }^\circ\text{K}$ ($932\text{ }^\circ\text{F}/500\text{ }^\circ\text{C}$).
- Αμελητέο φαινόμενο υπέρψυξης (supercooling).
- Μικρή μεταβολή όγκου κατά την τήξη.
- Χαμηλή πίεση ατμού στην τήξη.
- Εύρος πυκνότητας $700-770\text{ kg/m}^3$ ($43.7-48.1\text{ lb/ft}^3$)
- Χαμηλή θερμική αγωγιμότητα (βελτιώνεται με χρήση fillers)
- Εμπορική διαθεσιμότητα σε λογικό κόστος.
- Υψηλή ικανότητα διαβρεκτικότητας (wetting/wettability, ικανότητα του υγρού να "κολλά" στο στερεό που το περιέχει).
- Προβλεψιμότητα και αξιοπιστία.

Τέλος, παρατίθενται τέσσερις κύριοι εκπρόσωποι της οικογένειας των παραφινών:

Tetradecane, $C_{14}H_{30}$, $\Delta H_f=228\text{ joule/kg}$ (98 Btu/lb)

Hexadecane, $C_{16}H_{34}$, $\Delta H_f=237\text{ joule/kg}$ (102 Btu/lb)

Octadecane, $C_{18}H_{38}$, $\Delta H_f=244\text{ joule/kg}$ (105 Btu/lb)

Eicosane, $C_{20}H_{42}$, $\Delta H_f=244$ joule/kg (105 Btu/lb)

Μη παραφινώδη οργανικά υλικά

Τα μη παραφινώδη οργανικά υλικά αποτελούν την πιο πολυάριθμη κατηγορία PCM και οι ιδιότητες τους ποικίλουν, σε αντίθεση με τις παραφίνες που παρουσιάζουν παρόμοιες ιδιότητες.

Τα οργανικά υλικά δε θα πρέπει να εκτίθενται σε υψηλές θερμοκρασίες, φωτιά ή ισχυρούς οξειδωτικούς παράγοντες. Σχεδόν όλα τα οργανικά υλικά είναι εύφλεκτα. Οι υψηλές θερμοκρασίες προκαλούν αποσύνθεση, η οποία συνοδεύεται από την παραγωγή τοξικών αερίων. Είναι τύπος υλικού που απαιτεί προσεκτικούς χειρισμούς.

Σε έρευνα που έγινε, μεταξύ 500 οργανικών υλικών, τα 113 έλιωσαν σε θερμοκρασίες μεταξύ 303 και 473 °K (86-392 °F/30-200 °C). Η λανθάνουσα θερμότητα κυμάνθηκε μεταξύ $4.2 \cdot 10^4$ και $2.5 \cdot 10^5$ joules/kg (20-100 Btu/lb, 10-60 cal/gm).

Η θερμοκρασιακή αγωγιμότητα των οργανικών υλικών ($\alpha=k/\rho C_p$ [m^2/s]) κυμαίνεται από πολύ χαμηλές τιμές μέχρι τιμές συγκρίσιμες με του νερού ή υψηλότερες. Η χρήση υλικών fillers, γενικά, είναι κάτι που προτείνεται. Η κατηγορία των υλικών αυτών είναι υπερβολικά μεγάλη για την απαρίθμηση ιδιοτήτων που αντιστοιχούν σε όλα. Σε γενικές γραμμές, ιδιότητες που χαρακτηρίζουν τα οργανικά μη παραφινώδη υλικά είναι:

- Τα περισσότερα είναι εύφλεκτα.
- Κάποια είναι τοξικά.
- Μερικά από τα οξέα μακράς-αλυσίδας (long-chain acids) επιδεικνύουν πολυμορφισμό (polymorphism, δύο ή περισσότερες κρυσταλλικές μορφές).
- Το σημείο λάμψης (flash point, η χαμηλότερη θερμοκρασία ενός πτητικού υλικού στην οποία εξατμίζεται διαμορφώνοντας εύφλεκτο μίγμα με τον αέρα) πολλών είναι χαμηλό.
- Ακαθαρσίες επηρεάζουν σε μεγάλο βαθμό το σημείο τήξης.
- Υλικά fillers προτείνονται για τη βελτίωση της θερμικής αγωγιμότητας.
- Αυξημένες θερμοκρασίες προκαλούν την αποσύνθεση τους.
- Βολικές θερμοκρασίες τήξεως.
- Μεταβολές στερεής-στερεής φάσης είναι κοινές (solid-solid transitions).
- Ελκυστικές τιμές λανθάνουσας θερμότητας αλλαγής φάσης.

Επίσης, παρατίθενται οι κύριοι εκπρόσωποι της κατηγορίας:

Polyethynel glucol 600 $\Delta H_f=146 \cdot 10^3$ joules/kg (63 Btu/lb).

Acetic acid, $\Delta H_f=187 \cdot 10^3$ joules/kg (80.3 Btu/lb).

Tristearin, $\Delta H_f=191 \cdot 10^3$ joules/kg (82.1 Btu/lb).

Myristic acid, $\Delta H_f=199 \cdot 10^3$ joules/kg (85.5 Btu/lb).

Stearic acid, $\Delta H_f=199 \cdot 10^3$ joules/kg (85.5 Btu/lb).

Elaidic acid, $\Delta H_f=218 \cdot 10^3$ joules/kg (93.7 Btu/lb).

Acetamide, $\Delta H_f=241 \cdot 10^3$ joules/kg (104 Btu/lb).

Methyl fumarate, $\Delta H_f=242 \cdot 10^3$ joules/kg (104 Btu/lb).

Ένυδρα άλατα

Τα ένυδρα άλατα (salt hydrates) μπορούν να θεωρηθούν ως ενώσεις ανόργανων αλάτων (inorganic salts) και νερού, με συγκεκριμένο αριθμό moles νερού, σχηματίζοντας τυπικά κρυσταλλικά στερεά του γενικού τύπου $AB \cdot nH_2O$.

Τα ένυδρα άλατα με κάποιες εξαιρέσεις, έχουν, γενικά ομοιόμορφα σημεία τήξης. Ένυδρο άλας του τύπου $AB \cdot nH_2O$, συνήθως, λιώνει σχηματίζοντας νέα στερεή φάση, διαφορετική από την αρχική, περιέχοντας λιγότερα moles νερού. Για παράδειγμα:

$AB \cdot nH_2O \rightarrow AB \cdot mH_2O + (n-m)$ moles H_2O , ή στην άνυδρη μορφή:

$AB \cdot nH_2O \rightarrow AB + nH_2O$.

Στο μεταβατικό σημείο ένυδρου άλατος, το οποίο έχει ανομοιόμορφη θερμοκρασία τήξης, τρεις φάσεις βρίσκονται σε ισορροπία: δύο στερεές φάσεις και μία υγρή κορεσμένου διαλύματος άλατος. Λόγω της ανολοκλήρωτης διάλυσης, τα κατώτερα ένυδρα (lower hydrate) "ηρεμούν" στον πυθμένα του δοχείου και, έτσι, δε διατίθενται για επανασυνδυασμό τους με μόρια νερού, όταν η θερμοκρασία μειωθεί. Η ανομοιόμορφη τήξη (incongruent melting) οδηγεί σε έλλειψη αναστρεψιμότητας κύκλων τήξης/πήξης. Τα ένυδρα άλατα με ομοιόμορφο σημείο τήξης συμπεριφέρονται όπως τα καθαρά υλικά. Οι τρεις κύριοι εκπρόσωποι, που ακολουθούν, παρουσιάζουν εξαιρετική συμπεριφορά στερεοποίησης.

Χαρακτηριστικά ένυδρων αλάτων:

- Υψηλή λανθάνουσα θερμότητα αλλαγής φάσης, ανά μονάδα μάζας και ανά μονάδα όγκου.
- Χαμηλές ογκομετρικές αλλαγές κατά το λιώσιμο.
- Η πλειοψηφία των δοκιμασμένων ένυδρων αλάτων παρουσιάζει ανομοιόμορφες θερμοκρασίες τήξης και κατά συνέπεια δυσκολία αναστρεψιμότητας (εξαιρούνται: $\text{LiNO}_3 \cdot 3\text{H}_2\text{O}$, $\text{Ba}(\text{OH})_2 \cdot 8\text{H}_2\text{O}$, $\text{Na}_2\text{HPO}_4 \cdot 12\text{H}_2\text{O}$).
- Σχετικά υψηλή θερμική αγωγιμότητα μη μεταλλικών υλικών.
- Φαινόμενο υπέρψυξης (supercooling), περιορισμένης έκτασης, το οποίο περιορίζεται ακόμα περισσότερο με παράγοντες πυρηνοποίησης (nucleating agents).
- Διαβρωτικά.

Κύριοι εκπρόσωποι αυτής της κατηγορίας:

Sodium hydrogen phosphate dodecahydrate $\Delta H_f = 265 \cdot 10^3$ joules/kg (114 Btu/lb).

Lithium nitrate trihydratem $\Delta H_f = 296 \cdot 10^3$ joules/kg (128 Btu/lb).

Barium hydroxide octahydrate, $\Delta H_f = 302 \cdot 10^3$ joules/kg (130 Btu/lb).

Μεταλλικά

Αυτή η κατηγορία περιλαμβάνει μέταλλα χαμηλού σημείου τήξης (low melting metals) και εύτηκτα μέταλλα (metal eutectics). Πρόκειται για κατηγορία που δεν έχει διερευνηθεί διεξοδικά, λόγω του παράγοντα βάρους. Ωστόσο, τα υλικά αυτά παρουσιάζουν υψηλή λανθάνουσα θερμότητα τήξης ανά μονάδα όγκου. Επίσης, παρουσιάζουν υψηλές θερμικές αγωγιμότητες και έτσι, υλικά fillers, που προσθέτουν βάρος, δεν απαιτούνται. Κράματα χαμηλού σημείου τήξεως, συνήθως, περιλαμβάνουν μέταλλα χαμηλού σημείου τήξης (Bi, In, Cd, Pd, Sn). Σημαντική διαφορά μεταξύ μεταλλικών υλικών αλλαγής φάσης και υπολοίπων είναι η υψηλή θερμική αγωγιμότητα και οι μικροί αριθμοί Prandtl, της τάξεως 0.001-0.1.

Για μερικά από αυτά τα υλικά η διάβρωση αποτελεί σημαντικό θέμα. Για παράδειγμα, το Γάλλιο (Gallium) είναι ιδιαίτερα αντιδραστικό με το αλουμίνιο, ένα κοινό κατασκευαστικό υλικό. Η χρήση αναστολέων (inhibitors) μειώνει, ως ένα βαθμό, τα προβλήματα διάβρωσης με κάποια υλικά.

Τα χαρακτηριστικά των μεταλλικών υλικών είναι:

- Χαμηλή λανθάνουσα θερμότητα τήξης ανά μονάδα μάζας.
- Υψηλή λανθάνουσα θερμότητα τήξης ανά μονάδα όγκου.
- Υψηλή θερμική αγωγιμότητα (υλικά fillers κρίνονται μη απαραίτητα).
- Χαμηλή ειδική θερμοχωρητικότητα.
- Σχετικά χαμηλή πίεση ατμού.
- Χαμηλή ογκομετρική διαστολή κατά την τήξη.
- Υψηλή θερμική σταθερότητα.
- Ελάχιστη επικίνδυνη/επιζήμια συμπεριφορά.

Κύριοι εκπρόσωποι της κατηγορίας είναι:

Cerrobend eutectic $\Delta H_f = 32.6 \cdot 10^3$ joules/kg (14 Btu/lb).

Gallium $\Delta H_f = 80.3 \cdot 10^3$ joules/kg (34.4 Btu/lb).

Εύτηκτα άλατα

Ένα εύτηκτο άλας (fused salt eutectic) αποτελείται από δύο ή περισσότερα ανόργανα άλατα, τα οποία διαμορφώνουν ένα εύτηκτο μίγμα. Ένα εύτηκτο σύστημα έχει σαφές σημείο τήξεως, όπως μία καθαρή ουσία. Τα δεδομένα γι' αυτήν την κατηγορία είναι περιορισμένα, είτε λόγω έλλειψης πληροφοριών είτε λόγω μειωμένης απόδοσης, και, κατά συνέπεια, δεν υπάρχουν κύριοι εκπρόσωποι.

Μερικά περιέχουν νιτρικά άλατα (nitrates), γιατί έχουν χαμηλά σημεία τήξης σε αντίθεση με τα αντιστοιχα αλογονίδια (halides: chlorides, fluorides, bromides, iodides). Ενώ τα δεδομένα για τα εύτηκτα υλικά που περιέχουν νιτρικά άλατα δείχνουν ενθαρρυντικά, η θέρμανση τους δεν ενδείκνυται. Μπορεί να εκραγούν με την έκθεσή τους σε φλόγα ή φωτιά, με βίαιη μεταχείριση ή ακόμα και αυθόρμητα. Το νιτρικό αμμώνιο (ammonium nitrate) μπορεί να εκραγεί από μόνο του κάτω από αδιευκρίνιστες συνθήκες.

Πέρα από τα νιτρικά άλατα, υπάρχουν και πολλές άλλες πιθανότητες. Κάποια υλικά προσφέρουν ευρύ φάσμα θερμοκρασιών τήξης, με μεταβολή της αναλογίας των συστατικών τους. Για παράδειγμα, μια σειρά έχει ως συστατικάθειικό μαγνήσιο (magnesium sulfate),θειικό νάτριο (sodium sulfate) και νερό, με ποικίλα σημεία τήξης, σύμφωνα με την αναλογία των συστατικών τους. Το χλωριούχο αργίλιο (aluminum chloride, $AlCl_3$) χρησιμοποιείται ως συστατικό εύτηκτων υλικών επειδή μειώνει τη θερμοκρασία τήξης άλλων χλωριούχων (chlorides). Σχηματίζει δυαδικές και τριαδικές ενώσεις (binary and ternary compounds) και εύτηκτα μίγματα με χλωρίδια λιθίου (lithium chlorides, $LiCl$) και χλωριούχο νάτριο (sodium chloride, $NaCl$). Όταν θερμαίνεται, το χλωριούχο αργίλιο σχεδόν διπλασιάζει τον όγκο του κατά την τήξη, αν και, στα εύτηκτα μίγματα η ογκομετρική διαστολή περιορίζεται σημαντικά. Κάποια από τα εύτηκτα $AlCl_3$, μπορεί να έχουν επιθυμητές ιδιότητες σαν υλικά θερμικής αποθήκευσης.

Συστήματα που περιέχουν λίθιο (lithium), γενικά, παράγουν εύτηκτα χαμηλότερης θερμοκρασίας τήξης από τα ανάλογα συστήματα αλκαλίων και αλκαλικών γαιών (alkali-earth). Επίσης χαρακτηρίζονται από χαμηλότερο ιξώδες (viscosity) και υψηλή ηλεκτρική αγωγιμότητα.

Χαρακτηριστικά των εύτηκτων αλάτων είναι:

- Σε ορισμένες περιπτώσεις η αναλογία των συστατικών μπορεί να μεταβληθεί προσφέροντας, έτσι, τη δυνατότητα επιλογή τιμής σημείου και λανθάνουσας θερμότητας τήξης.
- Προσφέρουν ποικιλία πιθανών ιδιοτήτων.
- Γενικά, η λανθάνουσα θερμότητα τήξης είναι υψηλή.
- Η παρουσία υγρασίας κατά την παρασκευή τους, επηρεάζει το σημείο τήξης.
- Είναι διαβρωτικά υλικά.

Υλικά αλλαγής φάσης στερεής μετάβασης

Η πιθανότητα για υψηλές ενθαλπίες, χαμηλούς συντελεστές θερμικής διαστολής και ελαχιστοποίηση του φαινομένου της υπέρψυξης (supercooling), οδήγησαν στις μεταβάσεις στερεών φάσεων (solid state transition). Πολλά οργανικά υλικά περνούν από μετάβαση στερεών φάσεων (solid-solid transition) σε θερμοκρασία μετάβασης (transition temperature) χαμηλότερη της θερμοκρασίας τήξης. Οι πλαστικοί κρύσταλλοι (plastic crystals) αποτελούν μία σειρά οργανικών υλικών, που έχουν υψηλές μεταβατικές ενθαλπίες (transitional enthalpies). Οι μεταβατικές ενθαλπίες τους κυμαίνονται μεταξύ 184-301 Joules/kg (79-130 Btu/lb, 44-72 cal/gm) και οι μεταβατικές θερμοκρασίες μεταξύ 341-457 K (154-363 °F, 68-184 °C).

Χαρακτηριστικά πλαστικών κρυστάλλων (plastic crystals):

- Είναι μαλακά, κηρώδη στερεά.
- Ασυνήθιστα υψηλές πιέσεις ατμών για στερεά.
- Ογκομετρικές αλλαγές της τάξης 10-50 %.
- Η υπέρψυξη δεν αποτελεί εγγενές πρόβλημα των μεταβολών στερεών φάσεων.

-Οι θερμοκρασίες μετάβασης είναι ιδιαίτερα υψηλές.

Υγροσκοπικά

Πολλά οικοδομικά υλικά είναι υγροσκοπικά, μπορούν δηλαδή να απορροφούν υγρασία όταν η ατμόσφαιρα έχει αυξημένη υγρασία και να αποβάλλουν νερό όταν η ατμόσφαιρα είναι ξηρή. Συνήθως έχουν ινώδη υφή και η απορρόφηση υγρασίας μεταβάλλει τις φυσικές και μηχανικές τους ιδιότητες. Η διαδικασία έχει ως εξής:

-Συμπύκνωση (αέριο σε υγρό), μείωση ενθαλπίας ($\Delta H < 0$, εξωθερμική διεργασία), απελευθέρωση θερμότητας.

-Εξάτμιση (υγρό σε αέριο), αύξηση ενθαλπίας ($\Delta H > 0$, ενδοθερμική διεργασία), απορρόφηση θερμότητας.

Αν και η διαδικασία αυτή αφορά μικρά ποσά ενέργειας, μεγάλες επιφάνειες επιτρέπουν σημαντική ψύξη ή θέρμανση στα κτίρια ($1-2\text{ }^{\circ}\text{C}$).

Ανάμεσα στα υγροσκοπικά υλικά περιλαμβάνονται το βαμβάκι, το μέλι, πολλά άλατα, το καύσιμο ντίζελ, η αιθανόλη, η μεθανόλη και το θειϊκό οξύ. Από τα πιο υγροσκοπικά υλικά είναι ο χλωριούχος ψευδάργυρος, το χλωριούχο ασβέστιο, τα υδροξείδια καλίου και νατρίου σε τέτοιο βαθμό που διαλύονται με χαρακτηριστική ευκολία στο νερό που απορροφούν.

Διάφορα

Η κατηγορία αυτή περιλαμβάνει υλικά αλλαγής φάσης που δεν ανήκουν ξεκάθαρα σε κάποια από τις παραπάνω κατηγορίες. Η σειρά Transit Heet πρόκειται για ιδιοσκεύασμα άγνωστης σύστασης, πιθανότατα εύτηκτο μίγμα ένυδρων αλάτων. Το Dowtherm A είναι μίγμα διφενυλαιθέρα-οξειδίου διφενυλαιθέρα (diphenyl-diphenyl oxide), το οποίο χρησιμοποιείται σε ρευστό μεταφοράς θερμότητας. Διακρίνεται για την έντονη οσμή του, τη χαμηλή λανθάνουσα θερμότητα τήξης ($97.9 \cdot 10^3\text{ joules/kg}$ - 42.2 Btu/lb) και το βολικό θερμοκρασιακό εύρος τήξης του.

Σημαντικό μέλος αποτελεί το νερό. Δε θα πρέπει να παραμερίζεται επειδή είναι κοινό υλικό. Έχει την υψηλότερη θερμοχωρητικότητα, ανάμεσα στις συνηθισμένες ουσίες. Η λανθάνουσα θερμότητα αλλαγής φάσης ανά μονάδα μάζας ή όγκου είναι εξαιρετικά υψηλή. Είναι απαλλαγμένο από πολλά προβλήματα, καθώς είναι σταθερό, μη τοξικό, μη εύφλεκτο και η θερμική αγωγιμότητά του είναι σχετικά υψηλή, για μη μεταλλικό υλικό. Το χλωριούχο αργίλιο είναι ανόργανο άλας που χρησιμοποιείται εκτενώς στην τεχνολογία της οργανικής χημείας και οι ιδιότητες του έχουν μελετηθεί λεπτομερώς. Αυτή η υψηλά υγροσκοπική ουσία (ικανότητα απορρόφησης υγρασίας από τον αέρα) παρουσιάζει ασυνήθιστες ιδιότητες. Διπλασιάζει τον όγκο της κατά την τήξη και σχηματίζει εύτηκτα μίγματα με άλλα χλωρίδια (chloride) χαμηλότερης θερμοκρασίας τήξης.

Κυριότεροι εκπρόσωποι της κατηγορίας:

-Νερό $\Delta H_f = 333 \cdot 10^3\text{ joules/kg}$ (143 Btu/lb).

-Transit heet $\Delta H_f = 230 \cdot 10^3$ - $301 \cdot 10^3\text{ joules/kg}$ (99 - 129 Btu/lb).

A.2.3-ΚΡΙΤΗΡΙΑ ΕΠΙΛΟΓΗΣ ΥΛΙΚΩΝ ΑΛΛΑΓΗΣ ΦΑΣΗΣ [4], [5]

Ο θερμικός έλεγχος με τη χρήση υλικών υψηλής λανθάνουσας θερμότητας τήξης είναι μια σχετικά νέα τεχνολογία. Η επιλογή ενός υλικού αλλαγής φάσης, πέραν της θερμοκρασίας τήξης και της λανθάνουσας θερμότητας τήξης, περιλαμβάνει κι άλλα κριτήρια, ιδιότητες και παραμέτρους. Για παράδειγμα πολλά οργανικά νιτρικά (organic nitrates), αν και έχουν υψηλή λανθάνουσα θερμότητα και βολικές θερμοκρασίες τήξης, κρίνονται ακατάλληλα λόγω τοξικότητας. Η γλυκερίνη έχει σημείο πήξης τους $290\text{ }^{\circ}\text{K}$ (glycerine, $17\text{ }^{\circ}\text{C}$, $63\text{ }^{\circ}\text{F}$), ωστόσο πολλοί ερευνητές έχουν συναντήσει δυσκολίες στην στερεοποίηση του υλικού. Ένα PCM θα πρέπει να έχει αξιόπιστη συμπεριφορά στερεοποίησης.

Οι ογκομετρικές αλλαγές είναι εξίσου σημαντικές. Τα περισσότερα υλικά συρρικνώνονται καθώς ψύχονται και η αλλαγή όγκου ποικίλει ανάλογα με το PCM. Νερό, γάλλιο και κράματα γαλλίου διαστέλλονται με την ψύξη τους. Κάποια υλικά παρουσιάζουν υψηλές πιέσεις ατμών κοντά στο σημείο πήξης τους, ασκώντας υψηλές τάσεις στα δοχεία που τα περιέχουν.

Ένας αριθμός ένυδρων αλάτων (salt hydrates) λιώνουν, σχηματίζοντας ένυδρη ουσία (lower hydrate) και νερό. Σε αυτές τις περιπτώσεις δε συμβαίνει αναστρέψιμος ψυκτικός κύκλος. Ωστόσο, κάποια από τα ένυδρα άλατα λιώνουν ομοιόμορφα, ψύχονται αναστρέψιμα και κατατάσσονται ανάμεσα στα καλύτερα PCM.

Δεν υπάρχει τέλειο υλικό αλλαγής φάσης, καθώς κανένα υλικό δεν παρουσιάζει όλες τις επιθυμητές ιδιότητες στον ιδανικό βαθμό. Ένα PCM αναμένεται να παρέχει ισοθερμικό έλεγχο για συγκεκριμένο χρονικό διάστημα και για συγκεκριμένη εφαρμογή. Έτσι, γίνονται αναγκαστικά θυσιές κατά την επιλογή του κατάλληλου υλικού.

Συχνά, ανεπιθύμητες ιδιότητες PCM διορθώνονται. Για παράδειγμα, μεταλλικά "γεμίσματα" (fillers) με τη μορφή πλέγματος μπορούν να χρησιμοποιηθούν για να αυξήσουν την χαμηλή θερμική αγωγιμότητα των υλικών αλλαγής φάσης. Καταλύτες πυρηνοποίησης (nucleating catalysts) βρίσκονται σε υλικά, όπου το φαινόμενο υπέρψυξης (supercooling, η διαδικασία της μείωσης της θερμοκρασίας ενός υγρού ή αερίου κάτω από το σημείο πήξης του χωρίς να γίνει στερεό) αποτελεί κοινό πρόβλημα. Οι αλλαγές όγκου μπορούν να αντιμετωπιστούν με διάφορους τρόπους. Εάν οι περιορισμοί χώρου είναι σημαντικότεροι από τους περιορισμούς βάρους, ένα μεταλλικό PCM μπορεί να παρέχει την ίδια κατ' όγκο λανθάνουσα θερμότητα αλλαγής φάσης/τήξης (latent heat of fusion) με άλλα PCM, προσφέροντας ταυτόχρονα υψηλότερη θερμική αγωγιμότητα.

Γενικά, ένα ιδανικό PCM θα είχε τις ακόλουθες ιδιότητες:

-Υψηλή λανθάνουσα θερμότητα τήξης. Αυτή η ιδιότητα καθορίζει τη διαθέσιμη ενέργεια και είναι σημαντική είτε κατά βάρος είτε κατ' όγκο.

-Αναστρεψιμότητα μετάβασης από στερεή σε υγρή φάση. Η σύνθεση στερεής και υγρής φάσης θα πρέπει να είναι ίδια.

-Υψηλή θερμική αγωγιμότητα. Απαραίτητη για την αποφυγή μεγάλων θερμοκρασιακών διακυμάνσεων. Fillers χρησιμοποιούνται για την αύξηση της απόδοσης του υλικού.

-Υψηλή ειδική θερμοχωρητικότητα και πυκνότητα.

-Μακροχρόνια αξιοπιστία λόγω επαναλαμβανόμενων κύκλων λειτουργίας.

-Αξιόπιστη ψυκτική συμπεριφορά.

-Μικρές αλλαγές όγκου κατά την αλλαγή φάσεων.

-Χαμηλές πιέσεις ατμού.

Συνοψίζοντας, σύμφωνα με μια ρεαλιστικότερη προσέγγιση, τα κριτήρια επιλογής είναι:

-Θερμοδυναμικές ιδιότητες.

1. Σημείο τήξης εντός του επιθυμητού θερμοκρασιακού εύρους λειτουργίας.

2. Υψηλή λανθάνουσα θερμότητα αλλαγής φάσης ανά μονάδα όγκου.

3. Υψηλή θερμοχωρητικότητα, αγωγιμότητα και πυκνότητα.

4. Μικρές ογκομετρικές αλλαγές κατά την αλλαγή φάσεων και μικρή πίεση ατμού.

5. Ομοιόμορφη τήξη.

-Κινητικές ιδιότητες.

1. Υψηλός ρυθμός πυρηνοποίησης για την αποφυγή υπέρψυξης της υγρής φάσης.

2. Υψηλός ρυθμός κρυσταλλικής ανάπτυξης, ώστε να καλύπτει τις απαιτήσεις ανάκτησης θερμότητας από το σύστημα αποθήκευσης.

-Χημικές ιδιότητες.

1. Χημική σταθερότητα.

2. Πλήρης αναστρέψιμότητα κύκλου τήξης-πήξης.

3. Απουσία υποβάθμισης υλικού μετά από μεγάλο αριθμό κύκλων τήξης-πήξης.

4. Τα PCM θα πρέπει να είναι μη τοξικά, μη εύφλεκτα και μη διαβρωτικά.

-Οικονομικές ιδιότητες.

- 1.Κόστος.
- 2.Διαθεσιμότητα.

A.2.4-ΕΦΑΡΜΟΓΕΣ ΥΛΙΚΩΝ ΑΛΛΑΓΗΣ ΦΑΣΗΣ [4], [5]

Οι εφαρμογές των υλικών αλλαγής φάσης περιλαμβάνουν, χωρίς να περιορίζονται σε αυτές, τις εξής:

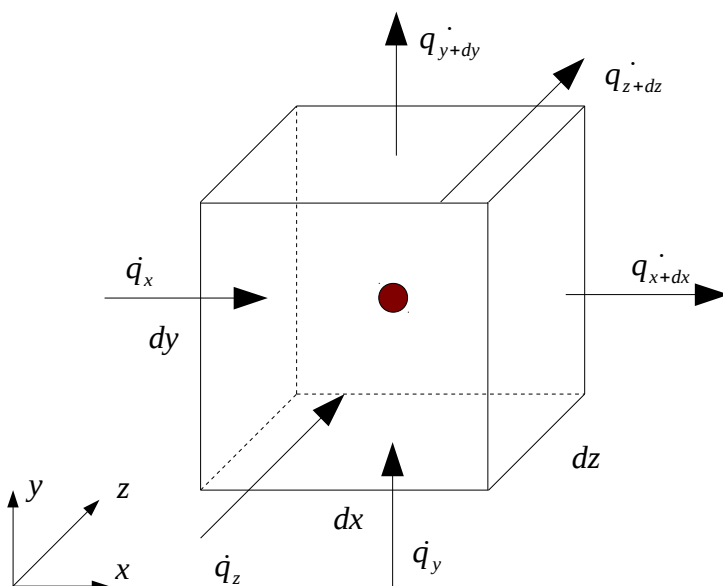
- αποθήκευση θερμικής ενέργειας (thermal energy storage)
- εκμετάλλευση/ανάκτηση αποβαλλόμενης θερμότητας
- κλιματισμός κτιρίων
- ψύξη/προστασία ηλεκτρονικών υπολογιστών και γενικά ηλεκτρονικών εξαρτημάτων/διατάξεων
- ψύξη/προστασία μηχανών,συσκευών και γενικότερα εγκαταστάσεων
- ιατρικές εφαρμογές (μεταφορά αίματος, ψυχο-θερμοθεραπείες κ.α.)
- συστήματα αντλιών θερμότητας
- συντήρηση τροφίμων
- οχήματα (εξασφάλιση θερμικής άνεσης)
- ρουχισμός (εξασφάλιση θερμικής άνεσης)

A.3-ΕΞΙΣΩΣΗ ΘΕΡΜΟΤΗΤΑΣ [8], [9], [10]

Η γενική μορφή της εξίσωσης διάχυσης θερμότητας (heat diffusion equation), γνωστή και απλά σαν εξίσωση θερμότητας είναι [8]:

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)+\dot{\Phi}=\rho C_p\frac{\partial T}{\partial t} \quad (\text{A.3.1})$$

Είναι το αποτέλεσμα της διατήρησης της ενέργειας σε στοιχειώδη όγκο αναφοράς (σχήμα A.3.1) και η λύση της αποδίδει τη θερμοκρασιακή κατανομή $T(x,y,z)$ συναρτήσει του χρόνου t . Οι όροι $\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)$, $\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)$, $\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)$ περιγράφουν τη θερμότητα που άγεται κατά τις διευθύνσεις x , y και z αντίστοιχα, ο όρος $\dot{\Phi}$ τον ρυθμό παραγωγής ή κατανάλωσης ενέργειας στο εσωτερικό του όγκου ελέγχου και ο όρος $\rho C_p\frac{\partial T}{\partial t}$ τον ρυθμό μεταβολής του ενεργειακού περιεχομένου στον όγκο ελέγχου.



Σχήμα A.3.1: Στοιχειώδης όγκος ελεγχου διαστάσεων dx , dy , dz [8].

Συχνά χρησιμοποιούνται απλοποιημένες εκδοχές της (A.3.1). Για μονοδιάστατη μεταφορά θερμότητας (κατά x), σταθερή θερμοκρασιακή αγωγιμότητα και μηδενική παραγωγή ενέργειας στο εσωτερικό του όγκου ελέγχου, η εξίσωση θερμότητας παίρνει τη μορφή:

$$\frac{\partial T}{\partial t}=\alpha\frac{\partial^2 T}{\partial x^2} \quad (\text{A.3.2})$$

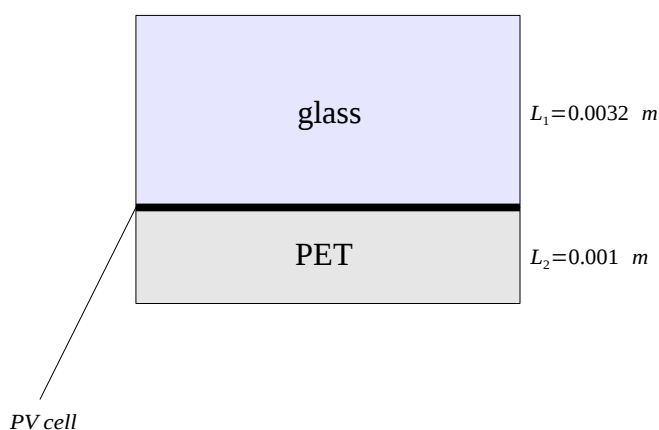
όπου $\alpha=\frac{k}{\rho C_p}\left[\frac{m^2}{s}\right]$, η θερμοκρασιακή αγωγιμότητα του υλικού (thermal diffusivity), η οποία αποτελεί μέτρο του πόσο γρήγορα διαχέεται η θερμότητα στο εσωτερικό ενός υλικού [10]. Η (A.3.2) είναι η μορφή της εξίσωσης θερμότητας που θα μας απασχολήσει στη συνέχεια.

A.4-MONTEΛΟΠΟΙΗΣΗ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ [8-16]

Η μονάδα που επιλέχθηκε προς μοντελοποίηση είναι η KC175GHT-2 της Kyocera. Πρόκειται για πολυκρυσταλλική μονάδα, υψηλής απόδοσης (ξεπερνά το 16 %) και μέγιστης ισχύος $175 W_p$ (ισχύς κάτω από STC, standard testing conditions). Τα χαρακτηριστικά της, καθώς και η γεωμετρία της είναι προσβάσιμα από τη σελίδα της εταιρείας [11]. Λόγω απλής γεωμετρίας και εξαιρετικά μικρού λόγου πάχους/επιφάνειας, θα θεωρήσουμε μονοδιάστατη αγωγή θερμότητας, κάθετη στη φωτοβολταϊκή επιφάνεια.

Πιο συγκεκριμένα θεωρούμε ότι η φωτοβολταϊκή μονάδα αποτελείται από:

- Ενισχυμένο γυαλί (tempered soda-lime glass), πάχους 3.2 mm.
- Πλαστικό υπόστρωμα (PET-polyethylene terephthalate) πάνω στο οποίο εντυπώνονται οι ηλιακές κυψέλες, πάχους 1 mm. Επειδή οι ηλιακές κυψέλες έχουν μικρό πάχος, θεωρούμε ότι το υπόστρωμα PET είναι εξ' ολοκλήρου πλαστικό. Ακολουθούν σκαρίφημα της γεωμετρίας της φωτοβολταϊκής μονάδας και πίνακας ιδιοτήτων των υλικών [12]:



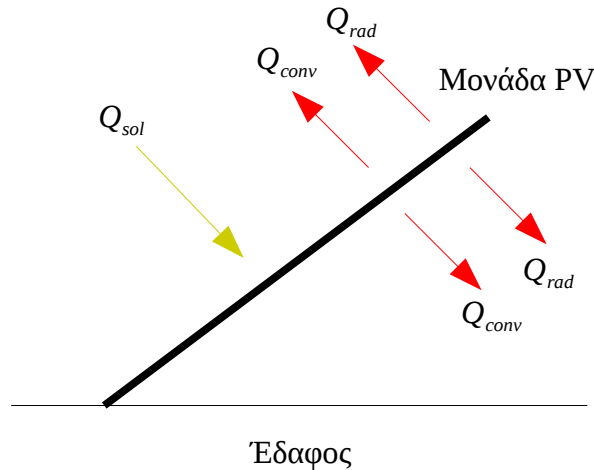
Σχήμα A.4.1: Απλοποιημένη γεωμετρία φωτοβολταϊκής μονάδας.

Πίνακας A.4.1: Πίνακας ιδιοτήτων των υλικών.

Ιδιότητα\Υλικό	1: glass	2: PET
$L [m]$	0.0032	0.001
$k [\frac{W}{mK}]$	1.05	0.195
$\rho [\frac{kg}{m^3}]$	2440	1470
$C_p [\frac{kJ}{kgK}]$	0.72	1.075

Η ενεργειακή συναλλαγή της μονάδας p_n με το περιβάλλον καθορίζεται από πολλές παραμέτρους: τη γεωμετρία της μονάδας, τις θερμοφυσικές ιδιότητες των υλικών της, τις καιρικές συνθήκες, καθώς και τους συντελεστές μεταφοράς θερμότητας (ακτινοβολίας και συναγωγής). Το σχήμα A.4.2 αποτελεί σχηματική αναπαράσταση της. Στην άνω επιφάνεια έχουμε εισροή θερμότητας με τη μορφή ηλιακής ακτινοβολίας, Q_{sol} , και απώλειες θερμότητας λόγω συναγωγής και ακτινοβολίας $Q_{conv} + Q_{rad}$, ενώ, στην κάτω επιφάνεια απώλειες θερμότητας λόγω συναγωγής και

ακτινοβολίας $Q_{conv}+Q_{rad}$.



Σχήμα A.4.2: Ενεργειακή συναλλαγή με το περιβάλλον

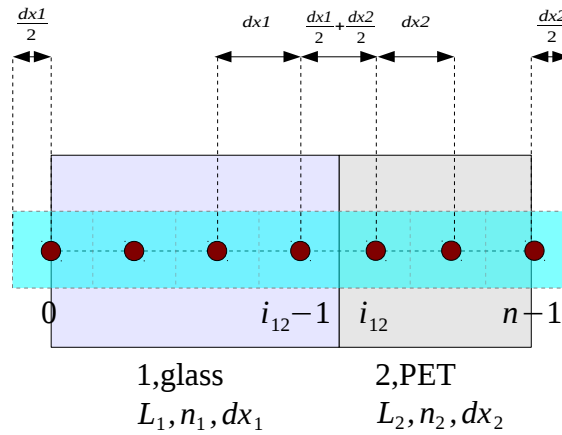
Εξίσωση θερμότητας

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, η μονοδιάστατη εξίσωση θερμότητας, που περιγράφει τη μεταφορά θερμότητας στο εσωτερικό της μονάδας PV, είναι:

$$\frac{\partial T}{\partial t} = \frac{k}{\rho C_p} \frac{\partial^2 T}{\partial x^2} \quad (\text{A.4.1})$$

και θα λυθεί με τη μέθοδο των πεπερασμένων διαφορών. Θα μετατραπεί, δηλαδή σε εξίσωση πεπερασμένων διαφορών, εκφράζοντας όλες τις παραγώγους ως συνδυασμό των τιμών της συνάρτησης σε διακριτές θέσεις στη γειτονιά του υπό εξέταση χώρου (αναπτύγματα κατά Taylor). Υπάρχουν διάφορες προσεγγίσεις, η κάθε μία με διαφορετικά πλεονεκτήματα και μειονεκτήματα. Οι κυριότερες εκφράσεις είναι η ρητή, η άρρητη/πεπλεγμένη και η Crank-Nicolson. Στην παρούσα εργασία θα χρησιμοποιηθεί η άρρητη/πεπλεγμένη έκφραση, καθώς παρουσιάζει άνευ όρων ευστάθεια και επιτρέπει μεγάλα χρονικά βήματα dt . Αν και έχει μεγαλύτερες προγραμματιστικές απαιτήσεις, καθώς οδηγεί στην επίλυση συστήματος αλγεβρικών εξισώσεων, η τριδιαγώνια μορφή του συστήματος και τα μεγάλα επιτρεπόμενα χρονικά βήματα οδηγούν σε σημαντική μείωση του υπολογιστικού χρόνου σε σχέση με τις άλλες μεθόδους [13].

Πριν μετατρέψουμε την (A.4.1) σε εξίσωση πεπερασμένων διαφορών, θα πρέπει, πρώτα, να λάβει χώρα η διακριτοποίηση του χώρου. Καθώς πρόκειται για πολυστρωματικό σώμα, θα χρησιμοποιηθεί ανομοιόμορφο πλέγμα (σχήμα A.4.3). Θεωρούμε στοιχειώδεις όγκους μήκους dx_1 για το υλικό 1 (γυαλί) και dx_2 για το υλικό 2 (υπόστρωμα PET) στο κέντρο των οποίων βρίσκονται οι κόμβοι. Κάθε κόμβος ορίζεται από το δείκτη i , ο οποίος παίρνει τιμές μεταξύ 0 και $(n-1)$. Οι στοιχειώδεις όγκοι είναι διατεταγμένοι έτσι ώστε ο πρώτος ($i=0$) και ο τελευταίος κόμβος ($i=n-1$) να συμπίπτουν με τα φυσικά όρια της μονάδας pv (άνω και κάτω επιφάνεια) αλλά και οι δύο διακριτοποιήσεις να "συναντιούνται" στη μεταξύ τους διεπιφάνεια. Είναι φανερό ότι ο κόμβος της κυψέλης (i_{12}), ο οποίος ανήκει στο υλικό 2, δεν θα συμπίπτει με τη διεπιφάνεια μεταξύ των υλικών 1-2, αλλά για αρκετά πυκνό πλέγμα θα βρίσκεται ικανοποιητικά κοντά. Σε διαφορετική περίπτωση, όπου ο κόμβος i_{12} συμπίπτει με τη διεπιφάνεια των υλικών, δε μπορούν να οριστούν οι θερμοφυσικές ιδιότητες που αντιστοιχούν στον κόμβο. Ακολουθεί σχηματική αναπαράσταση της διακριτοποίησης χώρου.



Σχήμα Α.4.3: Διακριτοποίηση χώρου μονάδας pv.

Υποθέτουμε αριθμό κόμβων n_1 , για το υλικό 1 και n_2 , για το υλικό 2. Τότε η απόσταση μεταξύ δύο διαδοχικών κόμβων θα είναι:

$$dx_1 = \frac{2L_1}{2n_1 - 1} \quad \text{για το υλικό 1 (γυαλί)}$$

$$dx_2 = \frac{2L_2}{2n_2 - 1} \quad \text{για το υλικό 2 (υπόστρωμα PET)}$$

Προφανώς, η απόσταση μεταξύ των κόμβων i_{12-1} , i_{12} είναι: $\frac{dx_1}{2} + \frac{dx_2}{2}$.

Επίσης $i_{12} = n_1$, αλλαγή υλικού 1-2 (κόμβος κυψέλης, γυαλί-PET).

Για λόγους προγραμματιστικής απλότητας, θεωρούμε ότι σε κάθε κόμβο i αντιστοιχούν οι τιμές k_i (θερμική αγωγιμότητα), ρ_i (πυκνότητα), C_{pi} (ειδική θερμοχωρητικότητα), x_i (απόσταση από την αρχή, $i=0$) και $dx_i = x_{i+1} - x_i$ (χωρικό βήμα). Έτσι, διαμορφώνονται τα παρακάτω μητρώα (σχήμα Α.4.4). Να σημειωθεί ότι για συνολικό αριθμό κόμβων $n = n_1 + n_2$, όλα τα μητρώα έχουν διαστάσεις $n \times 1$ εκτός του dx , που έχει $(n-1) \times 1$.

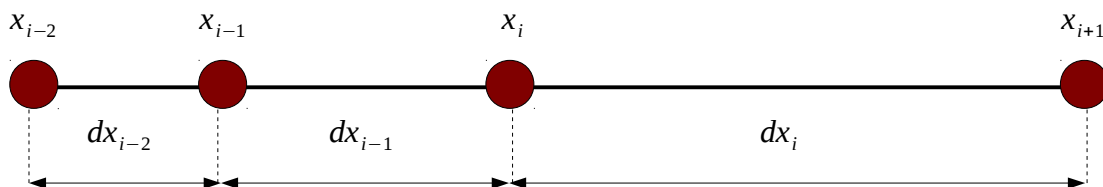
i	k_i	ρ_i	C_{pi}	x_i	dx_i
0	k_1	ρ_1	C_{p1}	x_0	dx_1
1	k_1	ρ_1	C_{p1}	x_1	dx_1
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$i_{12}-1$	k_1	ρ_1	C_{p1}	$x_{i_{12}-1}$	$\frac{dx_1}{2} + \frac{dx_2}{2}$
i_{12}	k_2	ρ_2	C_{p2}	$x_{i_{12}}$	\vdots
\vdots	\vdots	\vdots	\vdots	\vdots	dx_2
$n-1$	k_2	ρ_2	C_{p2}	x_{n-1}	

Σχήμα A.4.4: Μητρώα k, ρ, C_p, x, dx .

Σύμφωνα με την πεπλεγμένη μέθοδο η χρονική παράγωγος διακριτοποιείται με εμπρός διαφορά και η χωρική με κεντρική διαφορά όρων κατά την επόμενη χρονική στιγμή ($p+1$). Λόγω ανομοιόμορφου πλέγματος θα χρησιμοποιηθεί η αντίστοιχη έκφραση ανομοιόμορφου πλέγματος (σχήμα A.4.5). Εφεξής, ο δείκτης $i=0,1,2,\dots,(n-1)$ θα αναφέρεται στην κατεύθυνση x και ο δείκτης $p=0,1,2,\dots,(m-1)$ στη χρονική στιγμή.

$$\frac{\partial T}{\partial t} = \frac{T_i^{p+1} - T_i^p}{dt} \quad (\text{A.4.2})$$

$$\frac{\partial^2 T}{\partial t^2} = \frac{dx_i T_{i-1}^{p+1} - (dx_{i-1} + dx_i) T_i^{p+1} + dx_{i-1} T_{i+1}^{p+1}}{\frac{1}{2} dx_{i-1} dx_i (dx_{i-1} + dx_{i+1})} \quad (\text{A.4.3})$$



Σχήμα A.4.5: Ανομοιόμορφη διάταξη κόμβων.

Αντικαθιστώντας τις (A.4.2) και (A.4.3) στην (A.4.1):

$$\frac{T_i^{p+1} - T_i^p}{dt} = \frac{k_i}{\rho_i C_{p,i}} \frac{dx_i T_{i-1}^{p+1} - (dx_{i-1} + dx_i) T_i^{p+1} + dx_{i-1} T_{i+1}^{p+1}}{\frac{1}{2} dx_{i-1} dx_i (dx_{i-1} + dx_{i+1})} \rightarrow$$

$$T_i^{p+1} - T_i^p = \frac{k_i dt}{\rho_i C_{p,i} \frac{1}{2} dx_{i-1} dx_i (dx_{i-1} + dx_{i+1})} (dx_i T_{i-1}^{p+1} - (dx_{i-1} + dx_i) T_i^{p+1} + dx_{i-1} T_{i+1}^{p+1})$$

Για λόγους απλότητας θέτουμε ένα συντελεστή r_i , που εξαρτάται από τον κόμβο i :

$$r_i = \frac{k_i dt}{\rho_i C_{p,i} \frac{1}{2} dx_{i-1} dx_i (dx_{i-1} + dx_i)}$$

$$T_i^{p+1} - T_i^p = r_i (dx_i T_{i-1}^{p+1} - (dx_{i-1} + dx_i) T_i^{p+1} + dx_{i-1} T_{i+1}^{p+1}) \rightarrow$$

$$T_i^{p+1} - T_i^p = r_i dx_i T_{i-1}^{p+1} - r_i (dx_{i-1} + dx_i) T_i^{p+1} + r_i dx_{i-1} T_{i+1}^{p+1} \rightarrow$$

$$T_i^{p+1} - r_i dx_i T_{i-1}^{p+1} + r_i (dx_{i-1} + dx_i) T_i^{p+1} - r_i dx_{i-1} T_{i+1}^{p+1} = T_i^p \rightarrow$$

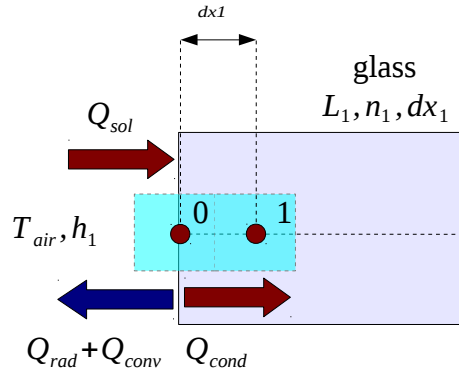
$$-r_i dx_i T_{i-1}^{p+1} + (1 + r_i (dx_{i-1} + dx_i)) T_i^{p+1} - r_i dx_{i-1} T_{i+1}^{p+1} = T_i^p, \quad i=1,2,\dots,n-2 \quad (\text{A.4.4})$$

Έτσι η εξίσωση (A.4.1) μετασχηματίστηκε στην (A.4.4), η οποία ισχύει για τους εσωτερικούς κόμβους $i=1,2,\dots,(n-2)$. Στους εξωτερικούς κόμβους $i=0$ και $i=(n-1)$ ισχύουν οι οριακές συνθήκες.

Οριακές συνθήκες

$i=0$

Η άνω επιφάνεια δέχεται θερμότητα με τη μορφή ηλιακής ακτινοβολίας, Q_{sol} , και παρουσιάζει απώλειες θερμότητας λόγω συναγωγής και ακτινοβολίας $Q_{\text{conv}} + Q_{\text{rad}}$ (σχήμα A.4.6).



Σχήμα A.4.6: Οριακή συνθήκη για $i=0$.

$$Q_{cond}^{p+1} = -Q_{rad}^{p+1} - Q_{conv}^{p+1} + Q_{sol}^{p+1} \quad (\text{A.4.5})$$

-Θερμικό κέρδος λόγω ηλιακής ακτινοβολίας:

$$Q_{sol}^{p+1} = a_{glass} * G_T^{p+1} \quad (\text{A.4.6})$$

όπου:

$a_{glass}=0.96$ [-] [12], ο συντελεστής απορρόφησης του γυαλιού
 G_T [kW/m²], η στιγμιαία ολική ηλιακή ακτινοβολία κεκλιμένου επιπέδου, ο υπολογισμός της οποίας θα μας απασχολήσει στη συνέχεια.

-Θερμικές απώλειες λόγω ακτινοβολίας σώματος προς το περιβάλλον:

$$Q_{rad}^{p+1} = \epsilon \sigma (T_0^{p+1} - T_{air}^{p+1})$$

όπου:

ϵ [-], ο συντελεστής εκπομπής της σώματος
 $\sigma=5.6697 \cdot 10^{-8}$ W/m²K⁴ [9], η σταθερά Boltzman

Επειδή, η εισαγωγή άγνωστης θερμοκρασίας στην τέταρτη δύναμη θα μετατρέψει το σύστημα σε μη γραμμικό, υποθέτουμε συντελεστή ακτινοβολίας h_{r1} , τέτοιο ώστε:

$$Q_{rad}^{p+1} = h_{r1} (T_0^{p+1} - T_{air}^{p+1})$$

όπου:

h_{r1} [W/m²K], ισοδύναμος συντελεστής μεταφοράς θερμότητας λόγω ακτινοβολίας.

-Θερμικές απώλειες λόγω ελεύθερης συναγωγής προς το περιβάλλον:

$$Q_{conv}^{p+1} = h_{c1} (T_0^{p+1} - T_{air}^{p+1})$$

όπου:

h_{c1} , η ειδική συναγωγιμότητα

Έτσι έχουμε:

$$Q_{conv}^{p+1} + Q_{rad}^{p+1} = h_1(T_0^{p+1} - T_{air}^{p+1})$$

όπου:

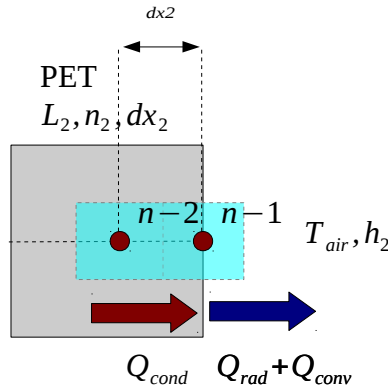
$h_1 = h_{c_1} + h_{r_1}$ [8], συντελεστής μεταφοράς θερμότητας λόγω ακτινοβολίας και συναγωγής.

Συνεπώς, η (A.4.5) γίνεται:

$$\begin{aligned} -k_1 \frac{T_1^{p+1} - T_0^{p+1}}{dx_1} &= -h_1(T_0^{p+1} - T_{air}^{p+1}) + a_{glass} G_T^{p+1} \rightarrow \\ -\left(\frac{k_1}{dx_1}\right)T_1^{p+1} + \frac{k_1}{dx_1}T_0^{p+1} + h_1 T_0^{p+1} &= h_1 T_{air}^{p+1} + a_{glass} G_T^{p+1} \rightarrow \\ \left(\frac{k_1}{dx_1} + h_1\right)T_0^{p+1} - \left(\frac{k_1}{dx_1}\right)T_1^{p+1} &= h_1 T_{air}^{p+1} + a_{glass} G_T^{p+1} \end{aligned} \quad (A.4.7)$$

i=(n-1)

Η κάτω επιφάνεια παρουσιάζει απώλειες θερμότητας λόγω συναγωγής και ακτινοβολίας $Q_{conv} + Q_{rad}$ (σχήμα 2.2).



Σχήμα A.4.7: Οριακή συνθήκη για $i=0$.

Ακολουθώντας την ίδια προσέγγιση, θεωρώ τέτοιο συντελεστή μεταφοράς θερμότητας h_2 , ώστε $h_2 = h_{c_2} + h_{r_2}$. Συνεπώς, από την ισότητα ρευμάτων θερμότητας, παίρνουμε:

$$\begin{aligned} Q_{cond}^{p+1} &= Q_{rad}^{p+1} + Q_{conv}^{p+1} \rightarrow \\ -k_2 \frac{T_{n-1}^{p+1} - T_{n-2}^{p+1}}{dx_2} &= h_2(T_{n-1}^{p+1} - T_{air}^{p+1}) \rightarrow \\ -\frac{k_2}{dx_2}T_{n-1}^{p+1} + \frac{k_2}{dx_2}T_{n-2}^{p+1} - h_2 T_{n-1}^{p+1} &= -h_2 T_{air}^{p+1} \rightarrow \end{aligned} \quad (A.4.8)$$

$$-\left(\frac{k_2}{dx_2}\right)T_{n-2}^{p+1} + \left(\frac{k_2}{dx_2} + h_2\right)T_{n-1}^{p+1} = h_2 T_{air}^{p+1} \quad (\text{A.4.9})$$

Ο υπολογισμός είτε της ειδικής συναγωγιμότητας h_c , είτε του γενικού συντελεστή μεταφοράς $h = h_c + h_r$, είναι μια πολύπλοκη διαδικασία και μπορεί να βρεθεί με τρόπο μαθηματικό ή και πειραματικό. Για λόγους απλότητας, στην παρούσα εργασία θα ληφθεί μία σχετικά χαμηλή τιμή η οποία θα λαμβάνει υπόψη ελεύθερη συναγωγή και ακτινοβολία [9]. Όπως αναφέρθηκε και στο κεφάλαιο A.1.2, η θερμοκρασία της κυψέλης κάτω από αντίξοες συνθήκες μπορεί να φτάσει μέχρι και 30 °C υψηλότερα από τη θερμοκρασία του αέρα. Επιλέχθηκε τιμή έτσι ώστε την περίοδο του Ιουλίου (περίοδος μέγιστης θερμοκρασίας αέρα, B.1) η θερμοκρασιακή διαφορά να "αγγίζει" τους 30 °C. Μετά από δοκιμές λήφθηκε (B.4):

$$h_1 = h_2 = 4 \frac{W}{m^2 K} \quad (\text{A.4.10})$$

Αρχικές συνθήκες

Υποθέτουμε αρχική τιμή θερμοκρασίας ίση με τη θερμοκρασία περιβάλλοντος για κάθε κόμβο:

$$T_i^0 = T_{air}^0, \quad i=0,1,2,\dots,(n-1).$$

Σύστημα εξισώσεων

Λαμβάνοντας υπόψη τα παραπάνω, καταλήγουμε στο παρακάτω σύστημα:

$$\left(\frac{k_1}{dx_1} + h_1\right)T_0^{p+1} - \left(\frac{k_1}{dx_1}\right)T_1^{p+1} = h_1 T_{air}^{p+1} + a_{glass} G_T^{p+1}, \quad i=0$$

$$-r_i dx_i T_{i-1}^{p+1} + (1 + r_i(dx_{i-1} + dx_i))T_i^{p+1} - r_i dx_{i-1} T_{i+1}^{p+1} = T_i^p, \quad i=0,1,2,\dots,(n-2)$$

$$-\left(\frac{k_2}{dx_2}\right)T_{n-2}^{p+1} + \left(\frac{k_2}{dx_2} + h_2\right)T_{n-1}^{p+1} = h_2 T_{air}^{p+1}, \quad i=(n-1)$$

$$\text{όπου: } r_i = \frac{k_i dt}{\rho_i C_{p,i} \frac{1}{2} dx_{i-1} dx_i (dx_{i-1} + dx_i)}$$

Συντελεστές συστήματος

Θεωρούμε τα μητρώα ld , d , ud , b στα οποία θα "αποθηκευτούν" οι συντελεστές του παραπάνω συστήματος εξισώσεων.

ld (κάτω-διαγώνια στοιχεία)
 d (διαγώνια στοιχεία)
 ud (άνω-διαγώνια στοιχεία)
 b (στοιχεία δεξιού πίνακα)

Για $i=0$:

$$d_0 = \left(\frac{k_1}{dx_1}\right) + h_1$$

$$ud_0 = -\left(\frac{k_1}{dx_1}\right)$$

$$b_0 = h_1 T_{air}^{p+1} + a_{glass} G_T^{p+1}$$

Για $i=1,2,3,\dots,(n-2)$:

$$ld_{i-1} = -r_i dx_i$$

$$d_i = 1 + r_i (dx_{i-1} + dx_i)$$

$$ud_i = -r_i dx_{i-1}$$

$$b_i = T_i^p$$

Για $i=(n-1)$:

$$ld_{n-2} = -\left(\frac{k_2}{dx_2}\right)$$

$$d_{n-1} = \left(\frac{k_2}{dx_2}\right) + h_2$$

$$b_{n-1} = h_2 T_{air}^{p+1}$$

Γράφοντας τους συντελεστές του συστήματος σε μορφή πινάκων, παίρνουμε το σχήμα A.4.8:

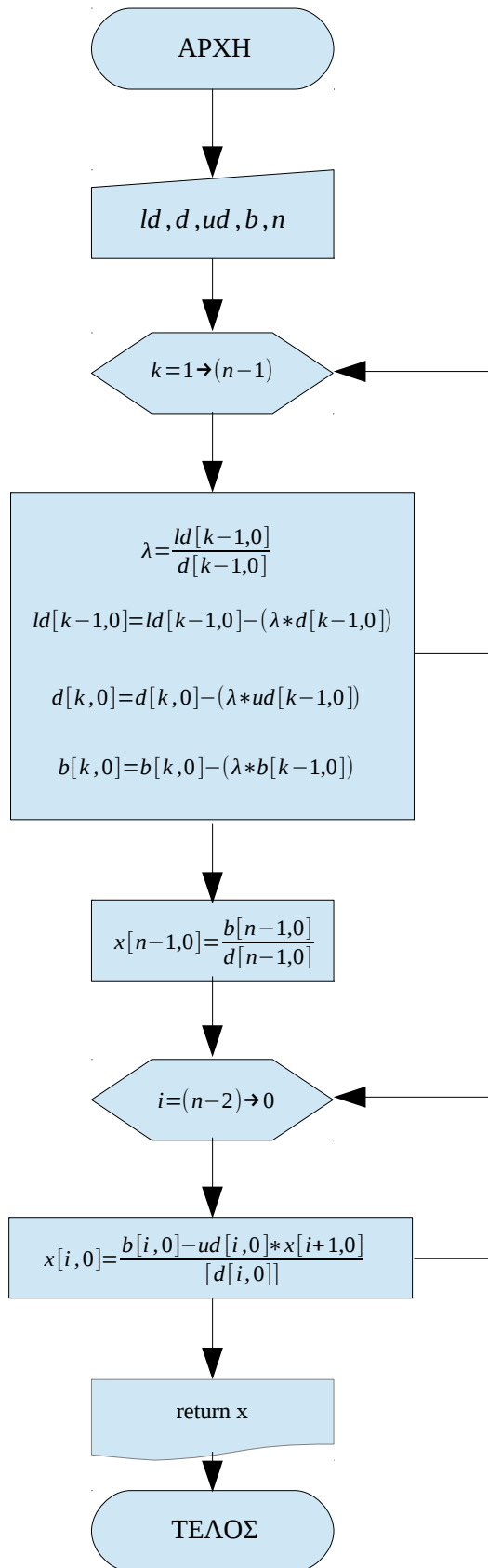
$$\begin{array}{|cccc|}
\hline
d_0 & ud_0 & & \\
\hline
ld_0 & d_1 & ud_1 & \\
\hline
& \ddots & \ddots & \ddots \\
\hline
& & ld_{i_{12}-1} & d_{i_{12}} & ud_{i_{12}} \\
\hline
& & & \ddots & \ddots & \ddots \\
\hline
& & & & ld_{n-3} & d_{n-2} & ud_{n-2} \\
\hline
& & & & & ld_{n-2} & d_{n-1} \\
\hline
\end{array}
\quad \times \quad
\begin{array}{|c|}
\hline
T_0^{p+1} \\
\hline
T_1^{p+1} \\
\hline
\vdots \\
\hline
T_{i_{12}}^{p+1} \\
\hline
\vdots \\
\hline
T_{n-2}^{p+1} \\
\hline
T_{n-1}^{p+1} \\
\hline
\end{array}
=
\begin{array}{|c|}
\hline
b_0 \\
\hline
b_1 \\
\hline
\vdots \\
\hline
b_{i_{12}} \\
\hline
\vdots \\
\hline
b_{n-2} \\
\hline
b_{n-1} \\
\hline
\end{array}$$

Σχήμα A.4.8: Σύστημα εξισώσεων τριδιαγώνιας μορφής.

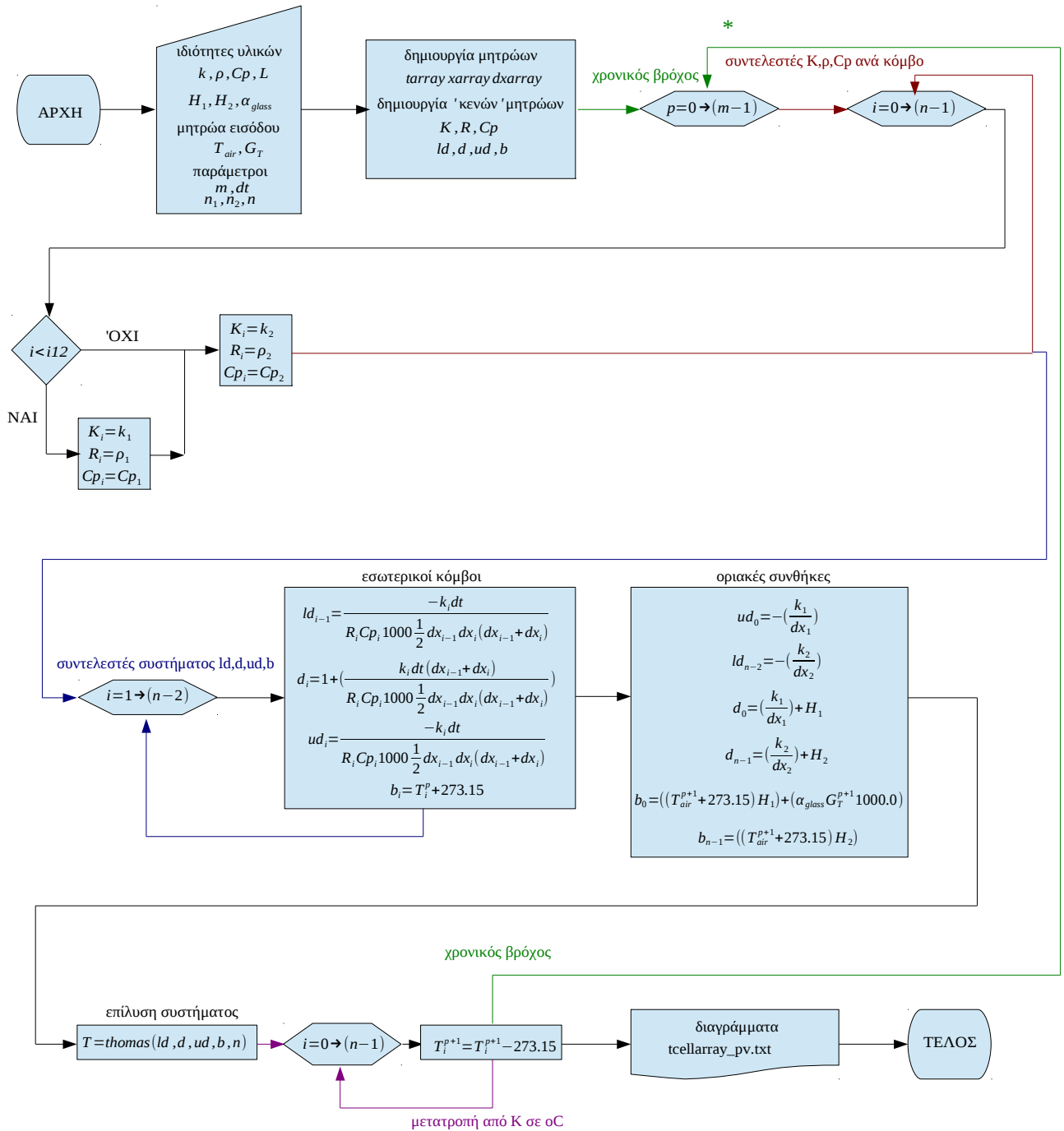
Το παραπάνω σύστημα τριδιαγώνιας μορφής θα λυθεί με τον αλγόριθμο Thomas ή αλλιώς TDMA (tridiagonal matrix algorithm) [14], [15]. Για το σκοπό αυτό γράφτηκε η ρουτίνα `thomas_algorithm.py` (Γ.8.1) η οποία δέχεται τα μητρώα `ld`, `d`, `ud`, `b` και το συνολικό αριθμό κόμβων `n` και επιστρέφει το μητρώο της λύσης.

Για την εύρεση της θερμοκρασιακής κατανομής της μονάδας `pn` σε κάθε χρονική στιγμή, τα στοιχεία που υπολείπονται είναι οι ωριαίες τιμές της θερμοκρασίας αέρα, T_{air} , οι τιμές της στιγμιαίας ολικής ακτινοβολίας κεκλιμένου επιπέδου ανά ώρα, G_T . Οι ωριαίες τιμές της θερμοκρασίας αέρα ανακτήθηκαν από το ιστορικό δεδομένων του μετεωρολογικού σταθμού του Ε.Μ.Π. [16], για το έτος 2000 (B.1). Ο υπολογισμός της ηλιακής ακτινοβολίας ακολουθεί στο επόμενο κεφάλαιο. Έχοντας όλα τα απαραίτητα δεδομένα, ο κώδικας `implicit_pn.py` (Γ.3) μας επιστρέφει τη θερμοκρασιακή κατανομή της μονάδας `pn` και κατά συνέπεια τη θερμοκρασία κυψέλης, καθώς και τα αντίστοιχα διαγράμματα. Τα αποτελέσματα παρατίθενται στο υπολογιστικό μέρος (B.3). Ακολουθούν τα διαγράμματα ροής του αλγορίθμου `thomas` και του προγράμματος `implicit_pn.py`.

Θα πρέπει να σημειωθεί ότι, επειδή η αρχική θερμοκρασιακή κατανομή είναι άγνωστη το σύστημα θα πρέπει να λυθεί αρχικά για ένα μικρό αριθμό επαναλήψεων (*). Θεωρούμε θερμοκρασία αέρα σταθερή και ίση με τη θερμοκρασία αέρα της πρώτης χρονικής στιγμής. Ομοίως πράττουμε και για την ένταση της ηλιακής ακτινοβολίας. Αρχική θερμοκρασία για όλους του κόμβους τίθεται ίση με την αρχική θερμοκρασία αέρα. Μετά από 10 επαναλήψεις και με γνωστή, πλέον, τη θερμοκρασιακή κατανομή οι υπολογισμοί συνεχίζονται κανονικά.



Σχήμα Α.4.9: Διάγραμμα ροής αλγορίθμου Thomas (TDMA).



Σχήμα Α.4.10: Διάγραμμα ροής κώδικα implicit_pn.py

Α.5-ΥΠΟΛΟΓΙΣΜΟΣ ΣΤΙΓΜΙΑΙΑΣ ΟΛΙΚΗΣ ΗΛΙΑΚΗΣ ΑΚΤΙΝΟΒΟΛΙΑΣ ΚΕΚΛΙΜΕΝΟΥ ΕΠΙΠΕΔΟΥ [3]

Ο υπολογισμός της στιγμιαίας ολικής ηλιακής ακτινοβολίας κεκλιμένου επιπέδου θα βασιστεί στον υπολογισμό της μέσης στιγμιαίας άμεσης και διάχυτης ακτινοβολίας οριζοντίου επιπέδου, G_b , G_d , από εξισώσεις που προέκυψαν μετά από λογιστική ανάλυση των μετρήσεων της ολικής ακτινοβολίας κατά την εικοσαετία 1961-1980. Οι μετρήσεις αυτές λήφθηκαν από το Αστεροσκοπείο Αθηνών επί ωριαίας βάσης [3].

Η θερμότητα λόγω ηλιακής ακτινοβολίας για κεκλιμένο επίπεδο και για κάθε χρονική στιγμή p , υπολογίζεται από την παρακάτω σχέση:

$$Q_{sol}^p = \alpha_{glass} G_T^p \left[\frac{W}{m^2} \right] \quad (A.5.1)$$

όπου:

α_{glass} [-] ο συντελεστής απορρόφησης ηλιακής ακτινοβολίας του υλικού

$G_T \left[\frac{W}{m^2} \right]$ η στιγμιαία ολική ηλιακή ακτινοβολία κεκλιμένου επιπέδου

Η στιγμιαία ολική ακτινοβολία κεκλιμένου επιπέδου G_T , με τη σειρά της, υπολογίζεται από την παρακάτω σχέση:

$$G_T = G_b R_b + G_d \left(\frac{1 + \cos \beta}{2} \right) + (G_b + G_d) \rho \left(\frac{1 - \cos \beta}{2} \right) \quad (A.5.2)$$

όπου:

$G_b \left[\frac{W}{m^2} \right]$, η στιγμιαία άμεση ακτινοβολία οριζοντίου επιπέδου

$G_d \left[\frac{W}{m^2} \right]$, η στιγμιαία διάχυτη ακτινοβολία οριζοντίου επιπέδου

$R_b = \frac{G_{bT}}{G_b} = \frac{\cos \theta}{\cos \theta_z}$ [-] (A.5.3), ο λόγος άμεσης ακτινοβολίας κεκλιμένου επιπέδου προς την άμεση ακτινοβολία οριζοντίου επιπέδου

ρ [-], η ανακλαστικότητα του εδάφους ($\rho=0.2$ για συνηθισμένο έδαφος, $\rho=0.7$ για χιονισμένο έδαφος)

β [degrees], η γωνία κλίσης επιφάνειας, σε σχέση με το οριζόντιο επίπεδο (0° για οριζόντια επιφάνεια, 90° για κάθετη επιφάνεια, για $\beta > 90^\circ$ η επιφάνεια "βλέπει" προς τα κάτω.)

θ [degrees], η γωνία πρόσπτωσης της αμέσου ακτινοβολίας πάνω σε μια επιφάνεια. Είναι ίση με τη γωνία μεταξύ των προσπιπτουσών ηλιακών ακτίνων και της καθέτου στην επιφάνεια.

$\theta_z [degrees]$, η γωνία ζενίθ. Είναι ίση με τη γωνία που σχηματίζεται από την κατακόρυφο και την ευθεία όρασης του ηλίου.

Μέσα από τη μελέτη της μεταβολής της ολικής ακτινοβολίας ετήσιας ενέργειας συναρτήσει του αζιμούθιου γ της επιφάνειας για διάφορες κλίσεις β , προκύπτει ότι ο βέλτιστος προσανατολισμός της επιφάνειας είναι ο νότιος ($\gamma=0^\circ$) για το βόρειο ημισφαίριο και ο βόρειος ($\gamma=180^\circ$) για το νότιο ημισφαίριο (κοινώς, η επιφάνεια πρέπει να αντικρίζει τον ισημερινό), όπου:

$\gamma [degrees]$, το αζιμούθιο επιφάνειας. Η γωνία που σχηματίζεται από την προβολή της ευθείας όρασης του ηλίου στο οριζόντιο επίπεδο και της νότιας κατεύθυνσης. Είναι $-180^\circ \leq \gamma \leq 180^\circ$ με $\gamma=0^\circ$ στο νότο, $\gamma=90^\circ$ στη δύση, $\gamma=180^\circ$ στο βορρά και $\gamma=-90^\circ$ στην ανατολή.

Με βάση το κεφάλαιο A.1.4, θέλοντας να μελετήσουμε τη θερμική συμπεριφορά του συστήματος για ένα εύρος χρονικών περιόδων, επιλέγουμε $\gamma=0^\circ$ για μέγιστη ετήσια ενέργεια και $\beta=\varphi=37.97^\circ$.

Ο υπολογισμός της μέσης στιγμιαίας άμεσης και διάχυτης ακτινοβολίας, G_b και G_d αντίστοιχα, για κάθε μέρα του έτους $D=1 \dots 365$ και για κάθε ώρα $t=7,8,\dots,18$ θα γίνει με τις σχέσεις:

$$G_b = M_t + A_t \sin\left(\frac{360}{365} D - P_t\right) \quad (A.5.4)$$

$$G_d = m_t + a_t \sin\left(\frac{360}{365} D - p_t\right) \quad (A.5.5)$$

Οι σχέσεις αυτές, όπως προαναφέρθηκε, προέκυψαν από τη λογιστική ανάλυση δεδομένων εικοσαετίας (μετρήσεις των αντίστοιχων μεγεθών από το Αστεροσκοπείο Αθηνών για την περίοδο 1961-1980). Οι τιμές των συντελεστών M_t , A_t , P_t , m_t , a_t , p_t για κάθε ώρα μεταξύ 7 π.μ. και 18 μ.μ. δίνονται από τον παρακάτω πίνακα A.5.1:

Πίνακας Α.5.1: Τιμές των συντελεστών των εξισώσεων (Α.5.4) και (Α.5.5) [3].

t [hours]	M_t [kW/m ²]	A_t [kW/m ²]	Pt [degrees]	m_t [kW/m ²]	a_t [kW/m ²]	pt [degrees]
07.00	0.007	0.046	86.18	0.048	0.111	81.23
08.00	0.057	0.068	87.31	0.134	0.127	82.45
09.00	0.119	0.104	87.94	0.208	0.133	83.94
10.00	0.191	0.129	88.31	0.261	0.129	85.60
11.00	0.258	0.153	89.35	0.283	0.117	86.05
12.00	0.319	-0.181	-88.88	0.271	0.099	82.71
13.00	0.342	-0.194	-89.18	0.243	0.081	81.95
14.00	0.325	0.183	87.21	0.210	0.083	87.23
15.00	0.269	0.166	86.22	0.176	0.087	89.53
16.00	0.183	0.147	82.10	0.141	-0.085	-88.19
17.00	0.078	0.131	79.85	0.104	-0.079	-88.97
18.00	-0.025	0.114	77.36	0.051	0.091	86.79

Για τον υπολογισμό του λόγου της άμεσης ακτινοβολίας κεκλιμένου επιπέδου προς την άμεση ακτινοβολία οριζοντίου επιπέδου, θα χρησιμοποιηθεί η εξίσωση (Α.5.3):

$$R_b = \frac{\cos \theta}{\cos \theta_z} \quad [-] \quad (\text{A.5.3})$$

Τα συνημίτονα της γωνίας πρόσπτωσης και της γωνίας ζενίθ υπολογίζονται από τις παρακάτω εξισώσεις:

$$\begin{aligned} \cos \theta = & \sin \delta \sin \varphi \cos \beta - \sin \delta \cos \varphi \sin \beta \cos \gamma \\ & + \cos \delta \cos \varphi \cos \beta \cos \omega + \cos \delta \sin \varphi \sin \beta \cos \gamma \cos \omega \\ & + \cos \delta \sin \beta \sin \gamma \sin \omega \end{aligned} \quad (\text{A.5.6})$$

$$\cos \theta_z = \sin \delta \sin \varphi + \cos \delta \cos \varphi \cos \omega \quad (\text{A.5.7})$$

όπου:

ω , η ωριαία γωνία. Πρόκειται για τη γωνιακή μετατόπιση του ηλίου ανατολικά ή δυτικά του τοπικού μεσημβρινού λόγω της περιστροφής της γης περί τον άξονά της με ρυθμό $\frac{15^\circ}{h} = \frac{0.25^\circ}{min}$.

Προφανώς ισχύει:

$\omega = 0.25 \times (\text{min από το ηλιακό μεσημέρι})$. Λαμβάνεται $\omega > 0$ για μ.μ. και $\omega < 0$ για π.μ.

δ , η ηλιακή απόκλιση. Είναι η γωνιακή θέση του ηλίου κατά το ηλιακό μεσημέρι ως προς το επίπεδο του ισημερινού. Ισχύει $-23.45 \leq \delta \leq 23.45$ με θετικές τιμές προς τον βορρά. Υπολογίζεται από τη σχέση:

$$\delta = 23.45 \sin \left(\frac{360(284+n)}{365} \right) \quad (\text{A.5.8})$$

όπου $n = 1 \dots 365$ η μέρα του έτους, η οποία μπορεί να υπολογιστεί εύκολα με τη βοήθεια του πίνακα 5.2. Ο πίνακας A.5.2 δίνει, επίσης, την δ για τη μέση ημέρα κάθε μήνα (μέρα του μήνα με ηλιακή εκτινοβολία εκτός ατμόσφαιρας ίση περίπου με την αντίστοιχη μέση του μήνα).

Πίνακας A.5.2: Αύξων αριθμός ημέρας έτους, n. Μέση ημέρα κάθε μήνα και αντίστοιχη τιμή της ηλιακής απόκλισης, δ .

Μήνας	Μέρα του μήνα n	Μέση μέρα του μήνα		
		ημερομηνία	μέρα έτους, n	απόκλιση, δ
Ιανουάριος	i	17	17	-20.9
Φεβρουάριος	31+i	16	47	-13
Μάρτιος	59+i	16	75	-2.4
Απρίλιος	90+i	15	105	9.4
Μάιος	120+i	15	135	18.8
Ιούνιος	151+i	11	162	23.1
Ιούλιος	181+i	17	198	21.2
Αύγουστος	212+i	16	228	13.5
Σεπτέμβριος	243+i	15	258	2.2
Οκτώβριος	273+i	15	288	-9.6
Νοέμβριος	304+i	14	318	-18.9
Δεκέμβριος	334+i	10	344	-23.0

Ο κώδικας insolation.py (Γ.2) για αρχική μέρα, συνολικό αριθμό ημερών και κλίση επιφάνειας αποδίδει σε μορφή κειμένου (insolation.txt) το μητρώο της ηλιακής ακτινοβολίας και τα αντίστοιχα διαγράμματα. Τα αποτελέσματα παρατίθενται στο υπολογιστικό μέρος (B.2).

A.6-ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΥΛΙΚΩΝ ΑΛΛΑΓΗΣ ΦΑΣΗΣ [17], [18]

Οι κυριότερες μέθοδοι μοντελοποίησης των υλικών αλλαγής φάσης είναι η μέθοδος ενθαλπίας και η μέθοδος ενεργού θερμοχωρητικότητας.

-Μοντέλο ενθαλπίας

Βασίζεται στην επίλυση της εξίσωσης μεταφοράς θερμότητας με την ενθαλπία σαν εξαρτημένη μεταβλητή. Υποτίθεται ισόθερμη αλλαγή φάσης. Επιτρέπει τον υπολογισμό της αισθητής και της λανθάνουσας θερμότητας καθώς και λεπτομερειών κατά την αλλαγή φάσης.

-Μοντέλο ενεργού θερμοχωρητικότητας

Το μοντέλο αυτό, παρότι δεν επιτρέπει τον υπολογισμό λεπτομερειών κατά την αλλαγή φάσης, διακρίνεται για την ευκολία υλοποίησης, καθώς πρόκειται περί αντικατάστασης της ειδικής θερμοχωρητικότητας του PCM, $C_{p_{pcm}}$ με την ενεργό θερμοχωρητικότητα $C_{p_{eff}} = C_{p_{eff}}(T)$, συνάρτηση της θερμοκρασίας, στη διαφορική εξίσωση που περιγράφει τη μεταφορά θερμότητας. Διαφορετικά:

$$C_p = C_{ps}, \quad T < T_s, \quad \text{στερεή φάση}$$

$$C_p = C_{eff}(T), \quad T_s \leq T \leq T_l, \quad \text{αλλαγή φάσης}$$

$$C_p = C_{pl}, \quad T > T_l, \quad \text{υγρή φάση}$$

Μοντέλο ενεργού θερμοχωρητικότητας

Για την εφαρμογή της συγκεκριμένης μεθόδου απαιτείται η έκφραση της ενεργού θερμοχωρητικότητας συναρτήσει της θερμοκρασίας για συγκεκριμένο υλικό αλλαγής φάσης. Στην παρούσα εργασία έγινε χρήση των αποτελεσμάτων της δημοσίευσης των Ε. Δ. Κραββαρίτη, Κ. Α. Αντωνόπουλο και Χ. Τζιβανίδη, στην οποία προσδιορίστηκε πειραματικά, μετά απο επεξεργασία καμπυλών ψύξης, η συνάρτηση $C_{p_{eff}} = C_{p_{eff}}(T)$ για διάφορα υλικά (ελληνική παραφίνη, RT20, RT27, RT58, SP25A8) [17]. Τα υλικά που μας αφορούν άμεσα είναι αυτά που η αλλαγή φάσης τους λαμβάνει χώρα στην περιοχή των 25 °C (θερμοκρασία αναφοράς των φωτοβολταϊκών μονάδων), ήτοι RT20 (15-26 °C, H=132.1 kJ/kg), RT27 (22-31 °C, H=167.4 kJ/kg), SP25A8 (22-32 °C, H=141.5 kJ/kg). Ο πίνακας Α.6.1 δίνει τις ιδιότητες των τριών υλικών [17], [18]. Μετά από εκτενείς δοκιμές βρέθηκε ότι η καλύτερη αναλυτική έκφραση των $C_{p_{eff}} = C_{p_{eff}}(T)$ επιτυγχάνεται χρησιμοποιώντας τέσσερις διακριτές συναρτήσεις για κάθε PCM.

RT20

$$\begin{aligned} C_{peff} &= 444.4711352 - 170.5210626T + 25.80658991T^2 \\ &- 1.912462169T^3 + 0.06910480624T^4 \\ &- 0.0009667426392T^5 \end{aligned} \quad (A.6.1)$$

για $15^\circ C \leq T \leq 19.5^\circ C$

$$\begin{aligned} C_{peff} &= 5529.086311 - 565.4394794T + 14.50378915T^2 \\ &\text{για } 19.5^\circ C < T \leq 20^\circ C \end{aligned} \quad (A.6.2)$$

$$\begin{aligned} C_{peff} &= 3780.001949 - 342.8478233T + 7.797040983T^2 \\ &\text{για } 20.5^\circ C < T \leq 22^\circ C \end{aligned} \quad (A.6.3)$$

$$\begin{aligned} C_{peff} &= 3735.906512 - 536.9879458T + 28.98735267T^2 \\ &- 0.6959772441T^3 + 0.006270432248T^4 \\ &\text{για } 22^\circ C < T \leq 26^\circ C \end{aligned} \quad (A.6.4)$$

RT27

$$\begin{aligned} C_{peff} &= -329.5000062 + 106.3162953T - 13.35423683T^2 \\ &+ 0.8223480772T^3 - 0.02488671396T^4 \\ &+ 0.0002972388922T^5 \end{aligned} \quad (A.6.5)$$

για $22^\circ C \leq T \leq 25^\circ C$

$$\begin{aligned} C_{peff} &= 9542.790195 - 774.7503271T + 15.74031481T^2 \\ &\text{για } 25^\circ C < T \leq 26.8^\circ C \end{aligned} \quad (A.6.6)$$

$$\begin{aligned} C_{peff} &= 41611.7679 - 3023.692869T + 54.96396748T^2 \\ &\text{για } 26.8^\circ C < T \leq 27.5^\circ C \end{aligned} \quad (A.6.7)$$

$$\begin{aligned} C_{peff} &= 31155.44075 - 3820.483549T + 175.7678837T^2 \\ &- 3.594806096T^3 + 0.02757236338T^4 \\ &\text{για } 27.5^\circ C < T \leq 31^\circ C \end{aligned} \quad (A.6.8)$$

SP25A8

$$\begin{aligned} C_{peff} &= -5822.7083 + 1559.960873T - 166.178906T^2 \\ &+ 8.806172538T^3 - 0.2322802438T^4 \\ &+ 0.002441877714T^5 \end{aligned} \quad (A.6.9)$$

για $22^\circ C \leq T \leq 23.8^\circ C$

$$C_{peff} = 30643.84674 - 2584.495777T + 54.51400097T^2$$

για $23.8^\circ C < T \leq 24.5^\circ C$ (A.6.10)

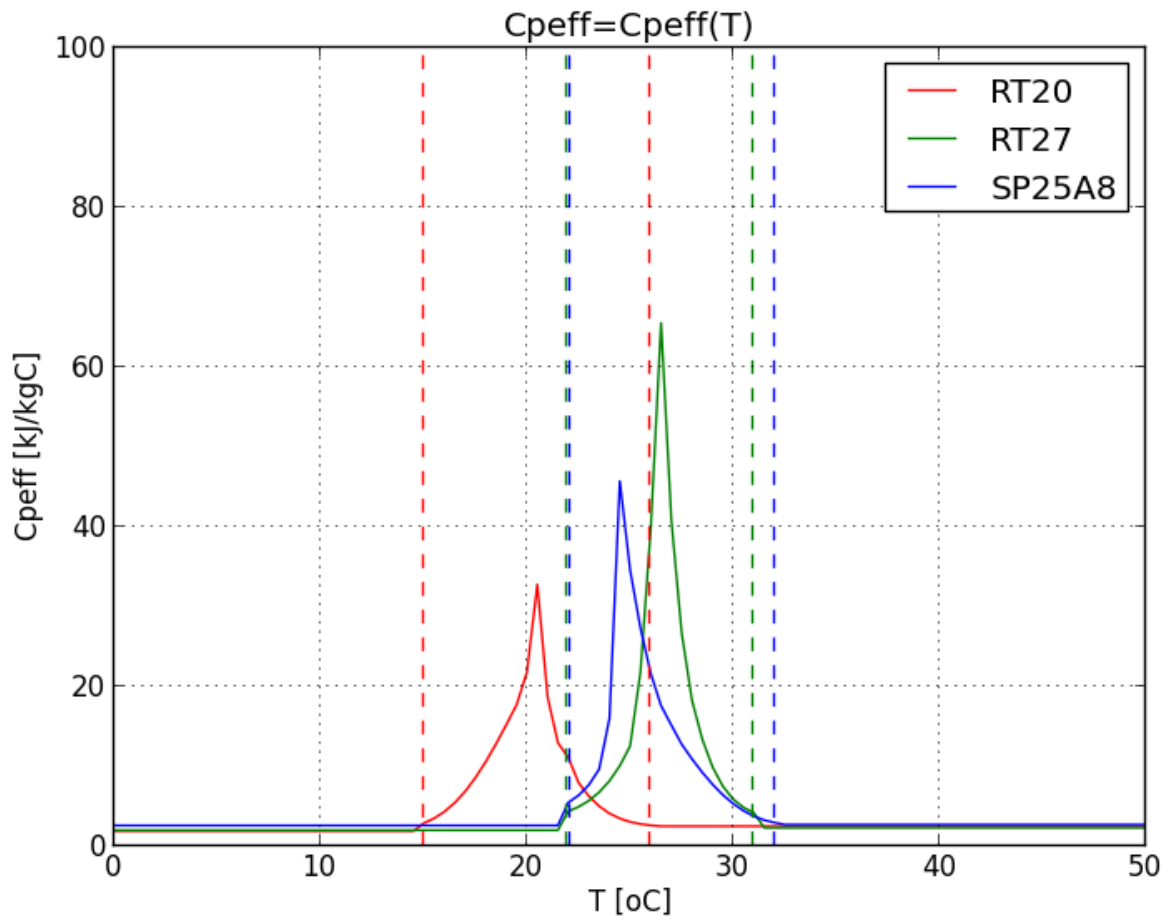
$$C_{peff} = 2456.776607 - 177.5933256T + 3.228251256T^2$$

για $24.5^\circ C < T \leq 27.2^\circ C$ (A.6.11)

$$C_{peff} = -3900.417315T + 553.0906948T - 28.64445051T^2 + 0.6468310842T^3 - 0.005396817351T^4$$

για $27.2^\circ C < T \leq 32^\circ C$ (A.6.12)

Οι ρουτίνες `effective_heat_capacity_rt20.py` (Γ.8.2), `effective_heat_capacity_rt27.py` (Γ.8.3) και `effective_heat_capacity_sp25a8.py` (Γ.8.4) αποδίδουν την ενεργή θερμοχωρητικότητα για τα υλικά RT20, RT27, SP25A8 αντίστοιχα. Ο κώδικας `effective_heat_capacity_comparison_plot.py` (Γ.5), αποδίδει το συγκριτικό διάγραμμα των ενεργών θερμοχωρητικοτήτων (σχήμα A.6.1).



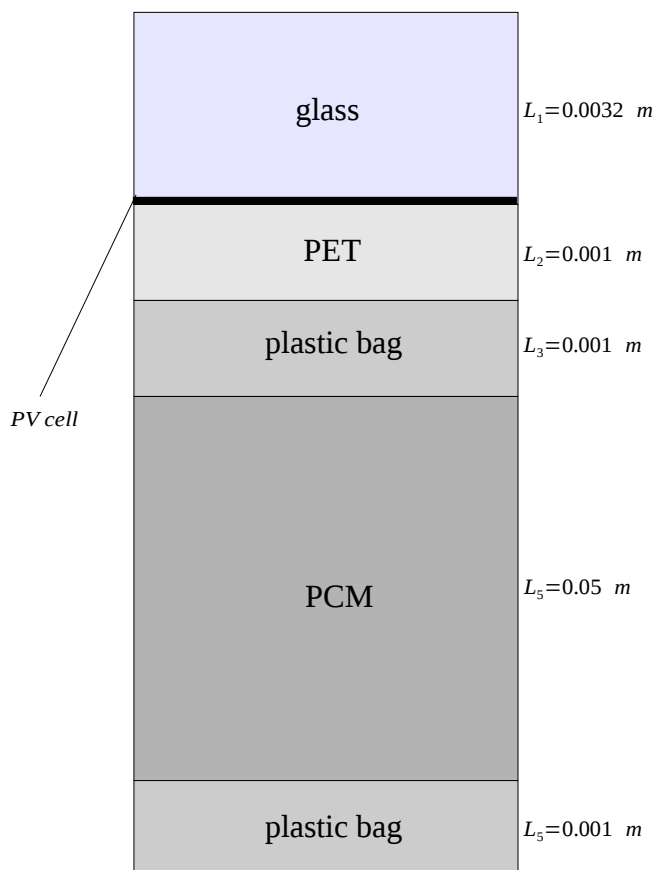
Διάγραμμα A.6.1: Συγκριτικό διάγραμμα ενεργών θερμοχωρητικοτήτων για τα υλικά RT20, RT27 και SP25A8.

Πίνακας Α.6.1: Πίνακας ιδιοτήτων για τα υλικά RT20, RT27 και SP25A8 [17], [18].

Ιδιότητα\Υλικό	RT20	RT27	SP25A8
$\rho \left[\frac{kg}{m^3} \right]$	880	880	1500
$k \left[\frac{W}{mK} \right]$	0.2	0.2	0.6
$C_p \left[\frac{kJ}{kgK} \right]$	$C_{peff}(T)$	$C_{peff}(T)$	$C_{peff}(T)$
$C_{ps} \left[\frac{kJ}{kgK} \right]$	1.9	2.0	2.6
$C_{pl} \left[\frac{kJ}{kgK} \right]$	2.5	2.3	2.7
$T_s \text{ [}^\circ\text{C]}$	15	22	22
$T_l \text{ [}^\circ\text{C]}$	26	31	32
$H \left[\frac{kJ}{kg} \right]$	132.1	167.4	141.5

A.7-ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-ΥΛΙΚΟΥ ΑΛΛΑΓΗΣ ΦΑΣΗΣ [12], [17], [18]

Όπως προαναφέρθηκε, ένας από τους προτεινόμενους τρόπους θερμικού ελέγχου μιας φωτοβολταϊκής μονάδας είναι η χρήση υλικών αλλαγής φάσης. Υποθέτουμε PCM πάχους 5 cm (0.05 m) περιεχόμενο σε πλαστική συσκευασία (LDPE, low density polyethylene) πάχους 1 mm (0.001 m) σε τέλεια επαφή με την κάτω επιφάνεια της μονάδας PV (διάκενο αέρα μεταξύ υποστρώματος PET και πλαστικής συσκευασίας αμελητέο). Πιο συγκεκριμένα, η γεωμετρία του συστήματος PV-PCM θα έχει την παρακάτω μορφή:



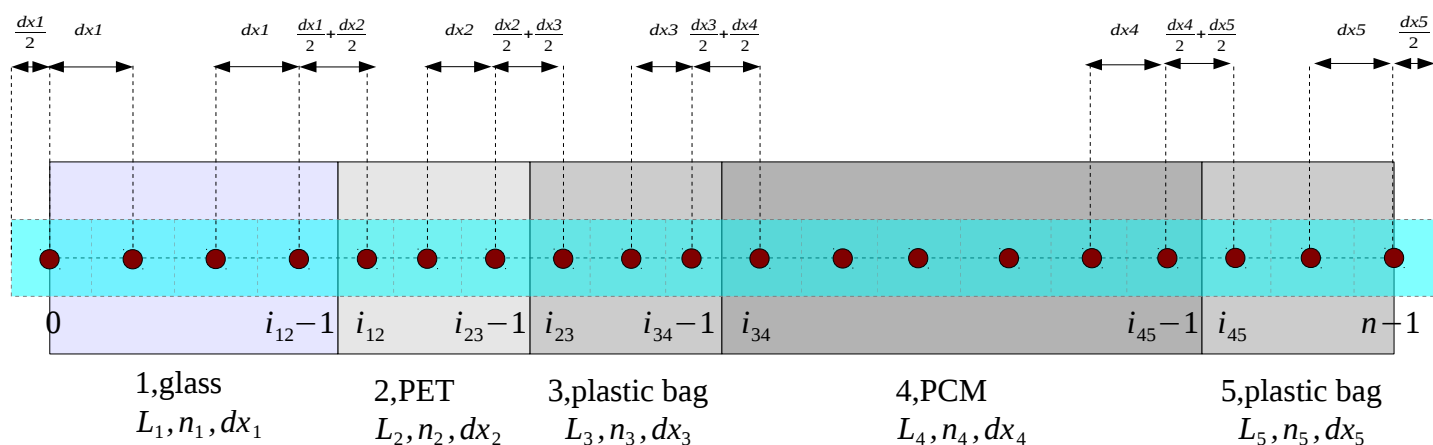
Σχήμα A.7.1: Απλοποιημένη γεωμετρία συστήματος PV-PCM.

Ακολουθεί πίνακας ιδιοτήτων των υλικών [17], [18]:

Πίνακας Α.7.1: Ιδιότητες υλικών ανά στρώμα [12], [17], [18].

Ιδιότητα\Υλικό	1: glass	2: PET	3: plastic	4: PCM RT20,RT27/SP25A8	5: plastic
L [m]	0.0032	0.001	0.001	0.05	0.001
k [$\frac{W}{mK}$]	1.05	0.195	0.36	0.2/0.6	0.36
ρ [$\frac{kg}{m^3}$]	2440	1470	920	880/1500	920
C_p [$\frac{kJ}{kgK}$]	0.72	1.075	2.3	$C_{peff}(T)$	2.3

Σε ό,τι αφορά τη διακριτοποίηση χώρου, ακολουθώντας την προσέγγιση του κεφαλαίου Α.2, παίρνουμε το παρακάτω σχήμα:



Σχήμα Α.7.2: Διακριτοποίηση χώρου συστήματος PV-PCM.

Υποθέτουμε αριθμό κόμβων n_1, n_2, n_3, n_4, n_5 , για τα υλικά 1, 2, 3, 4, 5 αντίστοιχα. Τότε η απόσταση μεταξύ δύο διαδοχικών κόμβων θα είναι:

$$dx_1 = \frac{2L_1}{2n_1 - 1} \quad \text{για το υλικό 1 (γυαλί)}$$

$$dx_2 = \frac{L_2}{n_2} \quad \text{για το υλικό 2 (υπόστρωμα PET)}$$

$$dx_3 = \frac{L_3}{n_3} \quad \text{για το υλικό 3 (πλαστική συσκευασία PCM)}$$

$$dx_4 = \frac{L_4}{n_4} \quad \text{για το υλικό 4 (PCM)}$$

$$dx_5 = \frac{2L_5}{2n_5 - 1} \text{ για το υλικό 5 (πλαστική συσκευασία PCM)}$$

$$\text{Απόσταση μεταξύ των κόμβων } i_{12-1}, i_{12}: \frac{dx_1}{2} + \frac{dx_2}{2}$$

$$\text{Απόσταση μεταξύ των κόμβων } i_{23-1}, i_{23}: \frac{dx_2}{2} + \frac{dx_3}{2}$$

$$\text{Απόσταση μεταξύ των κόμβων } i_{34-1}, i_{34}: \frac{dx_3}{2} + \frac{dx_4}{2}$$

$$\text{Απόσταση μεταξύ των κόμβων } i_{45-1}, i_{45}: \frac{dx_4}{2} + \frac{dx_5}{2}$$

Επίσης:

$$i_{12} = n_1, \text{ αλλαγή υλικού 1-2 (κόμβος κυψέλης, γυαλί-υπόστρωμα PET)}$$

$$i_{23} = n_1 + n_2, \text{ αλλαγή υλικού 2-3 (υπόστρωμα PET-πλαστική συσκευασία)}$$

$$i_{34} = n_1 + n_2 + n_3, \text{ αλλαγή υλικού 3-4 (πλαστική συσκευασία-PCM)}$$

$$i_{45} = n_1 + n_2 + n_3 + n_4, \text{ αλλαγή υλικού 4-5 (PCM-πλαστική συσκευασία)}$$

Όπως και στην περίπτωση της μοντελοποίησης της μονάδας PV, θεωρούμε ότι σε κάθε κόμβο i αντιστοιχούν οι τιμές k_i (θερμική αγωγιμότητα), ρ_i (πυκνότητα), C_{pi} (ειδική θερμοχωρητικότητα), x_i (απόσταση από την αρχή, $i=0$) και $dx_i = x_{i+1} - x_i$ (χωρικό βήμα). Έτσι, προκύπτουν τα παρακάτω μητρώα (σχήμα A.7.2). Για συνολικό αριθμό κόμβων $n = n_1 + n_2 + n_3 + n_4 + n_5$, όλα τα μητρώα έχουν διαστάσεις $n \times 1$ εκτός του dx που έχει $(n-1) \times 1$.

i	k_i	ρ_i	C_{p_i}	x_i	dx_i
0	k_1	ρ_1	C_{p1}	x_0	dx_1
1	k_1	ρ_1	C_{p1}	x_1	dx_1
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$i_{12}-1$	k_1	ρ_1	C_{p1}	$x_{i_{12}-1}$	$\frac{dx_1}{2} + \frac{dx_2}{2}$
i_{12}	k_2	ρ_2	C_{p2}	$x_{i_{12}}$	dx_2
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$i_{23}-1$	k_2	ρ_2	C_{p2}	$x_{i_{23}-1}$	$\frac{dx_2}{2} + \frac{dx_3}{2}$
i_{23}	k_3	ρ_3	C_{p3}	$x_{i_{23}}$	dx_3
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$i_{34}-1$	k_3	ρ_3	C_{p3}	$x_{i_{34}-1}$	$\frac{dx_3}{2} + \frac{dx_4}{2}$
i_{34}	k_4	ρ_4	C_{p4}	$x_{i_{34}}$	dx_4
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$i_{45}-1$	k_4	ρ_4	C_{p4}	$x_{i_{45}-1}$	$\frac{dx_4}{2} + \frac{dx_5}{2}$
i_{45}	k_5	ρ_5	C_{p5}	$x_{i_{45}}$	dx_5
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$n-1$	k_5	ρ_5	C_{p5}	x_{n-1}	

Σχήμα A.7.3: Μητρώα k, ρ, C_p, x, dx .

Σύστημα εξισώσεων

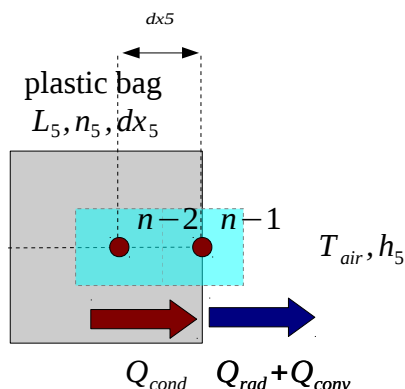
Το σύστημα εξισώσεων είναι της ίδιας μορφής (ίδια εξίσωση θερμότητας, οριακές συνθήκες, ίδιοι συντελεστές μεταφοράς θερμότητας $h_1=h_5=4 \text{ W/m}^2\text{K}$) με τις εξής διαφορές:

-Περιλαμβάνει περισσότερα στρώματα/υλικά.

-Στους κόμβους του PCM γίνεται αντικατάσταση της ειδικής θερμοχωρητικότητας με την ενεργή θερμοχωρητικότητα του, που αποτελεί συνάρτηση της θερμοκρασίας του κόμβου ($C_{\text{peff}}(T)$).

-Η οριακή συνθήκη της κάτω επιφάνειας ($i=n-1$), η οποία δεν περιλαμβάνει τον παράγοντα της ηλιακής ακτινοβολίας, κάνει χρήση των συντελεστών του υλικού 5 (πλαστική συσκευασία, σχήμα

A.7.4), δηλαδή:
$$-k_5 \frac{T_{n-1}^{p+1} - T_{n-2}^{p+1}}{dx_5} = h_5 (T_{n-1}^{p+1} - T_{air}^{p+1}) \quad (\text{A.7.1})$$



Σχήμα A.7.4: Οριακή συνθήκη για $i=(n-1)$ συστήματος PV-PCM.

Συνεπώς, το σύστημα είναι:

$$\left(\frac{k_1}{dx_1} + h_1\right) T_0^{p+1} - \left(\frac{k_1}{dx_1}\right) T_1^{p+1} = h_1 T_{air}^{p+1} + a_{\text{glass}} G_T^{p+1}, \quad i=0$$

$$-r_i dx_i T_{i-1}^{p+1} + (1 + r_i (dx_{i-1} + dx_i)) T_i^{p+1} - r_i dx_{i-1} T_{i+1}^{p+1} = T_i^p, \quad i=0,1,2,\dots,(n-2)$$

$$-\left(\frac{k_5}{dx_5}\right) T_{n-2}^{p+1} + \left(\frac{k_5}{dx_5} + h_5\right) T_{n-1}^{p+1} = h_5 T_{air}^{p+1}, \quad i=(n-1)$$

όπου:
$$r_i = \frac{k_i dt}{\rho_i C_{p,i} \frac{1}{2} dx_{i-1} dx_i (dx_{i-1} + dx_i)}$$

Συντελεστές συστήματος

Θεωρούμε τα μητρώα ld , d , ud , b στα οποία θα "αποθηκευτούν" οι συντελεστές του παραπάνω συστήματος εξισώσεων.

ld (κάτω-διαγώνια στοιχεία)

d (διαγώνια στοιχεία)

ud (άνω-διαγώνια στοιχεία)

b (στοιχεία δεξιού πίνακα)

Για $i=0$:

$$d_0 = \left(\frac{k_1}{dx_1}\right) + h_1$$

$$ud_0 = -\left(\frac{k_1}{dx_1}\right)$$

$$b_0 = h_1 T_{air}^{p+1} + a_{glass} G_T^{p+1}$$

Για $i=1,2,3,\dots,(n-2)$:

$$ld_{i-1} = -r_i dx_i$$

$$d_i = 1 + r_i (dx_{i-1} + dx_i)$$

$$ud_i = -r_i dx_{i-1}$$

$$b_i = T_i^p$$

Για $i=(n-1)$:

$$ld_{n-2} = -\left(\frac{k_5}{dx_5}\right)$$

$$d_{n-1} = \left(\frac{k_5}{dx_5}\right) + h_5$$

$$b_{n-1} = h_5 T_{air}^{p+1}$$

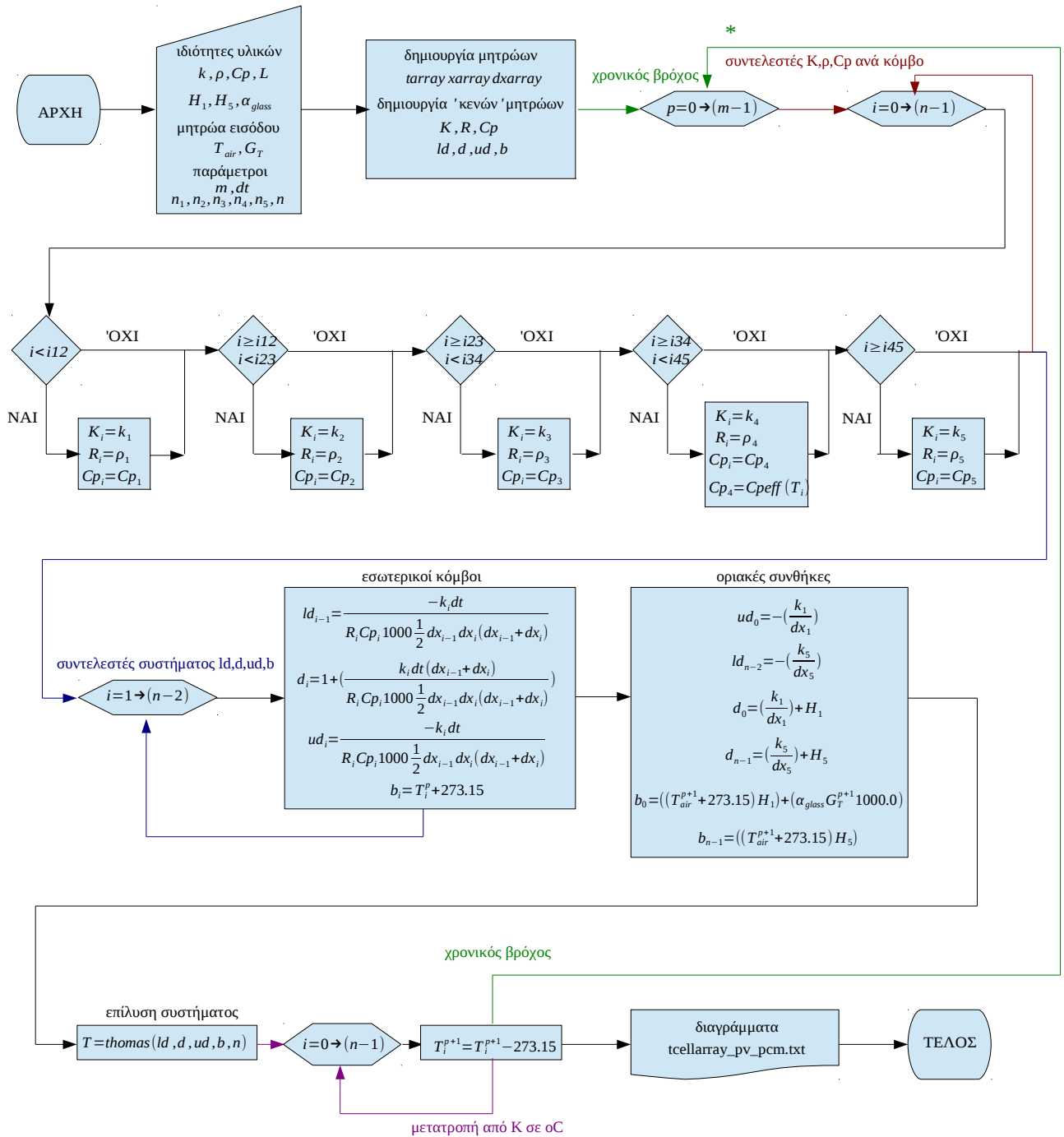
Γράφοντας τους συντελεστές του συστήματος σε μορφή πινάκων, παίρνουμε το σχήμα A.7.5:

$ \begin{matrix} d_0 & & ud_0 \\ ld_0 & d_1 & ud_1 \\ & \ddots & \ddots \\ & & \ddots \\ & ld_{i_{12}-1} & d_{i_{12}} & ud_{i_{12}} \\ & & \ddots & \ddots \\ & & & \ddots \\ & & ld_{i_{23}-1} & d_{i_{23}} & ud_{i_{23}} \\ & & & \ddots & \ddots \\ & & & & \ddots \\ & & ld_{i_{34}-1} & d_{i_{34}} & ud_{i_{34}} \\ & & & \ddots & \ddots \\ & & & & \ddots \\ & & ld_{i_{45}-1} & d_{i_{45}} & ud_{i_{45}} \\ & & & \ddots & \ddots \\ & & & & \ddots \\ & ld_{n-3} & d_{n-2} & ud_{n-2} \\ & & ld_{n-2} & d_{n-1} \end{matrix} $	X	$ \begin{matrix} T_0^{p+1} \\ T_1^{p+1} \\ \vdots \\ T_{i_{12}}^{p+1} \\ \vdots \\ T_{i_{23}}^{p+1} \\ \vdots \\ T_{i_{34}}^{p+1} \\ \vdots \\ T_{i_{45}}^{p+1} \\ \vdots \\ T_{n-2}^{p+1} \\ T_{n-1}^{p+1} \end{matrix} $	$ \begin{matrix} b_0 \\ b_1 \\ \vdots \\ b_{i_{12}} \\ \vdots \\ b_{i_{23}} \\ \vdots \\ b_{i_{34}} \\ \vdots \\ b_{i_{45}} \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{matrix} $
--	---	--	--

Σχήμα A.7.5: Σύστημα εξισώσεων τριδιαγώνιας μορφής.

Έτσι, προκύπτει το παραπάνω παρόμοιο τριδιαγώνιο σύστημα. Οι κώδικες `implicit_pn_rt20.py` (Γ.6.1), `implicit_pn_rt27.py` (Γ.6.2) και `implicit_pn_sp25a8.py` (Γ.6.3), που γράφτηκαν για τα υλικά RT20, RT27 και SP25A8 αντίστοιχα, αποδίδουν τη θερμοκρασιακή κατανομή του συστήματος PV-PCM και τη θερμοκρασία κυψέλης για κάθε χρονική στιγμή, καθώς και τα αντίστοιχα διαγράμματα. Τα αποτελέσματα παρατίθενται στο υπολογιστικό μέρος (B.6.1, B.6.2 και B.6.3). Ακολουθεί το διάγραμμα ροής των προγραμμάτων `implicit_pn_rt20.py`, `implicit_pn_rt27.py` και `implicit_pn_sp25a8.py`.

Όπως και στην περίπτωση του φωτοβολταϊκού συστήματος, επειδή η αρχική θερμοκρασιακή κατανομή είναι άγνωστη λαμβάνει χώρα ένας μικρός αριθμός επαναλήψεων (*) με βάση τις τιμές της πρώτης χρονικής στιγμής (θερμοκρασία αέρα, ένταση ηλιακής ακτινοβολίας).



Σχήμα Α.7.6: Κοινό διάγραμμα ροής για τους κώδικες `implicit_pv_rt20.py`, `implicit_pv_rt27.py` και `implicit_pv_sp25a8.py`.

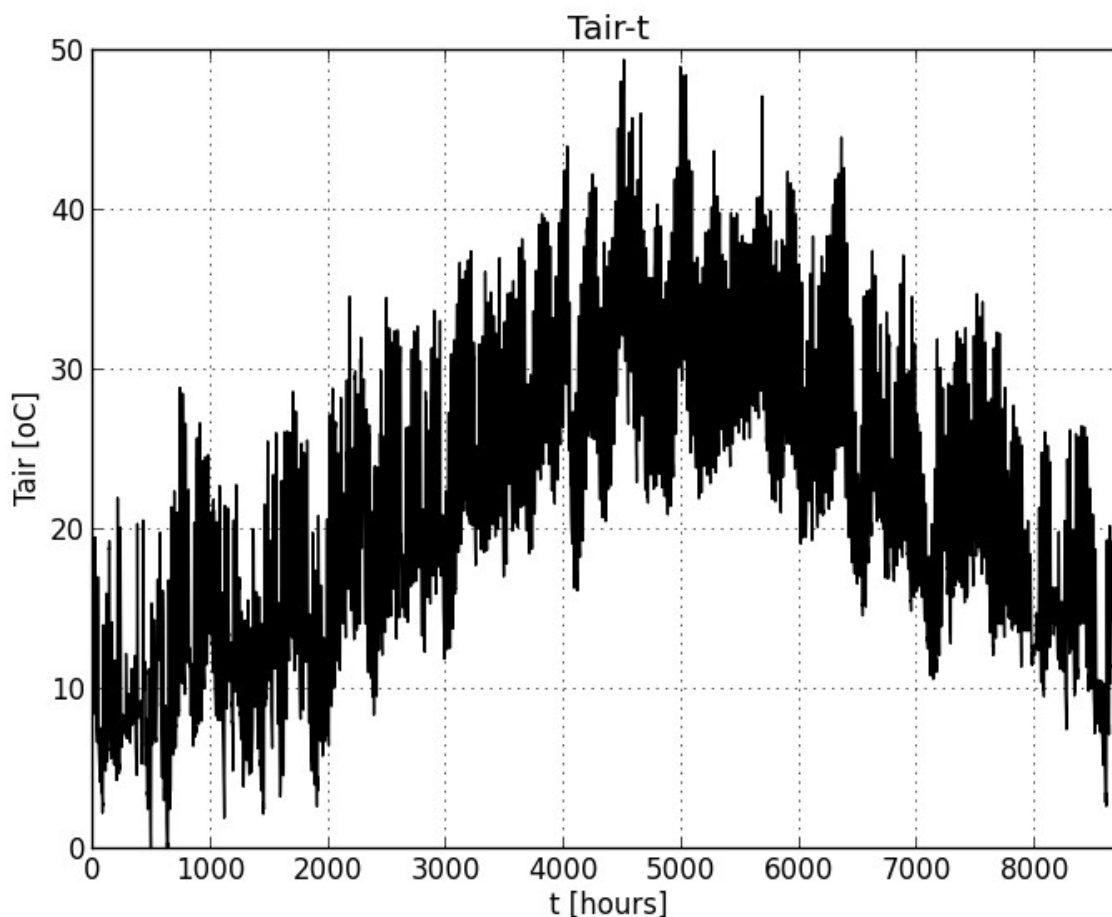
Β-ΥΠΟΛΟΓΙΣΤΙΚΟ ΜΕΡΟΣ

Β.1-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΑΕΡΑ [16]

Οι απαιτούμενες ωριαίες τιμές της θερμοκρασίας αέρα ανακτήθηκαν από το ιστορικό δεδομένων του μετεωρολογικού σταθμού του Ε.Μ.Π. Τα στοιχεία αυτά, καθώς και πολλά άλλα (ταχύτητα ανέμου, υγρασία κ.τ.λ.) είναι προσβάσιμα από την σελίδα: <http://openmeteo.org/stations/d/1334/> [16]. Ο κώδικας `meteorological_data.py` (Γ.1), επιλέγοντας αρχική ημερομηνία και συνολικό αριθμό ημερών, αποδίδει το μητρώο θερμοκρασιών αέρα, T_{air} [$^{\circ}\text{C}$], σε μορφή `tairarray.txt` καθώς και το αντίστοιχο διάγραμμα. Οι τιμές που χρησιμοποιήθηκαν, αφορούν το έτος 2000.

Ως χρονικές περιόδους ενδιαφέροντος θα θεωρήσουμε τις πρώτες τρεις μέρες κάθε μήνα από τον Ιανουάριο έως και τον Αύγουστο. Τα επίπεδα θερμοκρασίας αέρα και ηλιακής ακτινοβολίας υποχωρούν μετά το πέρας του καλοκαιριού (εικόνες Β.1.1 και Β.2.1), γι' αυτό δεν κρίθηκε απαραίτητο να εξεταστούν περίοδοι μετά τον Αύγουστο.

Παρακάτω, παρατίθεται ενδεικτικά το διάγραμμα της θερμοκρασίας αέρα κατά τη διάρκεια του έτους 2000:

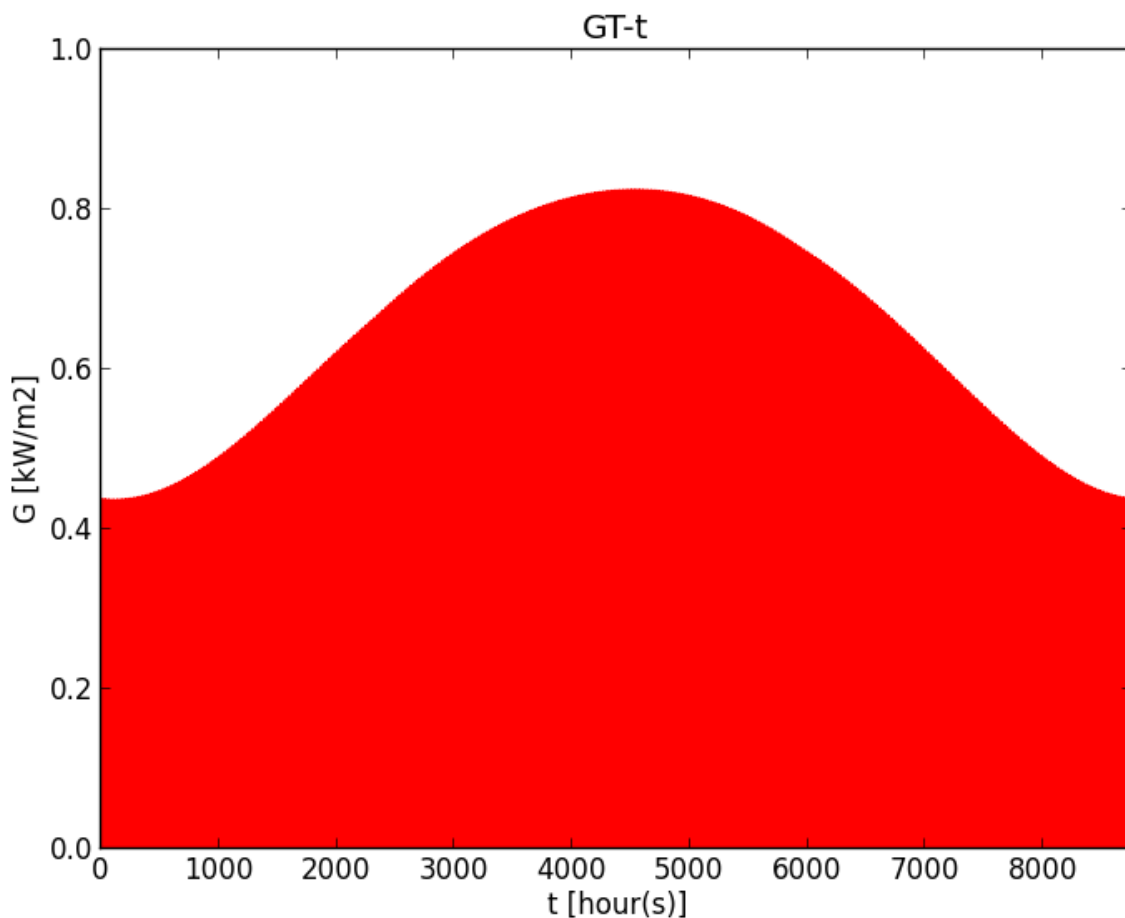


Διάγραμμα Β.1.1: Θερμοκρασία αέρα κατά τη διάρκεια του έτους 2000.

Β.2-ΥΠΟΛΟΓΙΣΜΟΣ ΣΤΙΓΜΙΑΙΑΣ ΟΛΙΚΗΣ ΗΛΙΑΚΗΣ ΑΚΤΙΝΟΒΟΛΙΑΣ ΚΕΚΛΙΜΕΝΟΥ ΕΠΙΠΕΔΟΥ [3]

Η μεθοδολογία υπολογισμού της στιγμιαίας ολικής ηλιακής ακτινοβολίας κεκλιμένου επιπέδου έχει αναλυθεί στο θεωρητικό μέρος (Α.5, [3]). Ο κώδικας `insolation.py`, σε συνδυασμό με το αρχείο συντελεστών `insolation_coefficients.txt` (Γ.2), για δείκτη αρχικής ημέρας (1-365), συνολικό αριθμό ημερών και κλίση επιφάνειας, αποδίδει το μητρώο στιγμιαίας ολικής ακτινοβολίας κεκλιμένου επιπέδου, G_T [kW/m^2], σε μορφή `insolation.txt` καθώς και το αντίστοιχο διάγραμμα. Υπενθυμίζουμε ότι, για μέγιστη ετήσια ενέργεια απαιτείται κλίση ίση με το γεωγραφικό πλάτος, $\beta = \varphi = 37.97^\circ$ [3].

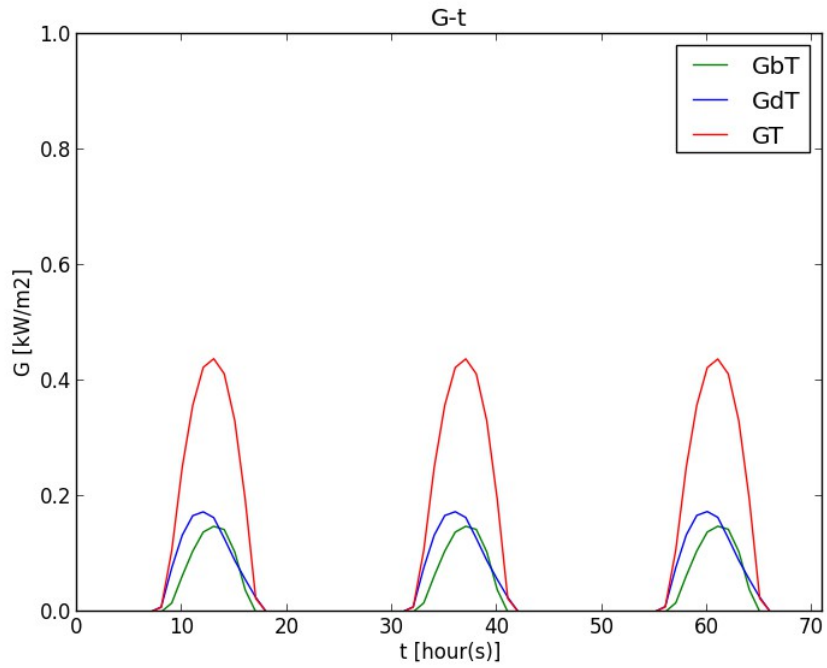
Παρακάτω, παρατίθεται ενδεικτικά το διάγραμμα της στιγμιαίας ολικής ηλιακής ακτινοβολίας κατά τη διάρκεια ενός έτους:



Διάγραμμα Β.2.1: Ηλιακή ακτινοβολία κατά τη διάρκεια ενός έτους για επιφάνεια κλίσης $\beta = 37.97^\circ$ (γεωγραφικό πλάτος Αθήνας).

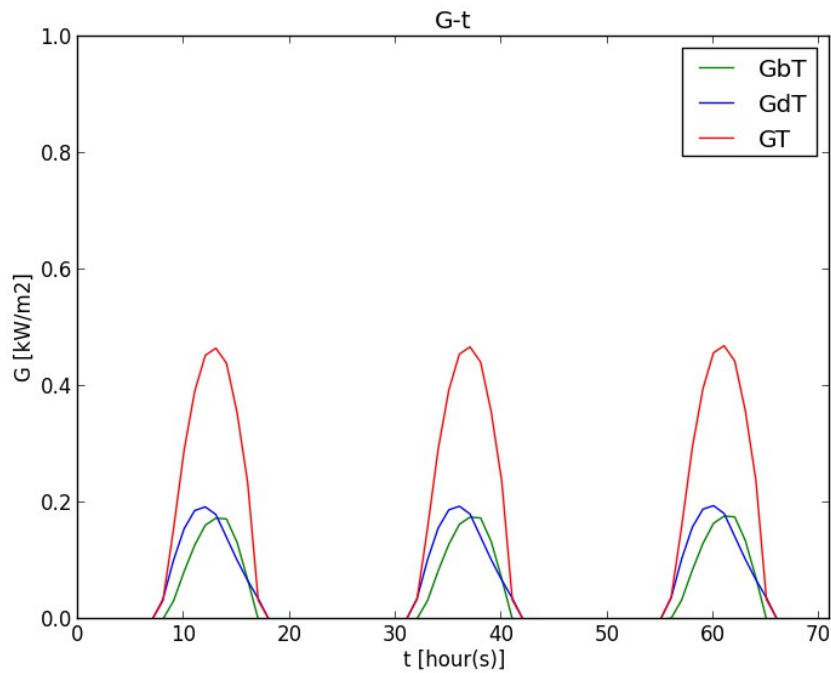
Στη συνέχεια ακολουθούν τα διαγράμματα στιγμιαίας άμεσης, διάχυτης και ολικής ακτινοβολίας κεκλιμένου επιπέδου (G_{bT} , G_{dT} και G_T αντίστοιχα) κατά τις περιόδους ενδιαφέροντος (τρεις πρώτες μέρες κάθε μήνα, από τον Ιανουάριο ως και τον Αύγουστο).

Περίοδος:1-3/1



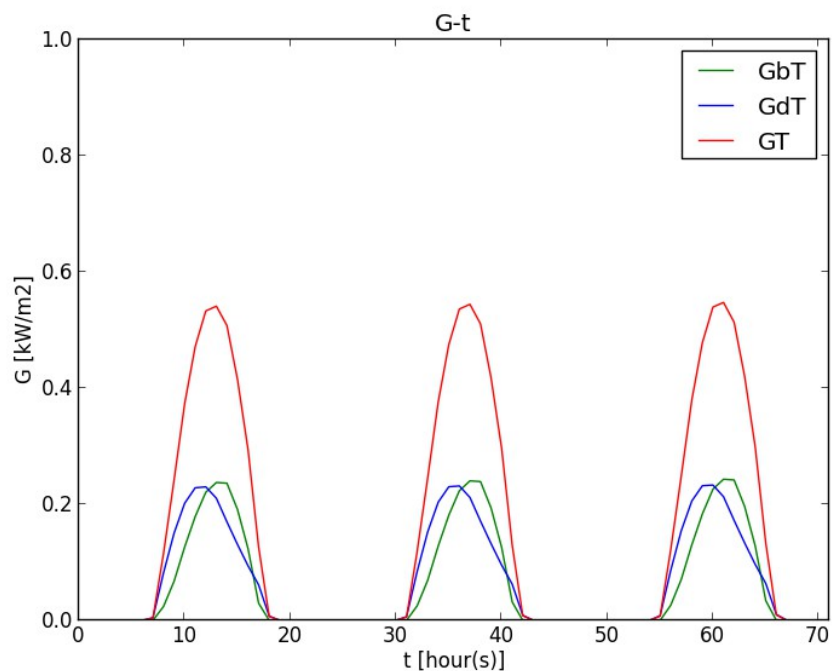
Διάγραμμα Β.2.2:Στιγμιαία ηλιακή ακτινοβολία (άμεση,διάχυτη και ολική), 1-3/1.

Περίοδος:1-3/2



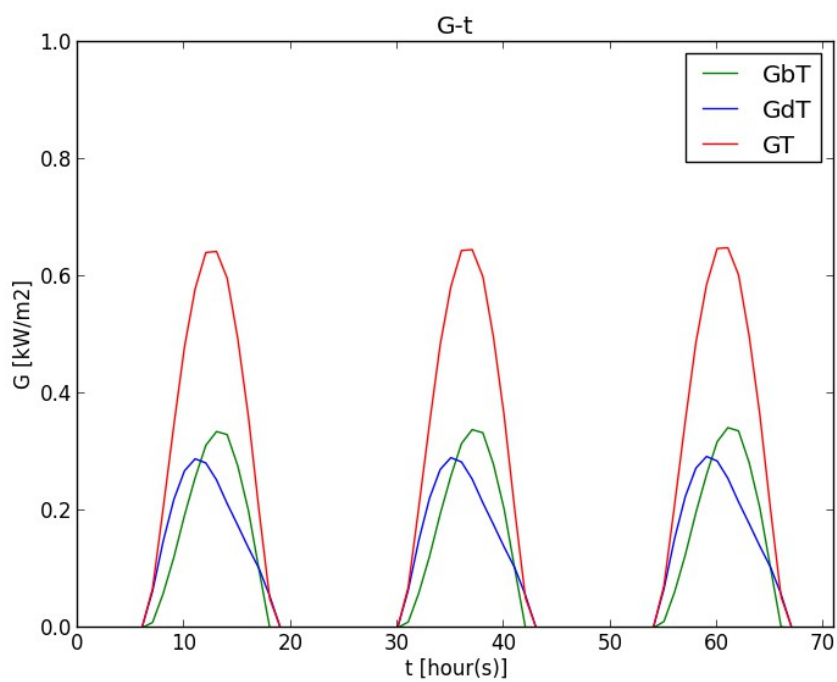
Διάγραμμα Β.2.3:Στιγμιαία ηλιακή ακτινοβολία (άμεση,διάχυτη και ολική), 1-3/2.

Περίοδος:1-3/3



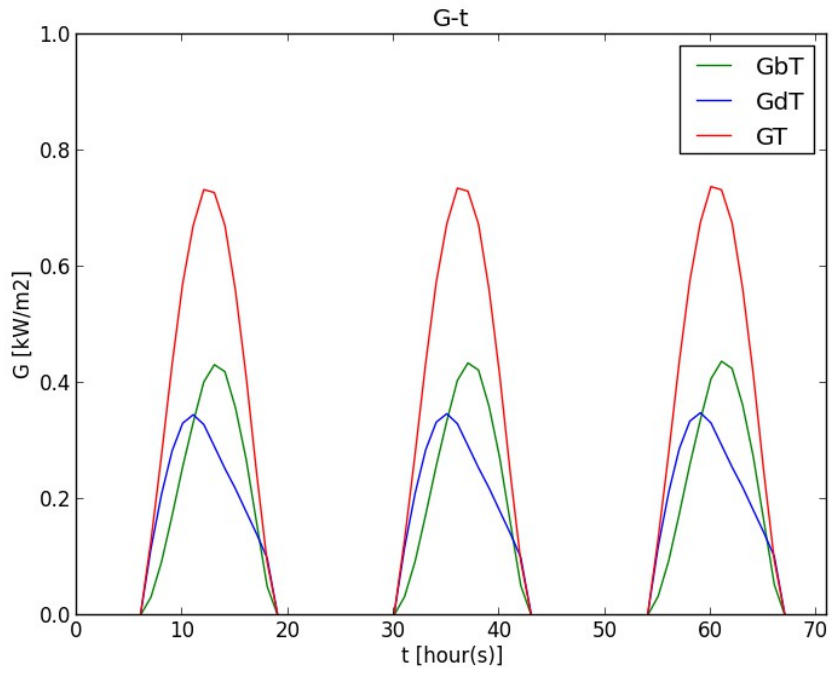
Διάγραμμα Β.2.4:Στιγμαία ηλιακή ακτινοβολία (άμεση,διάχυτη και ολική), 1-3/3.

Περίοδος:1-3/4



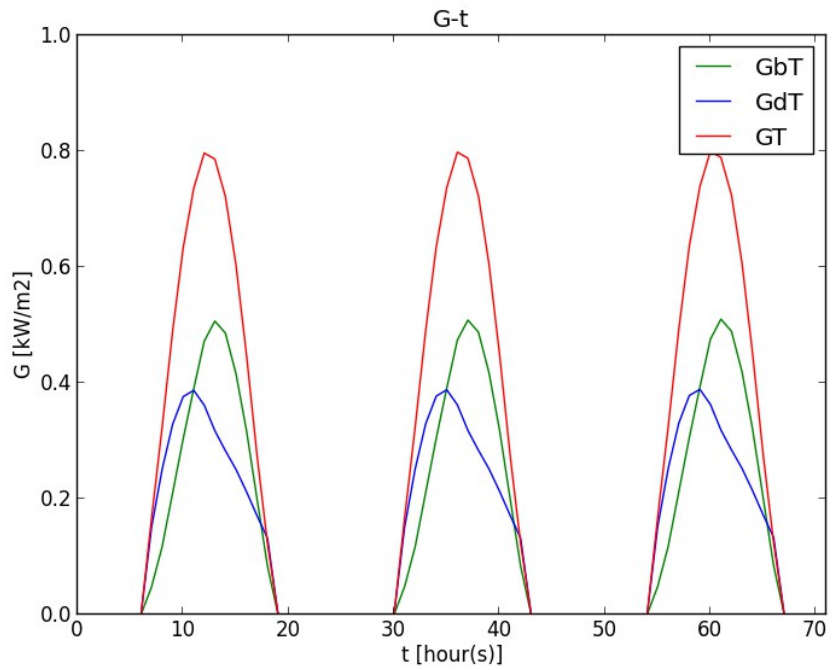
Διάγραμμα Β.2.5:Στιγμαία ηλιακή ακτινοβολία (άμεση,διάχυτη και ολική), 1-3/4.

Περίοδος:1-3/5



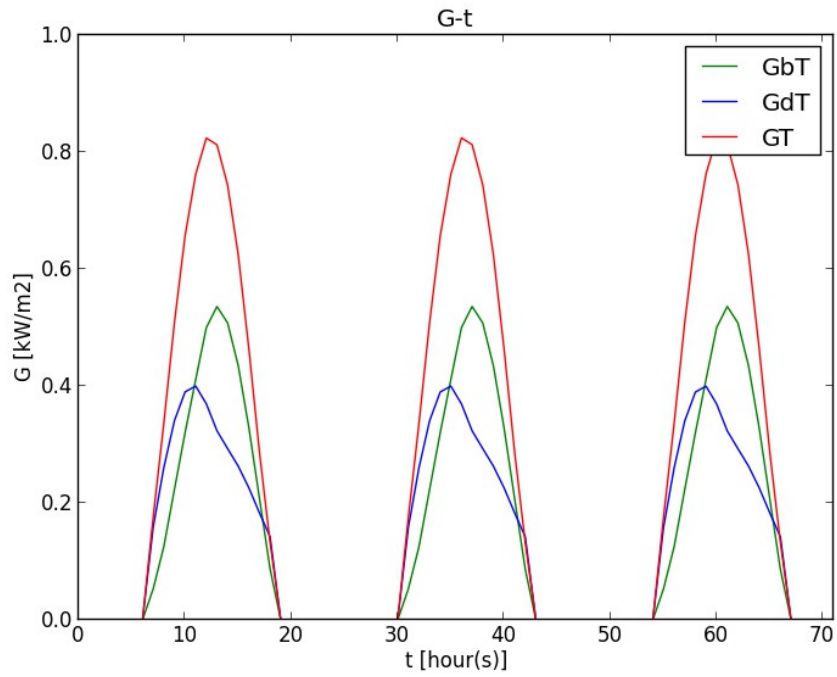
Διάγραμμα Β.2.6:Στιγμιαία ηλιακή ακτινοβολία (άμεση,διάχυτη και ολική), 1-3/5

Περίοδος:1-3/6



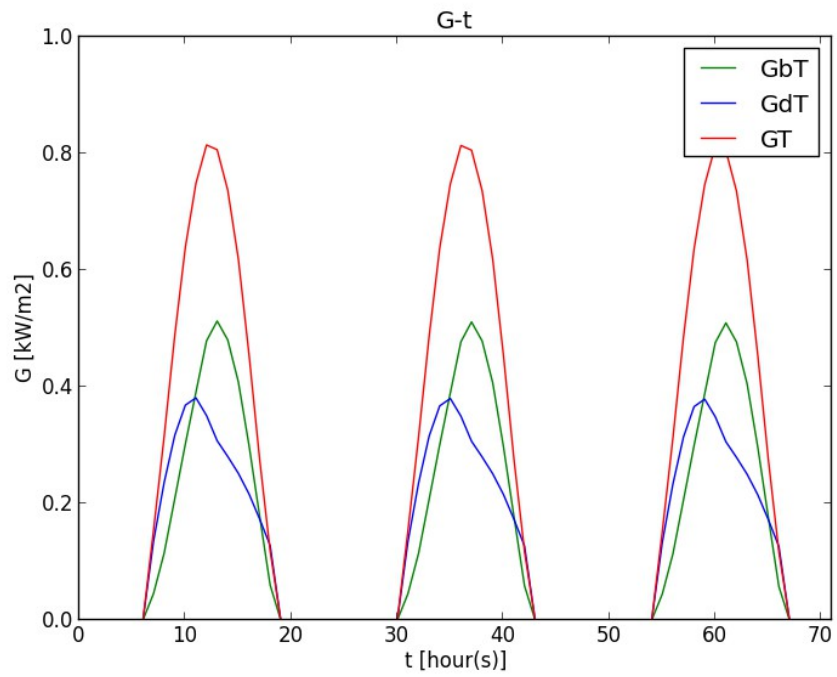
Διάγραμμα Β.2.7:Στιγμιαία ηλιακή ακτινοβολία (άμεση,διάχυτη και ολική), 1-3/6.

Περίοδος:1-3/7



Διάγραμμα Β.2.8:Στιγμαία ηλιακή ακτινοβολία (άμεση,διάχυτη και ολική), 1-3/7.

Περίοδος:1-3/8

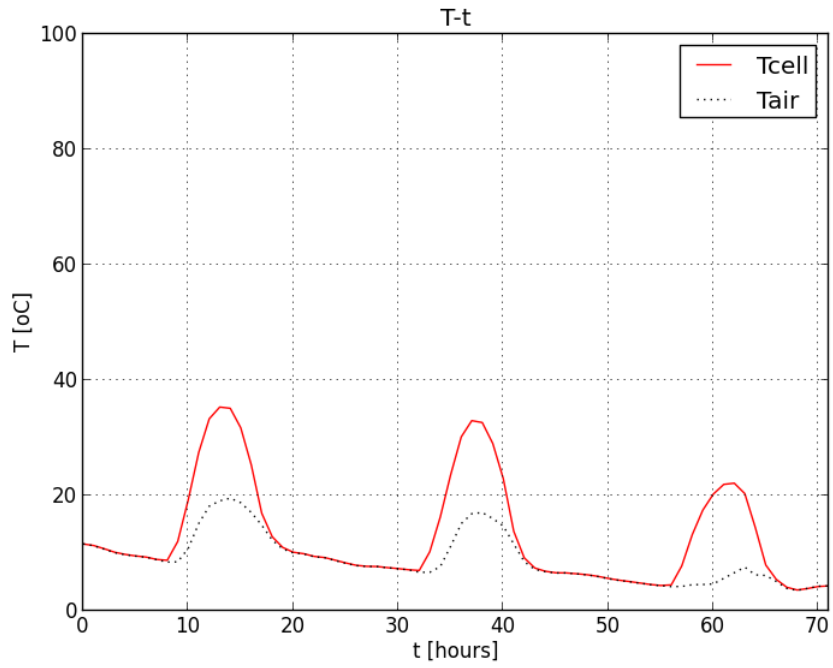


Διάγραμμα Β.2.9:Στιγμαία ηλιακή ακτινοβολία (άμεση,διάχυτη και ολική), 1-3/8.

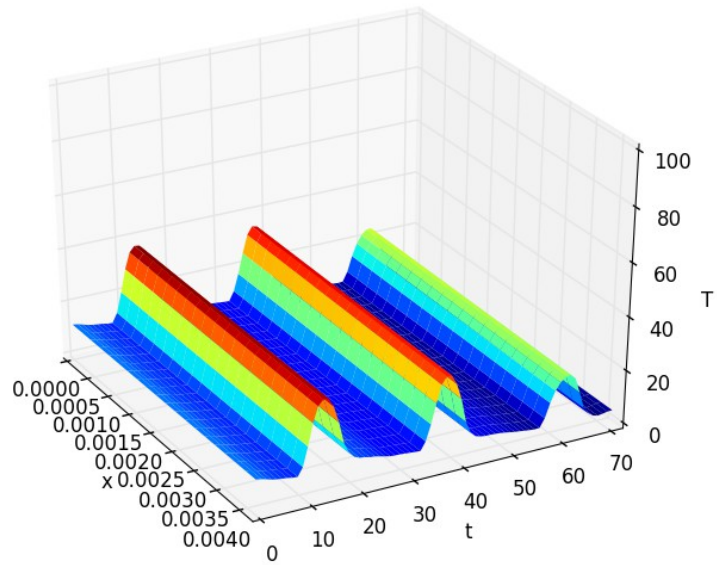
B.3-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ

Ο κώδικας `implicit_pn.py` (Γ.3), ο οποίος με τη σειρά του κάνει χρήση της ρουτίνας επίλυσης τριδιαγώνιων συστημάτων `thomas_algorithm.py` (Γ.8.1), αποδίδει τα παρακάτω διαγράμματα θερμοκρασίας κυψέλης της φωτοβολταϊκής μονάδας για τις χρονικές περιόδους επιλογής. Τα αποτελέσματα αποθηκεύονται και ως αρχεία `tcellarray_pn.txt`. Να σημειωθεί ότι ο κώδικας δέχεται ως είσοδο δεδομένων (`input`) τα αρχεία θερμοκρασίας αέρα (`tairarray.txt`, Γ.1) και ηλιακής ακτινοβολίας (`insolation.txt`, Γ.2).

Περίοδος:1-3/1

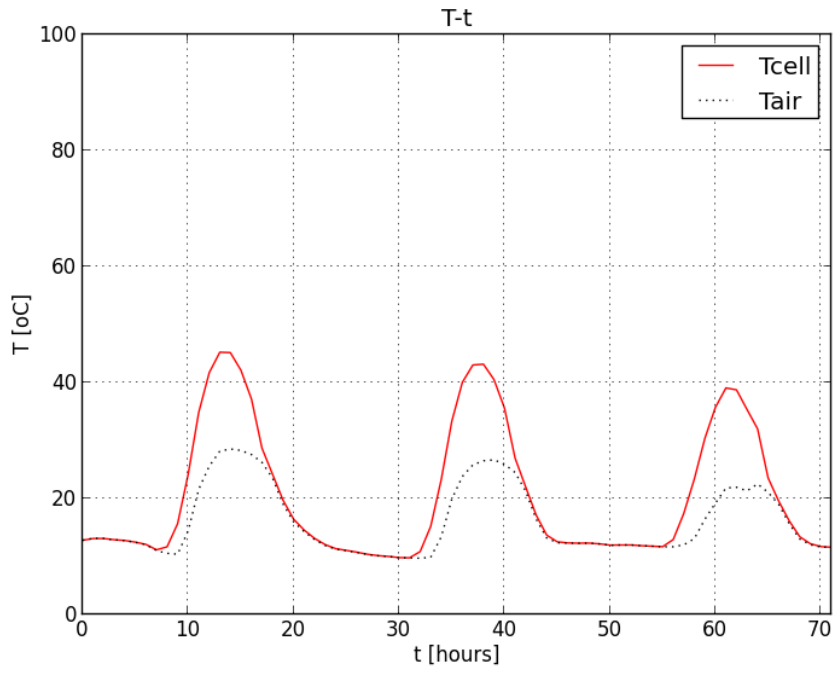


Διάγραμμα Β.3.1: Θερμοκρασία κυψέλης, PV, 1-3/1.

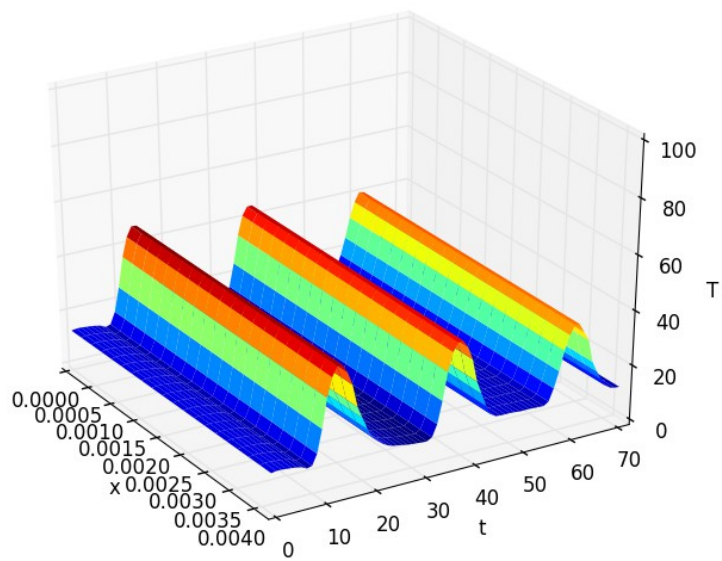


Διάγραμμα Β.3.2: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV, 1-3/1.

Περίοδος:1-3/2

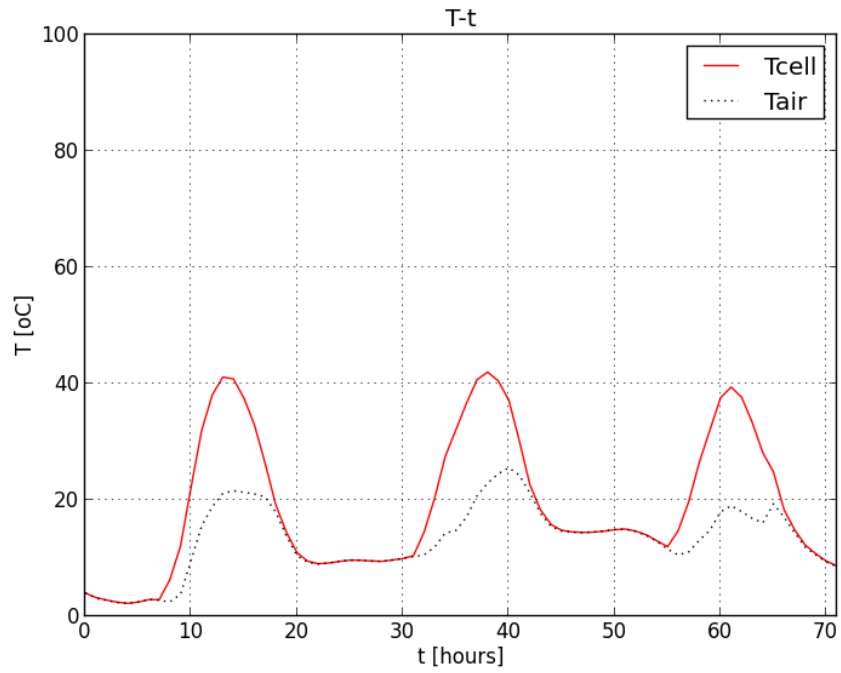


Διάγραμμα Β.3.3: Θερμοκρασία κυψέλης, PV, 1-3/2.

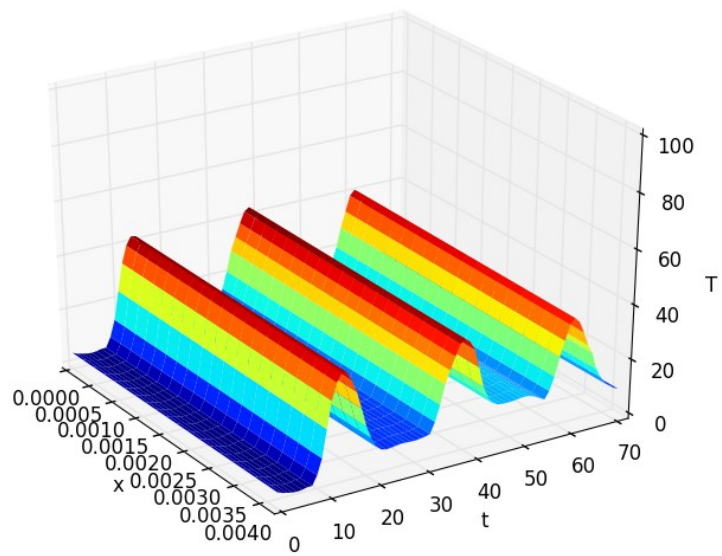


Διάγραμμα Β.3.4: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV, 1-3/2.

Περίοδος:1-3/3

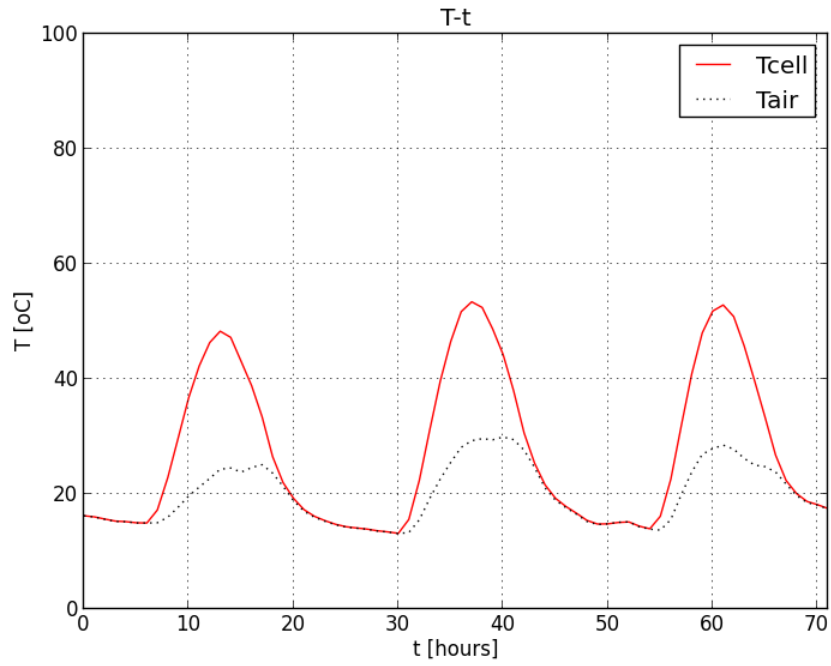


Διάγραμμα Β.3.5: Θερμοκρασία κυψέλης, PV, 1-3/3.

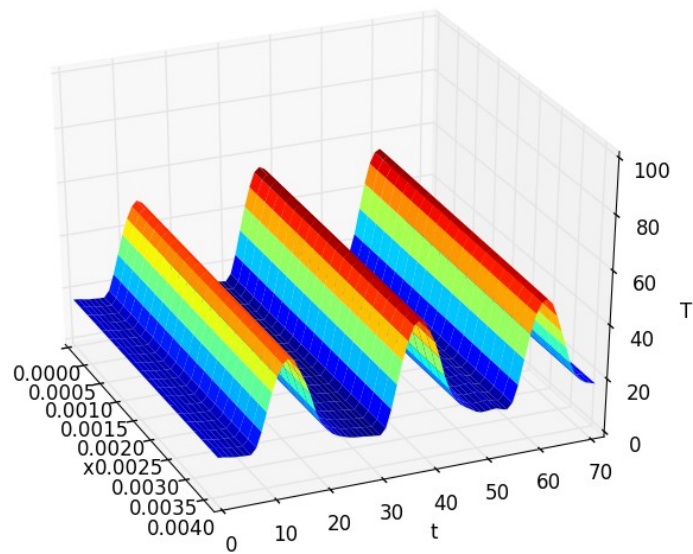


Διάγραμμα Β.3.6: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV, 1-3/3.

Περίοδος:1-3/4

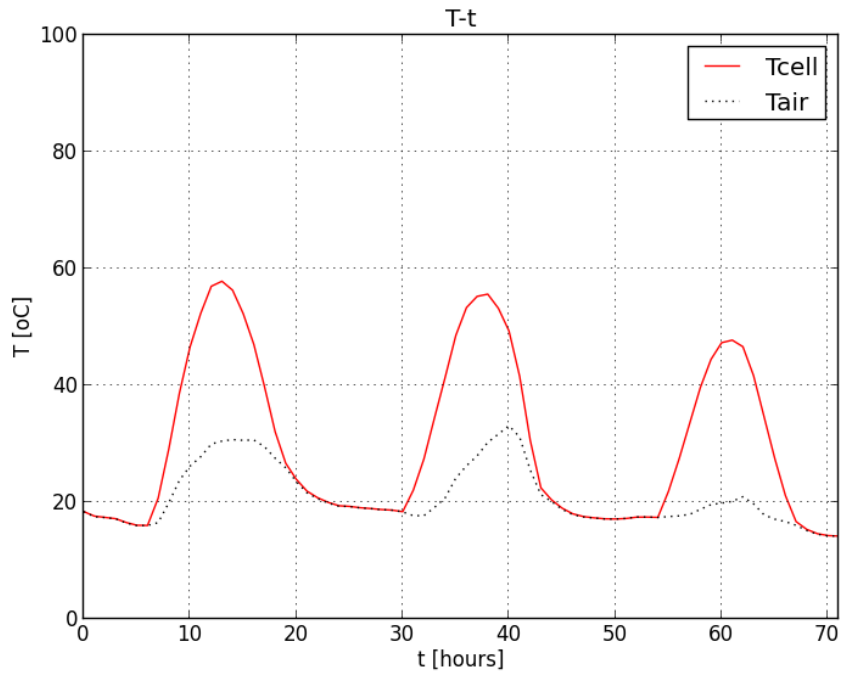


Διάγραμμα Β.3.7: Θερμοκρασία κυψέλης, PV, 1-3/4.

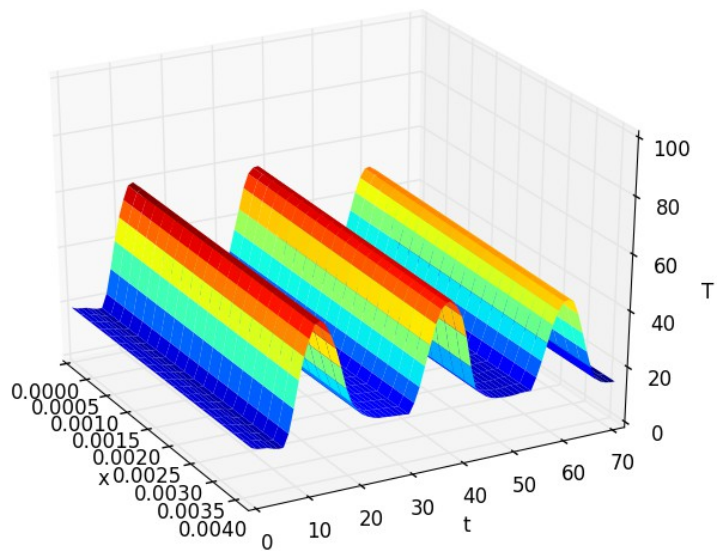


Διάγραμμα Β.3.8: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV, 1-3/4.

Περίοδος:1-3/5

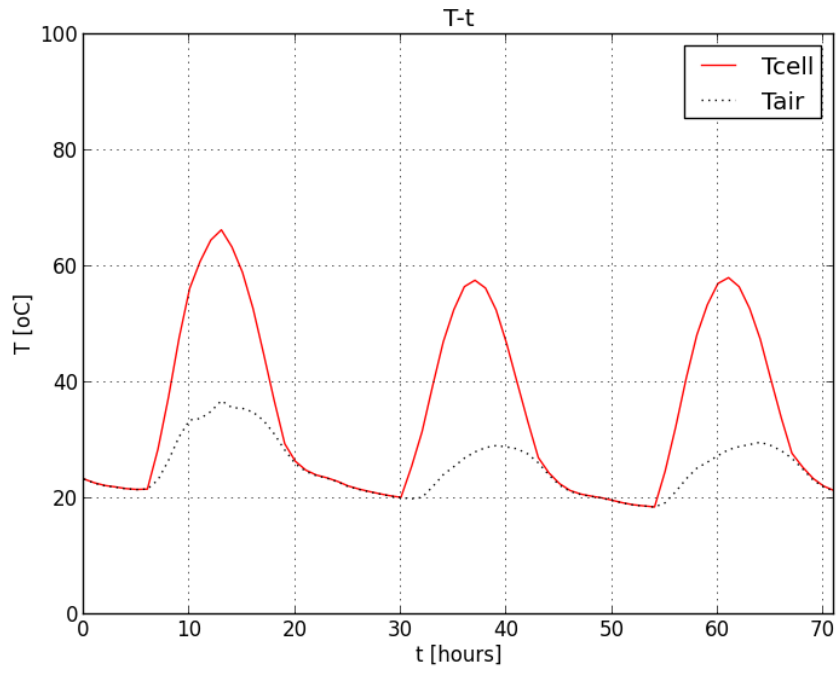


Διάγραμμα Β.3.9: Θερμοκρασία κυψέλης, PV, 1-3/5.

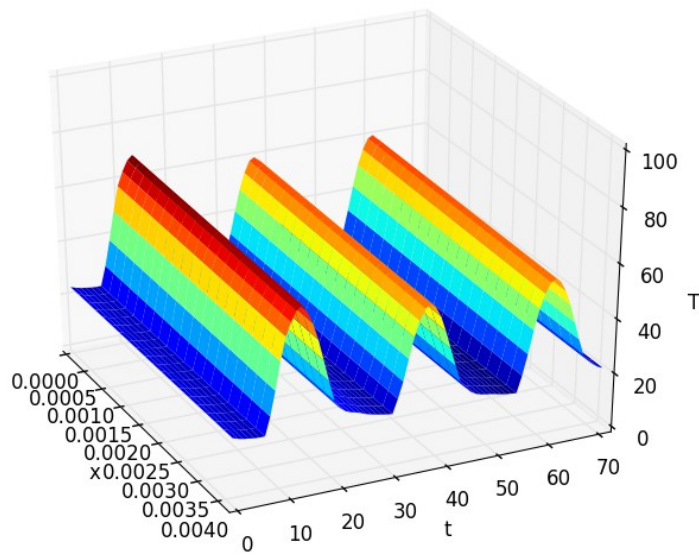


Διάγραμμα Β.3.10: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV, 1-3/5.

Περίοδος:1-3/6

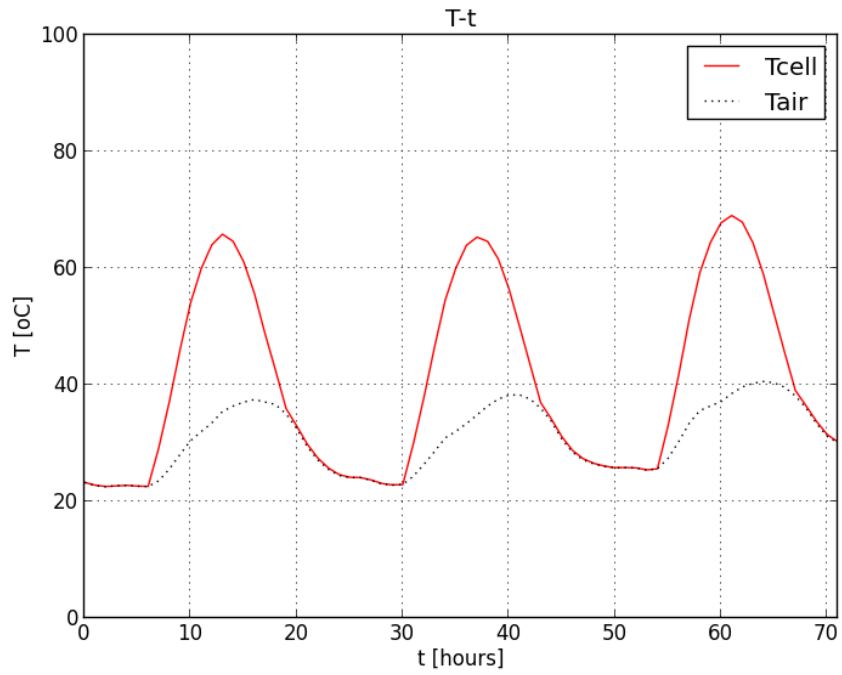


Διάγραμμα Β.3.11: Θερμοκρασία κυψέλης, PV, 1-3/6.

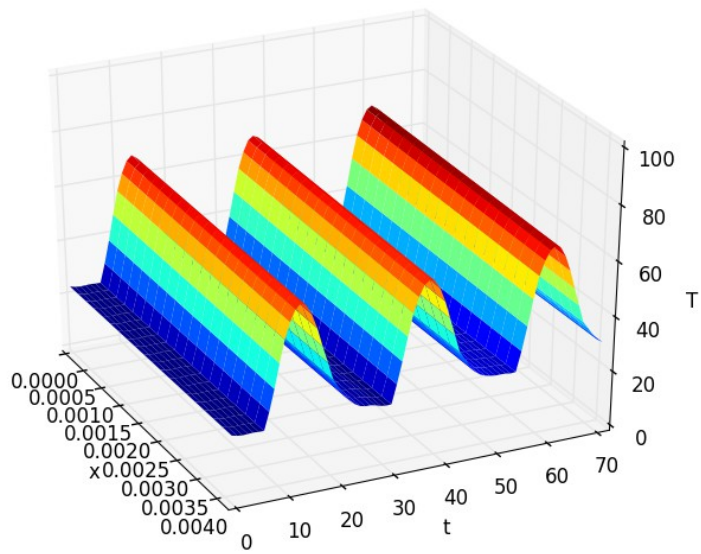


Διάγραμμα Β.3.12: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV, 1-3/6.

Περίοδος:1-3/7

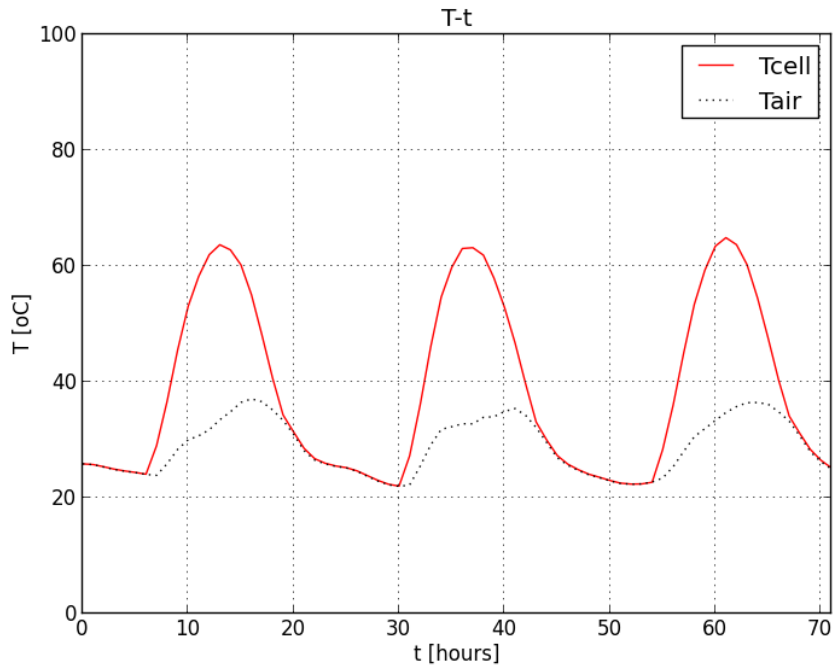


Διάγραμμα Β.3.13: Θερμοκρασία κυψέλης, PV, 1-3/7.

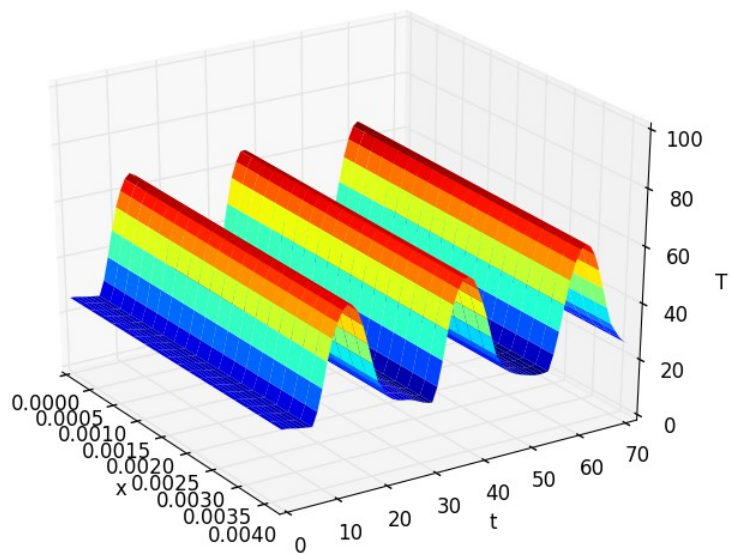


Διάγραμμα Β.3.14: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV, 1-3/7.

Περίοδος:1-3/8



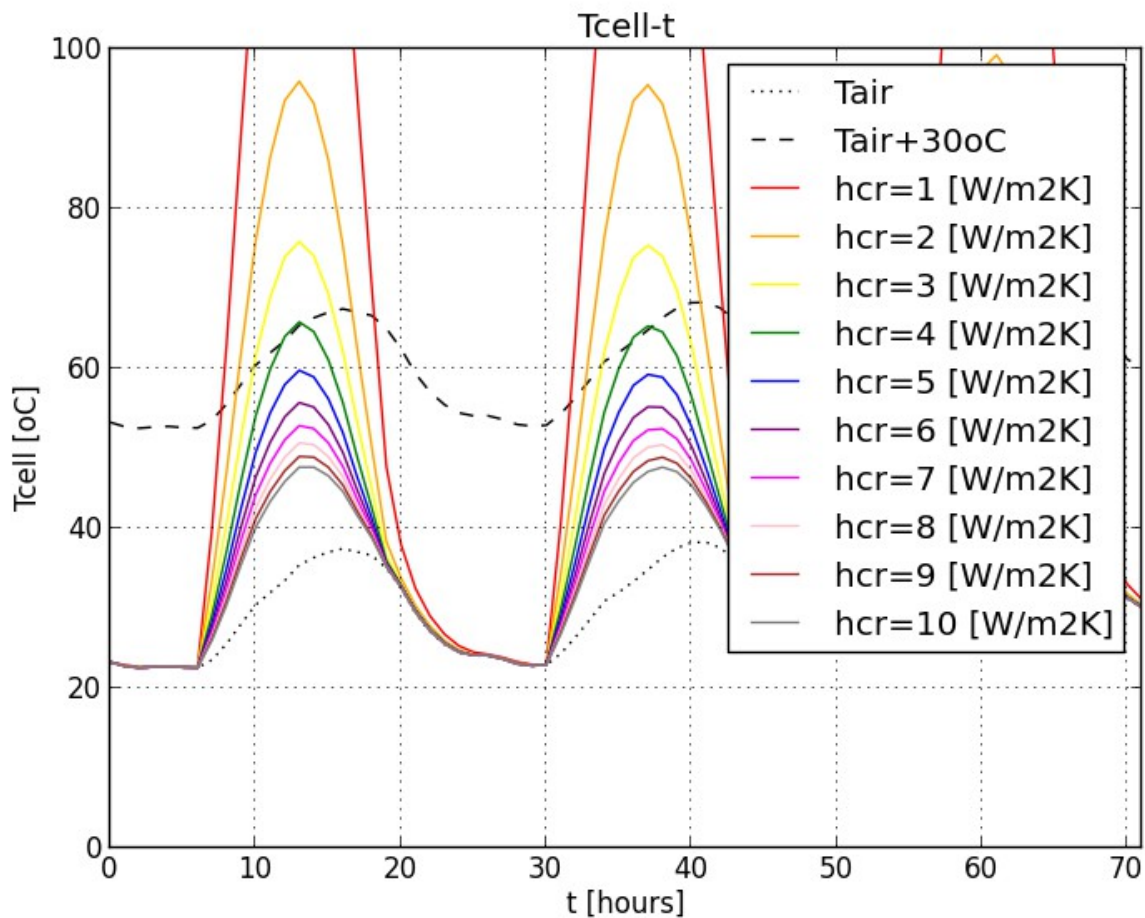
Διάγραμμα Β.3.15: Θερμοκρασία κυψέλης, PV, 1-3/8.



Διάγραμμα Β.3.16: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV, 1-3/8.

Β.4-ΥΠΟΛΟΓΙΣΜΟΣ ΣΥΝΤΕΛΕΣΤΗ ΜΕΤΑΦΟΡΑΣ ΘΕΡΜΟΤΗΤΑΣ

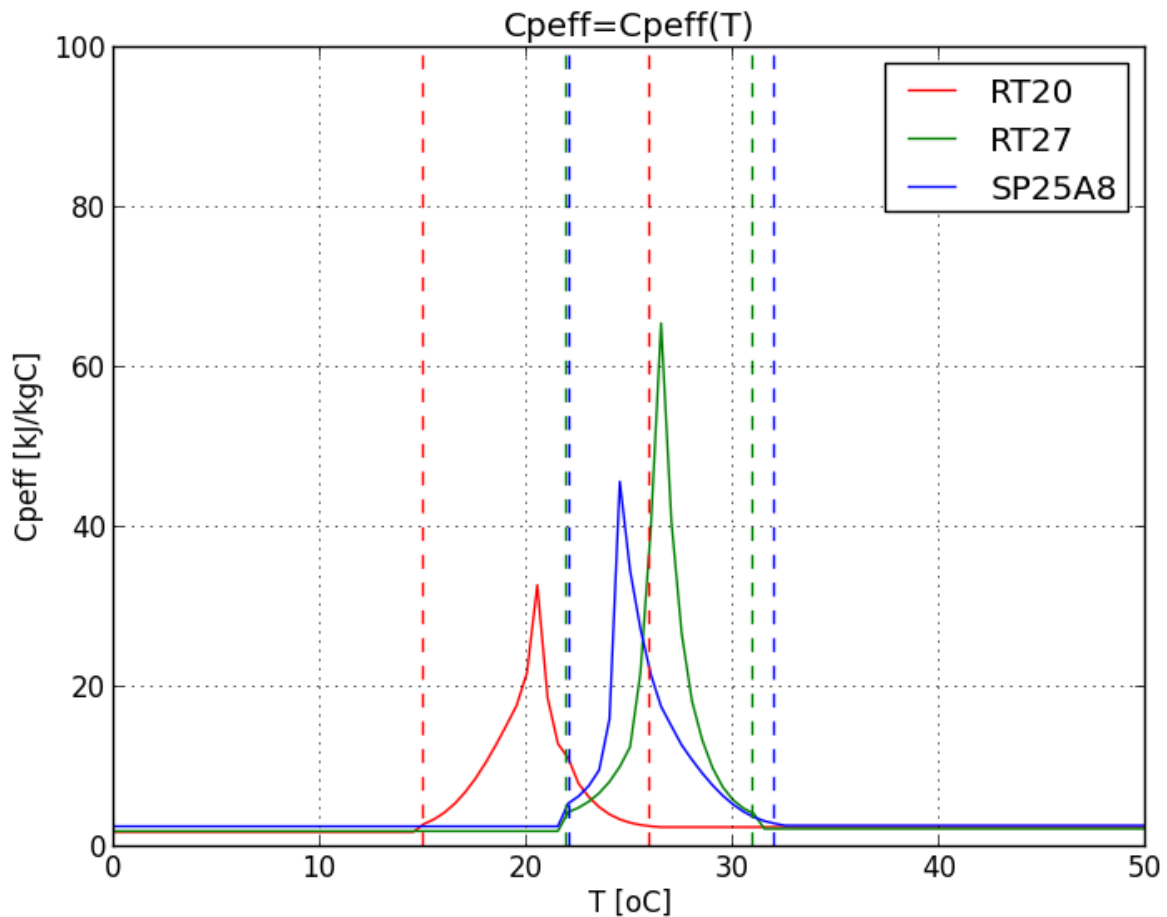
Εκτελώντας τον κώδικα `implicit_pn.py` (Γ.3) για διάφορες τιμές του συνδυαστικού συντελεστή μεταφοράς θερμότητας h (από 1 έως 10 $\text{W/m}^2\text{K}$) παίρνουμε τα αντίστοιχα μητρώα θερμοκρασιών κυψέλης, τα οποία μετονομάζουμε κατάλληλα (`tcellarray_pn_1.txt`, `tcellarray_pn_2.txt`,..., `tcellarray_pn_10.txt`). Από τα παραπάνω μητρώα, ο κώδικας `heat_transfer_coefficient.py` (Γ.4) αποδίδει το παρακάτω συγκριτικό διάγραμμα θερμοκρασιακής κατανομής της κυψέλης με την τιμή h να αποτελεί παράμετρο. Κάτω από αντίξοες συνθήκες οι ηλιακές κυψέλες μπορούν να αναπτύξουν θερμοκρασία έως και 30 °C υψηλότερη από τη θερμοκρασία αέρα (Α.1.2). Για να "επιτύχουμε" θερμοκρασιακή διαφορά 30 °C κατά την περίοδο του θερμότερου Ιουλίου επιλέγουμε $h=h_r+h_c=4 \text{ W/m}^2\text{K}$.



Διάγραμμα Β.4.1: Θερμοκρασία κυψέλης μονάδας pn συναρτήσει του χρόνου, h παράμετρος.

Β.5-ΣΥΓΚΡΙΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΕΝΕΡΓΟΥ ΘΕΡΜΟΧΩΡΗΤΙΚΟΤΗΤΑΣ ΓΙΑ ΤΑ ΥΛΙΚΑ ΑΛΛΑΓΗΣ ΦΑΣΗΣ RT20, RT27, SP25A8

Εκτελώντας τον κώδικα `effective_heat_capacity_comparison_plot.py` (Γ.5), ο οποίος κάνει χρήση των ρουτίνων υπολογισμού της ενεργού θερμοχωρητικότητας `effective_heat_capacity_rt20.py` (Γ.8.2), `effective_heat_capacity_rt27.py` (Γ.8.3), `effective_heat_capacity_sp25a8.py` (Γ.8.4), παίρνουμε το παρακάτω διάγραμμα της ενεργού θερμοχωρητικότητας C_{peff} συναρτήσει της θερμοκρασίας για κάθε υλικό.



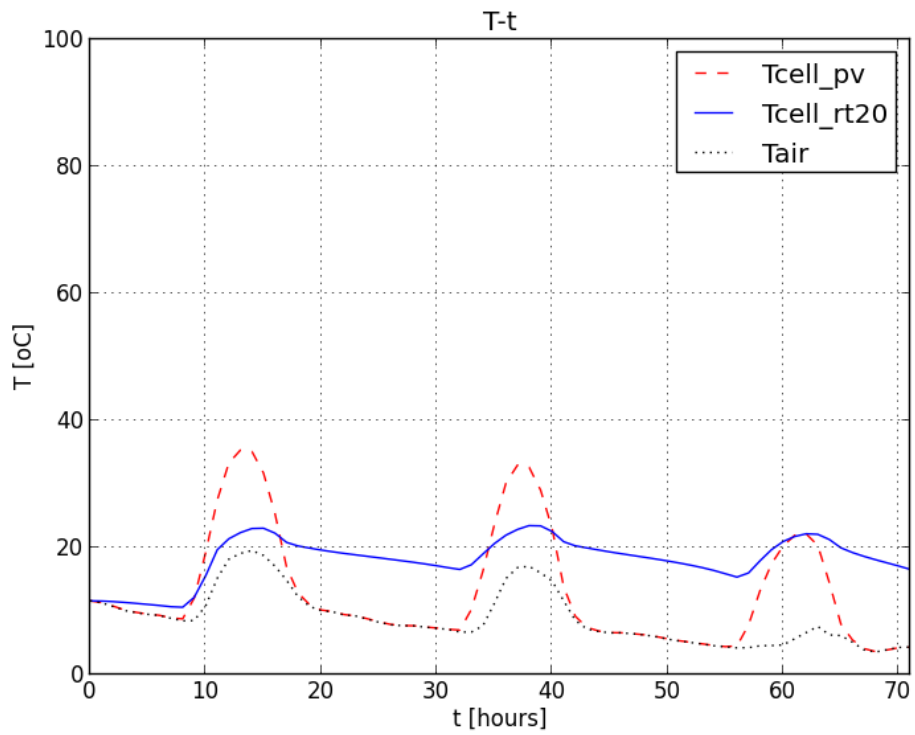
Διάγραμμα Β.5.1: Διάγραμμα ενεργού θερμοχωρητικότητας C_{peff} συναρτήσει της θερμοκρασίας.

B.6-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-ΥΛΙΚΟΥ ΑΛΛΑΓΗΣ ΦΑΣΗΣ

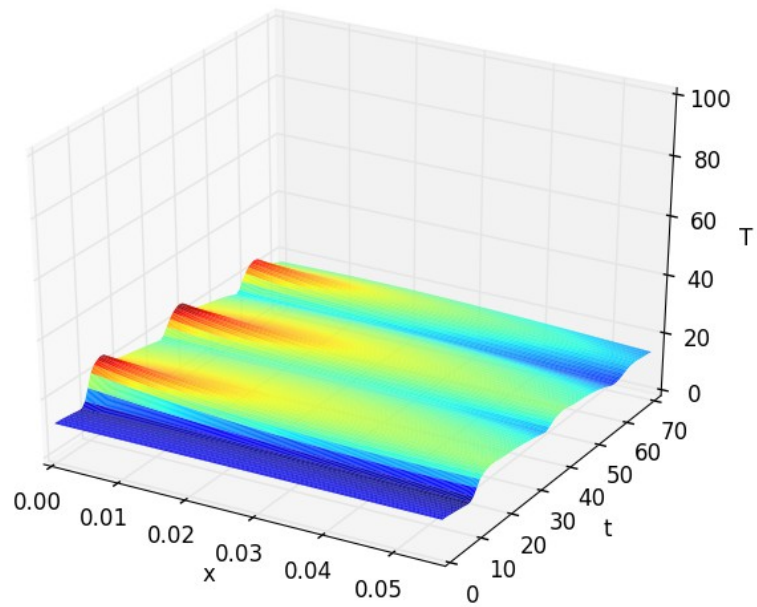
B.6.1-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-RT20

Ο κώδικας `implicit_pn_rt20.py` (Γ.6.1), για μητρώα θερμοκρασίας αέρα (`tairarray.txt`) και ηλιακής ακτινοβολίας (`insolation.txt`) που αντιστοιχούν σε συγκεκριμένη περίοδο, σε συνδυασμό με τις ρουτίνες επίλυσης τριδιαγώνιων συστημάτων (`thomas_algorithm.py`, Γ.8.1) και υπολογισμού της ενεργού θερμοχωρητικότητας (`effective_heat_capacity_rt20.py`, Γ.8.2), αποδίδει τα παρακάτω διαγράμματα θερμοκρασίας κυψέλης συναρτήσει του χρόνου καθώς και τα αντίστοιχα μητρώα (`tcellarray_pn_rt20.txt`). Επίσης, μας δίνει και τα διαγράμματα θερμοκρασιών στο εσωτερικό του PCM, τα οποία θα χρησιμοποιηθούν στη συνέχεια (διαγράμματα B.7.1, B.7.5, B.7.9, B.7.13, B.7.17, B.7.21, B.7.25, B.7.29).

Περίοδος:1-3/1

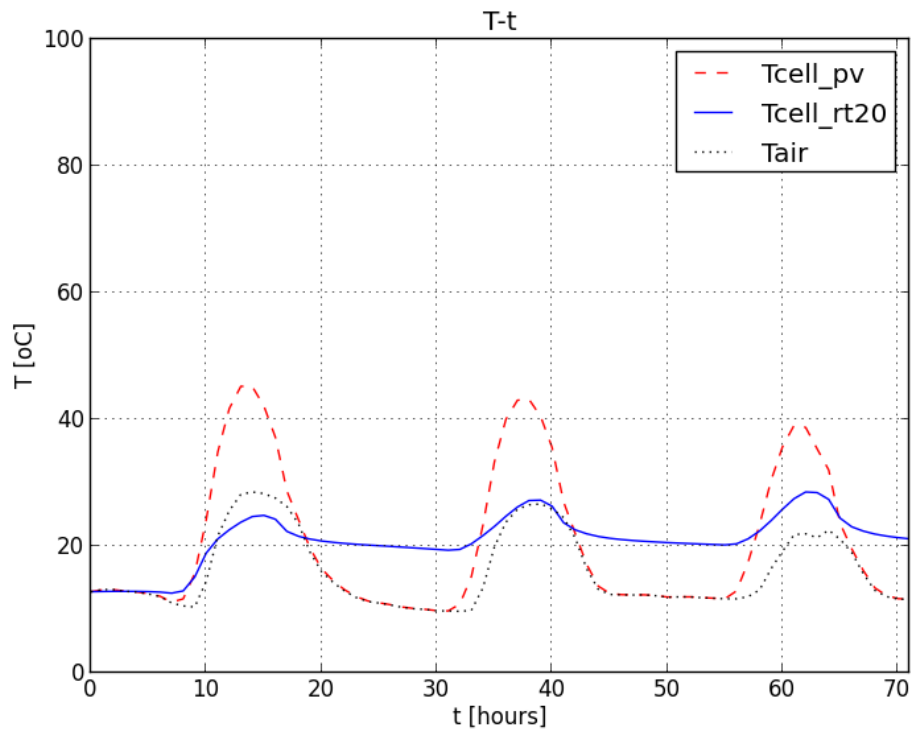


Διάγραμμα Β.6.1.1: Θερμοκρασία κυψέλης, PV-RT20, 1-3/1.

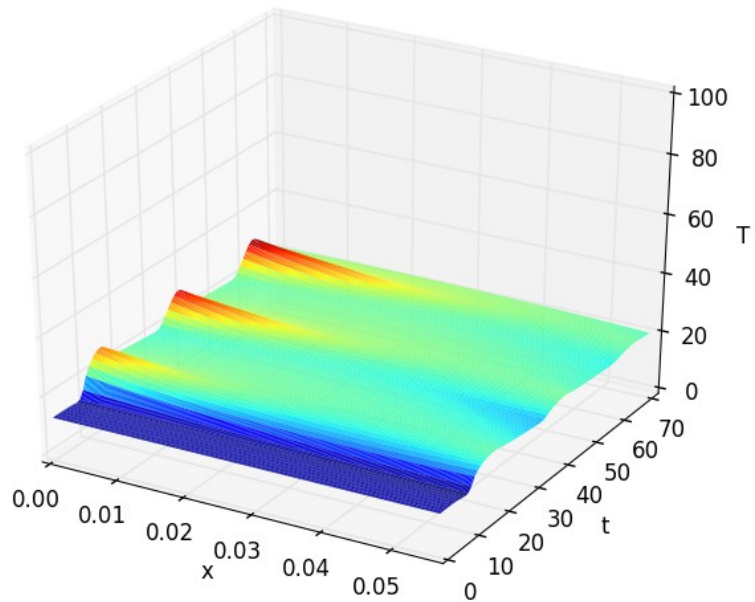


Διάγραμμα Β.6.1.2: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT20, 1-3/1.

Περίοδος:1-3/2

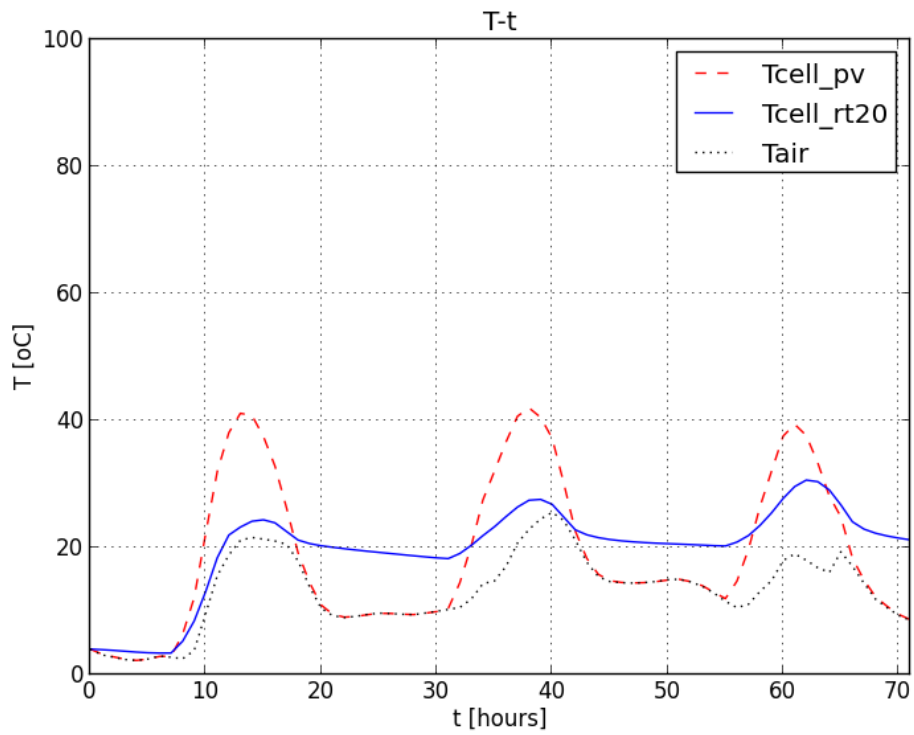


Διάγραμμα Β.6.1.3: Θερμοκρασία κυψέλης, PV-RT20, 1-3/2.

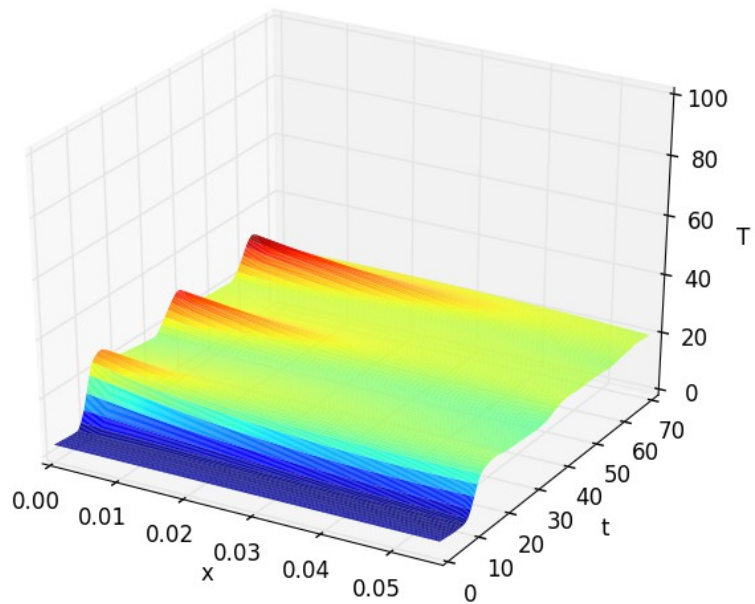


Διάγραμμα Β.6.1.4: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT20, 1-3/2.

Περίοδος:1-3/3

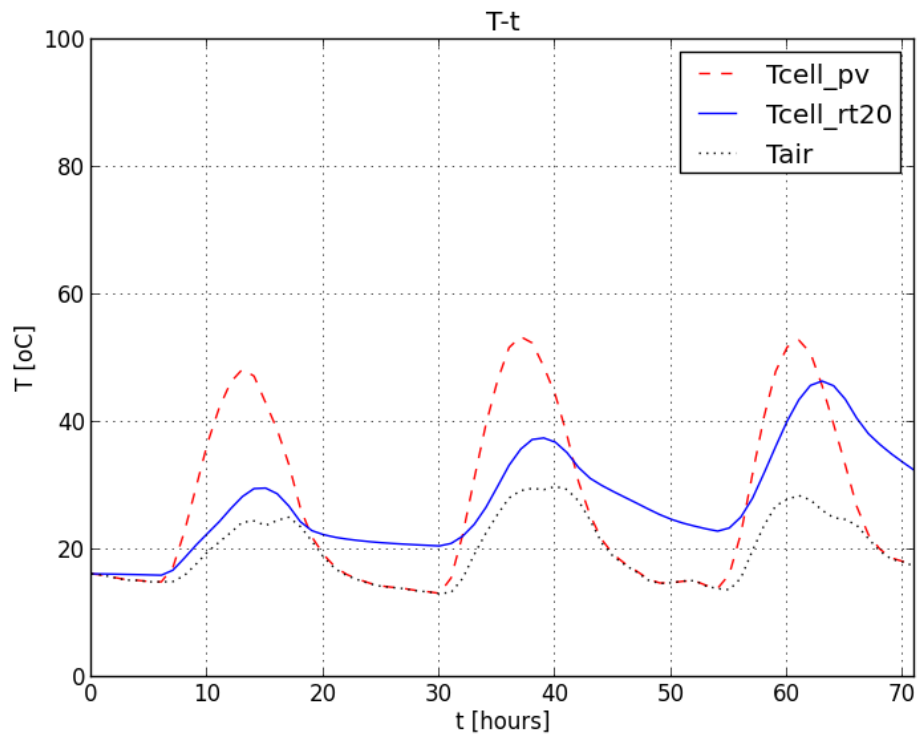


Διάγραμμα Β.6.1.5: Θερμοκρασία κυψέλης, PV-RT20, 1-3/3.

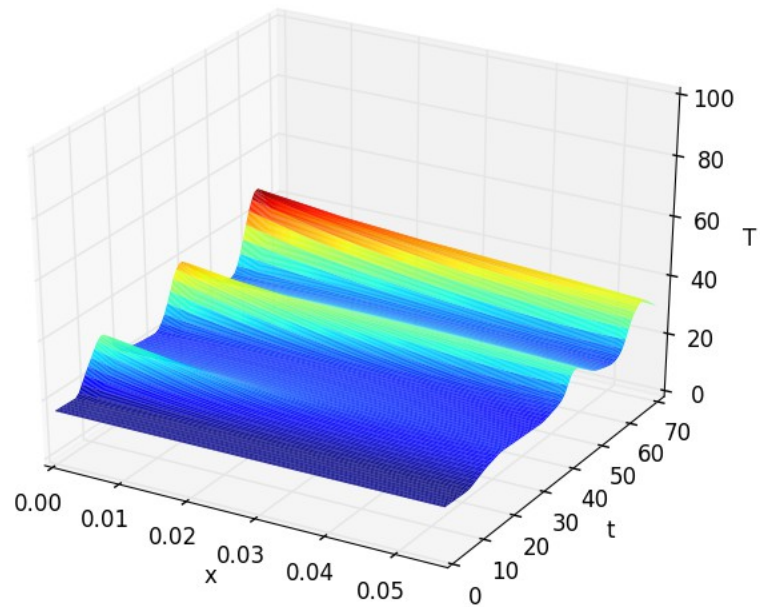


Διάγραμμα Β.6.1.6: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT20, 1-3/3.

Περίοδος:1-3/4

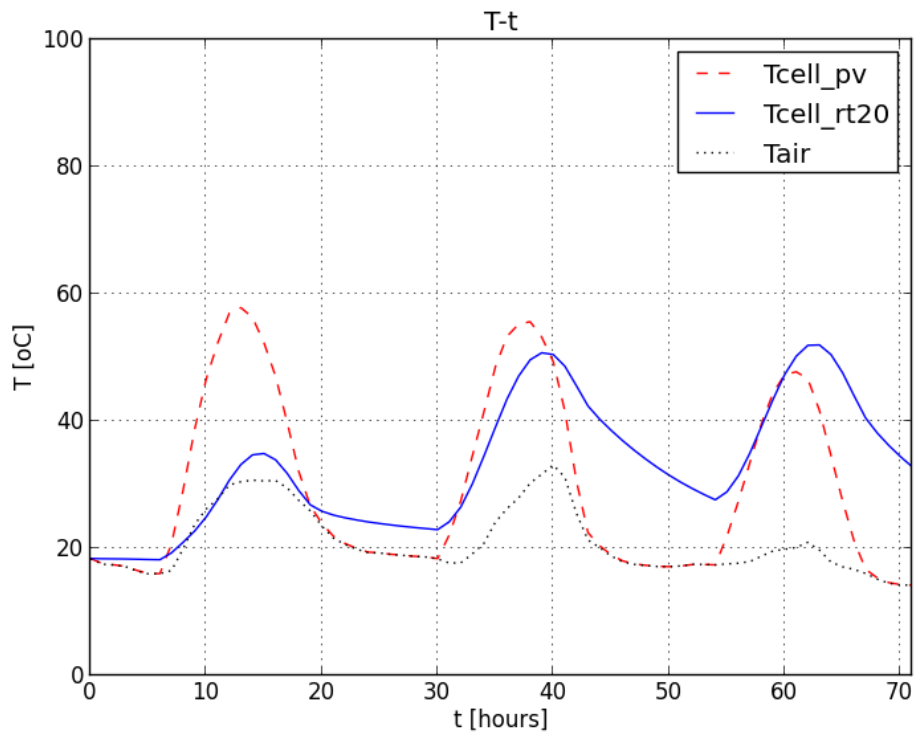


Διάγραμμα Β.6.1.7: Θερμοκρασία κυψέλης, PV-RT20, 1-3/4.

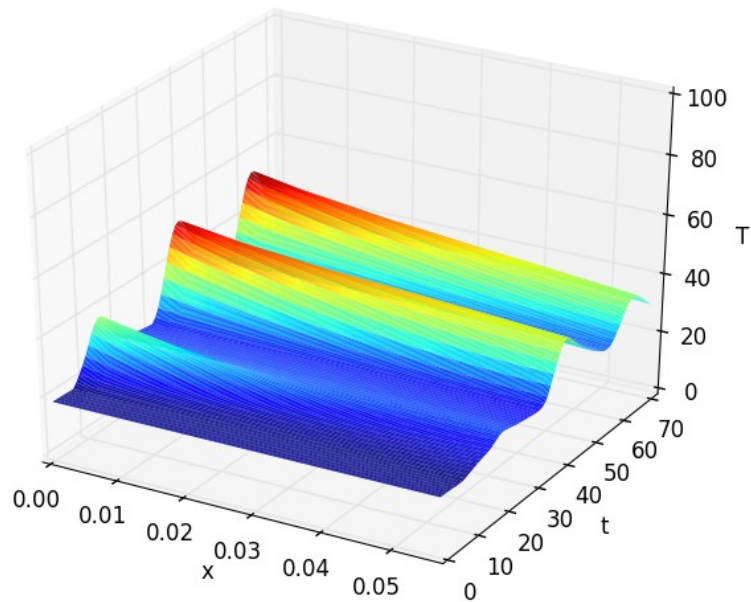


Διάγραμμα Β.6.1.8: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT20, 1-3/4.

Περίοδος:1-3/5

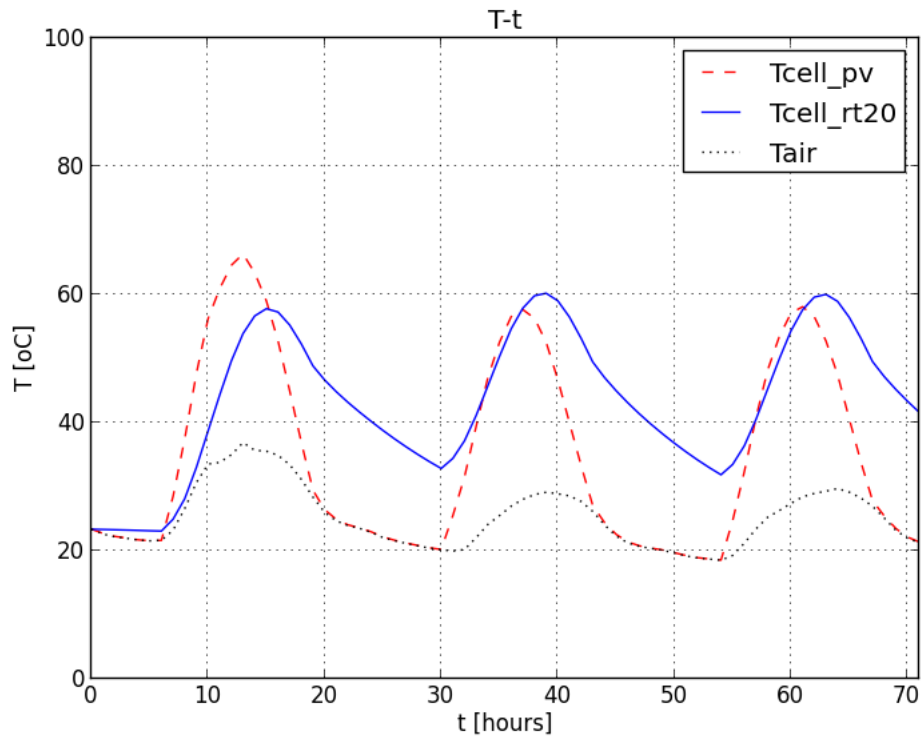


Διάγραμμα Β.6.1.9: Θερμοκρασία κυψέλης, PV-RT20, 1-3/5.

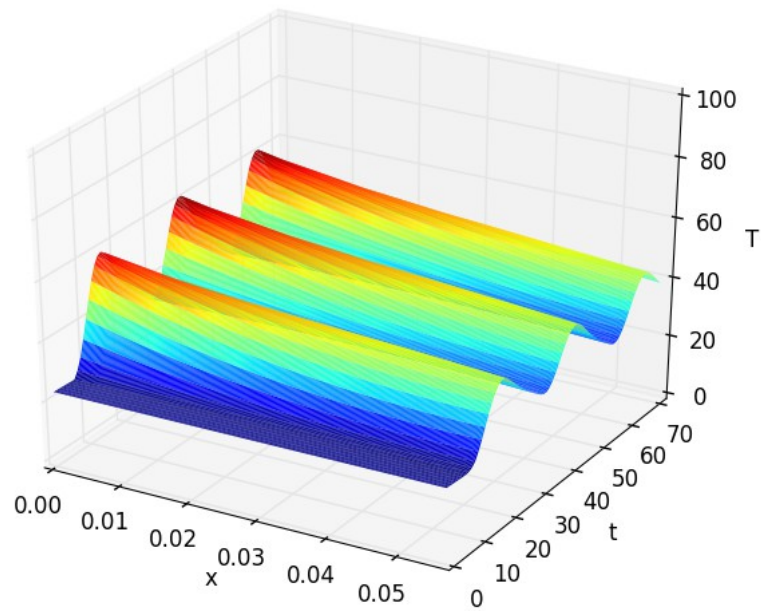


Διάγραμμα Β.6.1.10: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT20, 1-3/5.

Περίοδος:1-3/6

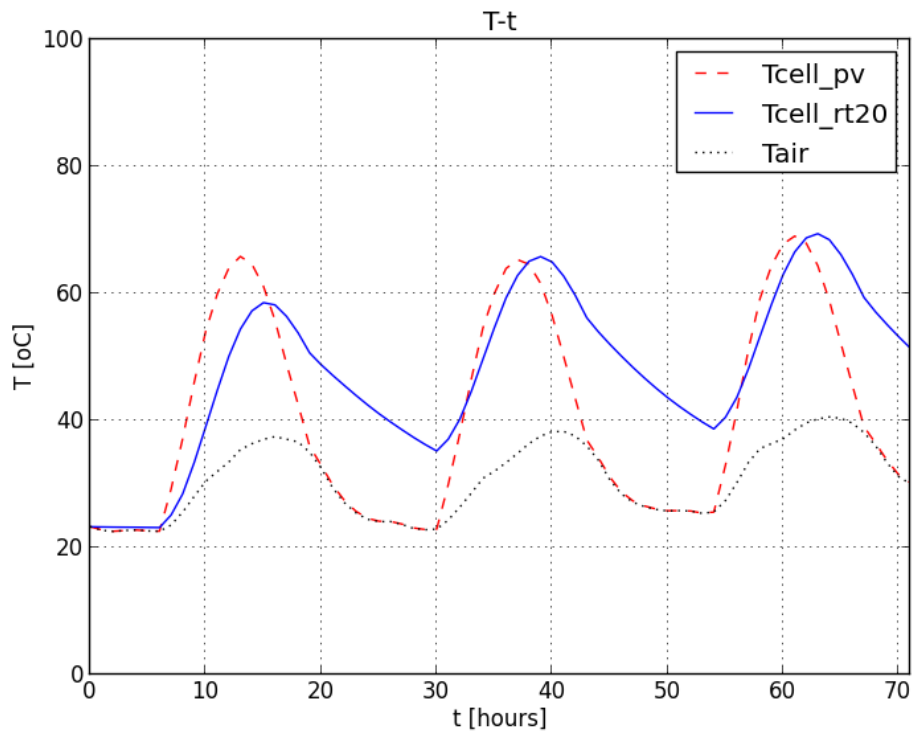


Διάγραμμα Β.6.1.11: Θερμοκρασία κυψέλης, PV-RT20, 1-3/6.

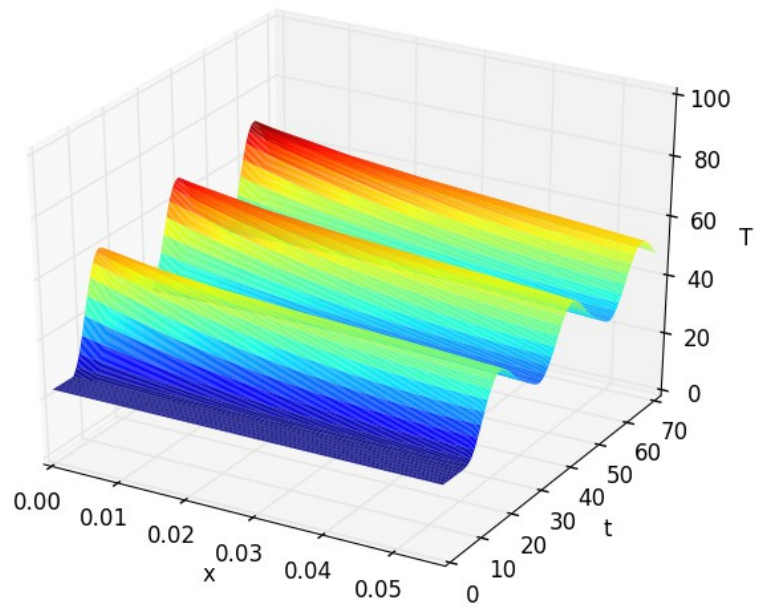


Διάγραμμα Β.6.1.12: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT20, 1-3/6.

Περίοδος:1-3/7

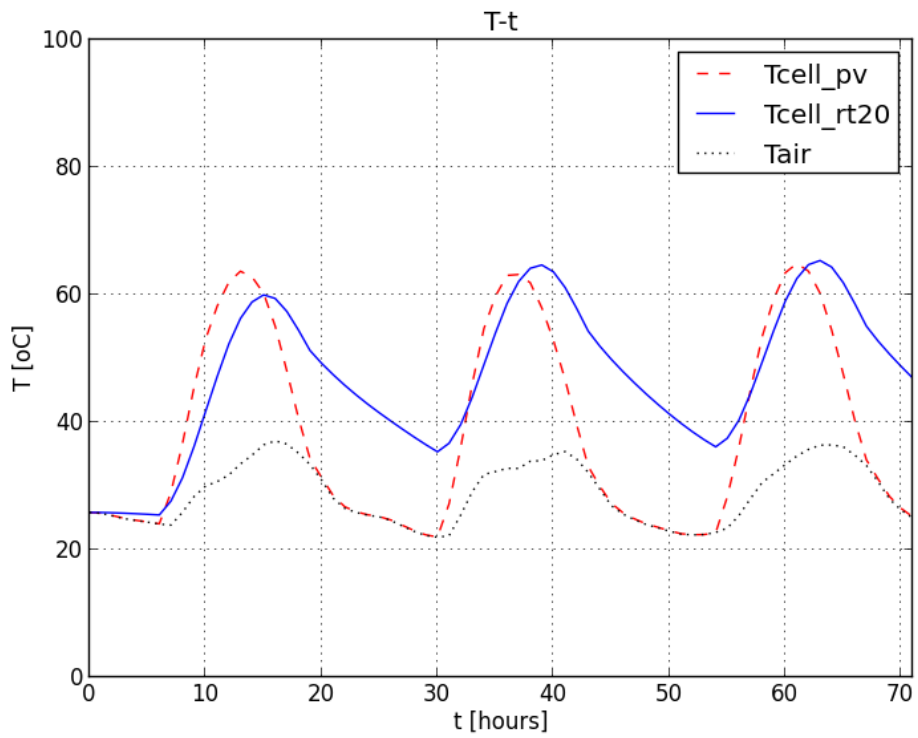


Διάγραμμα Β.6.1.13: Θερμοκρασία κυψέλης, PV-RT20, 1-3/7.

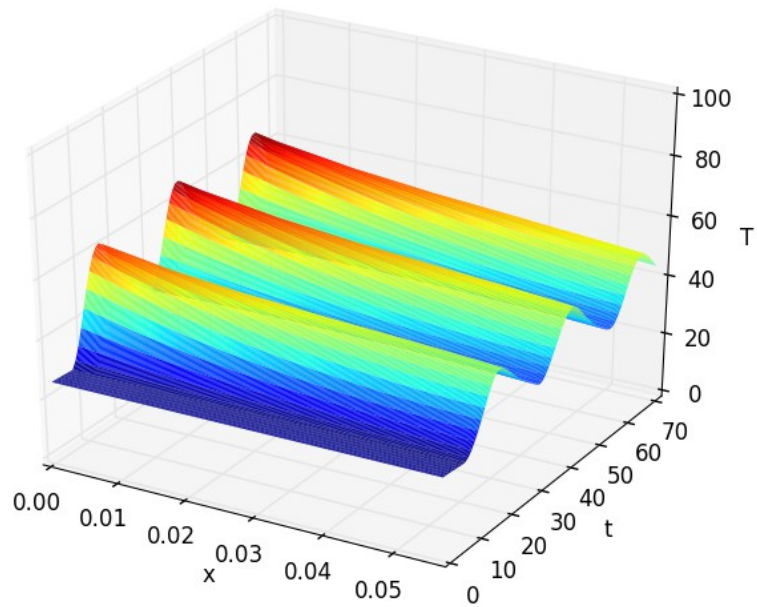


Διάγραμμα Β.6.1.14: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT20, 1-3/7.

Περίοδος:1-3/8



Διάγραμμα Β.6.1.15: Θερμοκρασία κυψέλης, PV-RT20, 1-3/8.

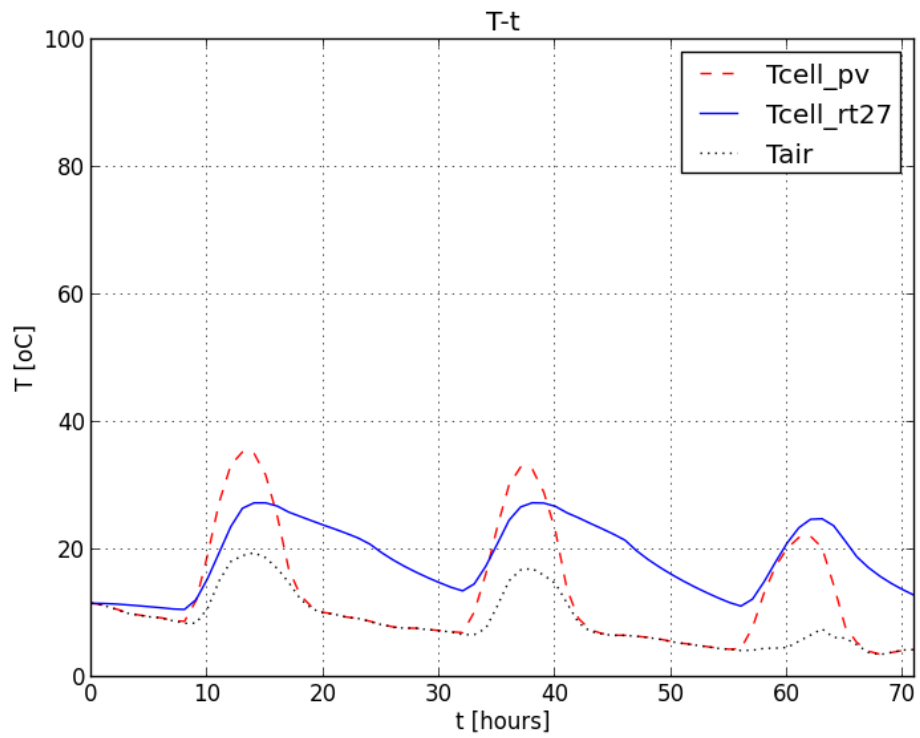


Διάγραμμα Β.6.1.16: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT20, 1-3/8.

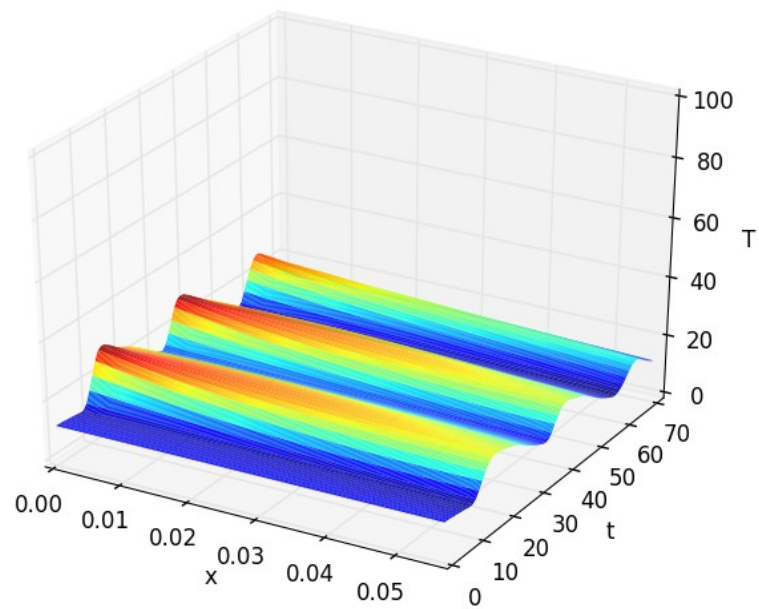
B.6.2-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-RT27

Στην περίπτωση του συστήματος PV-RT27, ο κώδικας `implicit_pv_rt27.py` (Γ.6.2) σε συνδυασμό με τις ρουτίνες επίλυσης τριδιαγώνιων συστημάτων (`thomas_algorithm.py`, Γ.8.1) και υπολογισμού της ενεργού θερμοχωρητικότητας του υλικού RT27 (`effective_heat_capacity_rt27.py`, Γ.8.2) αποδίδει το μητρώο θερμοκρασίας κυψέλης (`tcellarray_pv_rt27.txt`), τα παρακάτω διαγράμματα, καθώς και τα διαγράμματα θερμοκρασίας στο εσωτερικό του PCM, που θα χρησιμοποιηθούν στη συνέχεια (διαγράμματα B.7.2, B.7.6, B.7.10, B.7.14, B.7.18, B.7.22, B.7.26, B.7.30)

Περίοδος:1-3/1

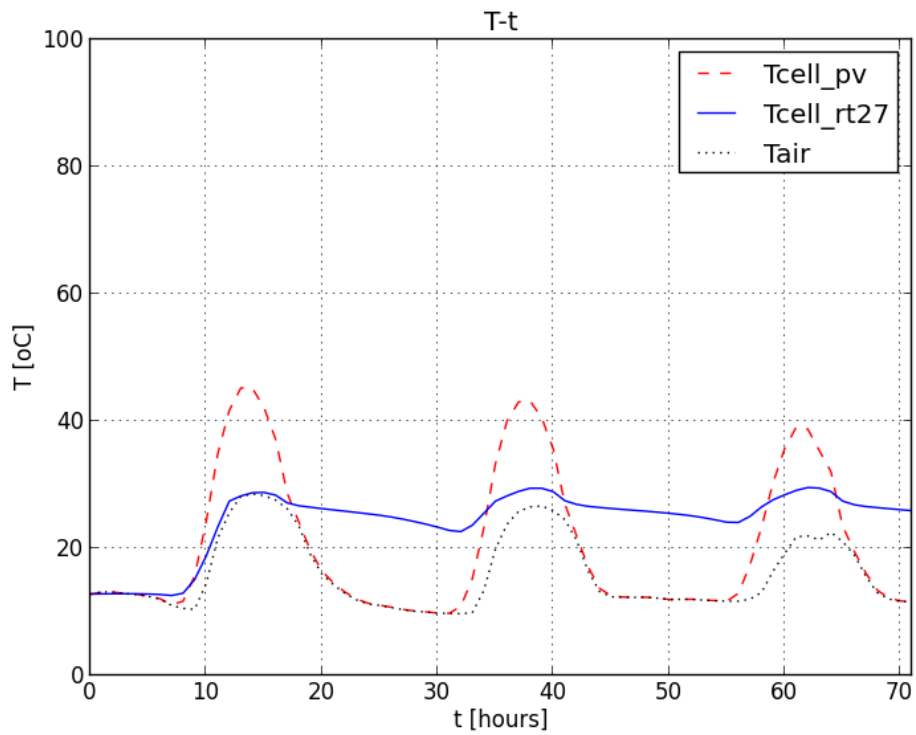


Διάγραμμα Β.6.2.1: Θερμοκρασία κυψέλης, PV-RT27, 1-3/1.

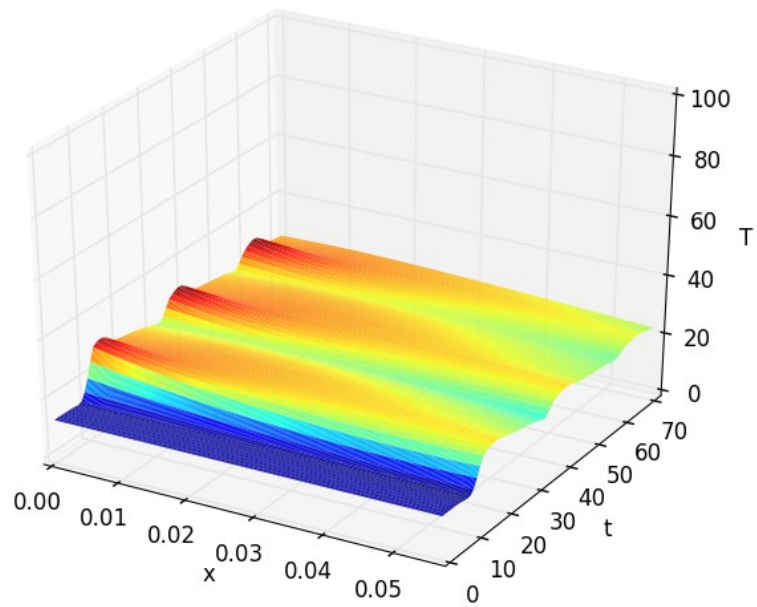


Διάγραμμα Β.6.2.2: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT27, 1-3/1.

Περίοδος:1-3/2

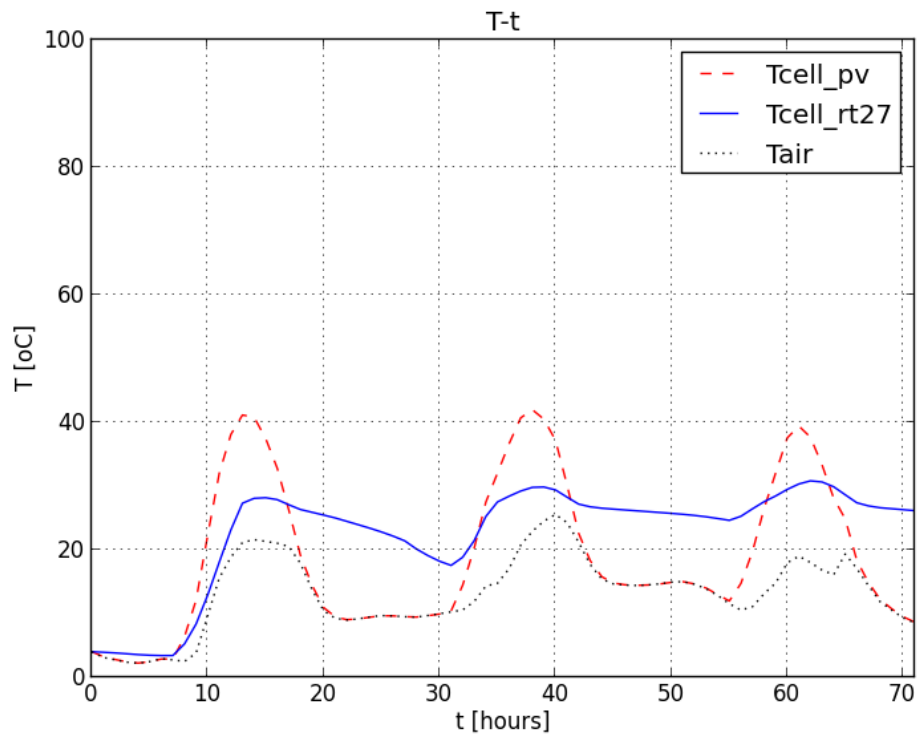


Διάγραμμα Β.6.2.3: Θερμοκρασία κυψέλης, PV-RT27, 1-3/2.

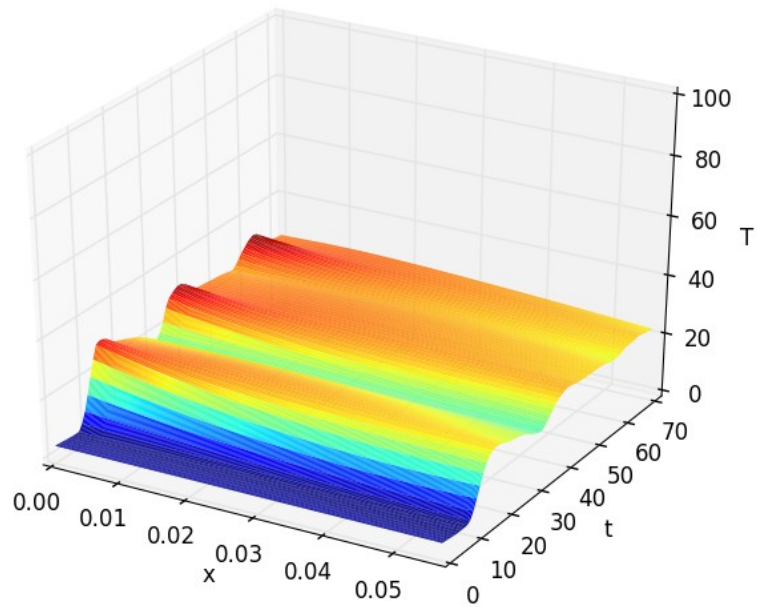


Διάγραμμα Β.6.2.4: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT27, 1-3/2.

Περίοδος:1-3/3

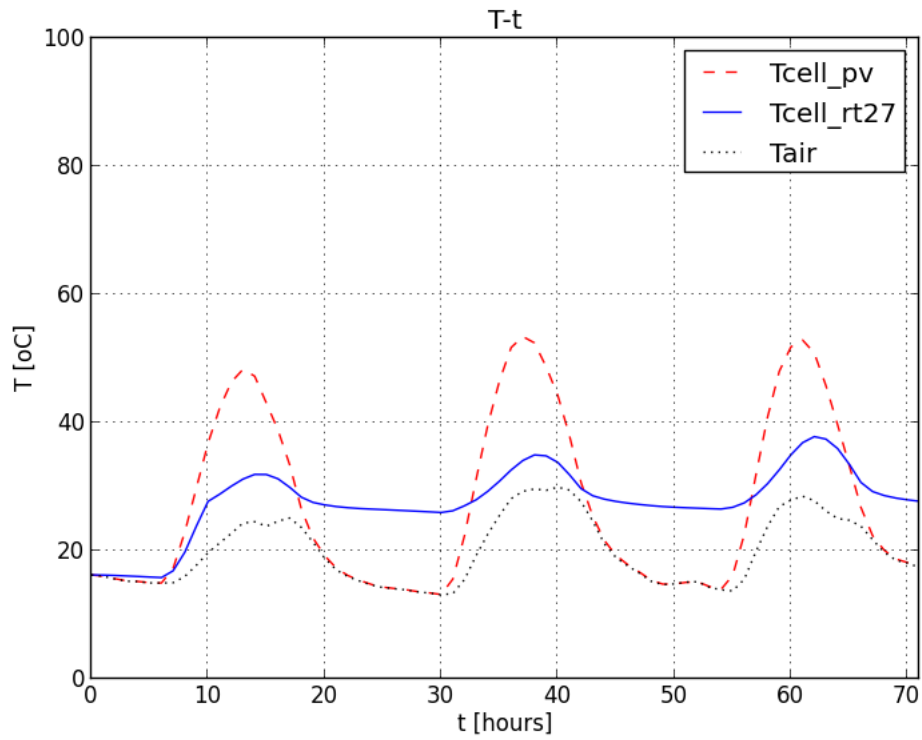


Διάγραμμα Β.6.2.5: Θερμοκρασία κυψέλης, PV-RT27, 1-3/3.

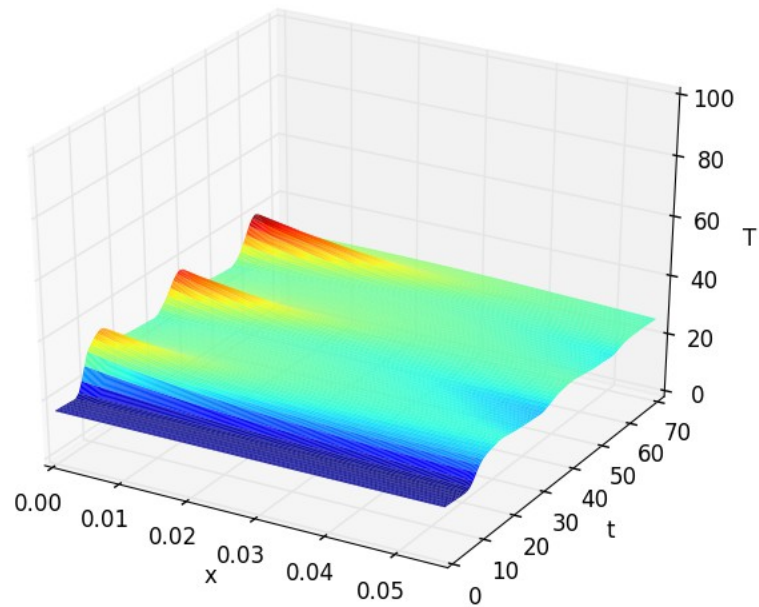


Διάγραμμα Β.6.2.6: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT27, 1-3/3.

Περίοδος:1-3/4

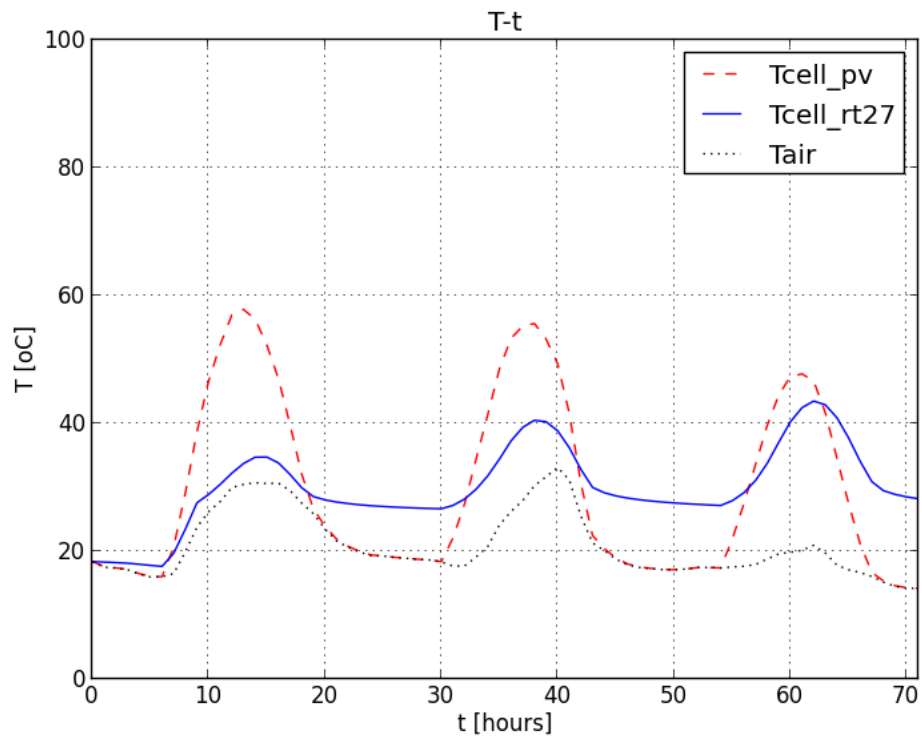


Διάγραμμα Β.6.2.7: Θερμοκρασία κυψέλης, PV-RT27, 1-3/4.

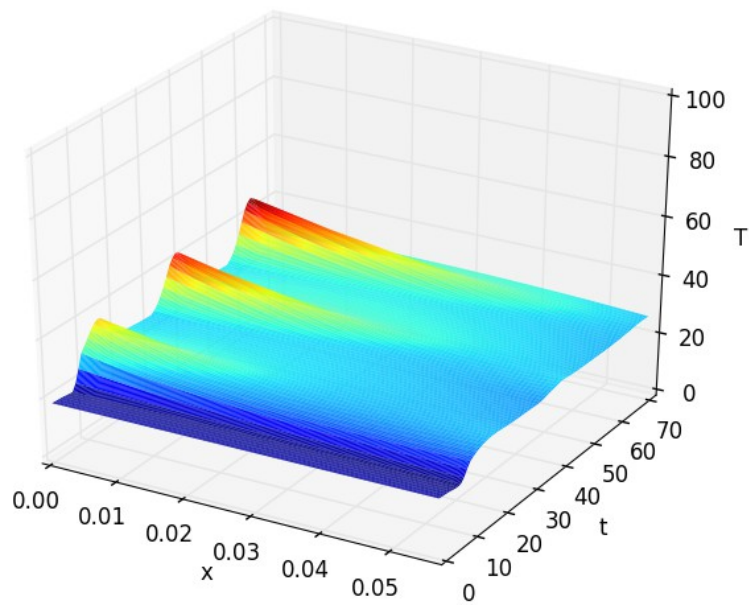


Διάγραμμα Β.6.2.8: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT27, 1-3/4.

Περίοδος:1-3/5

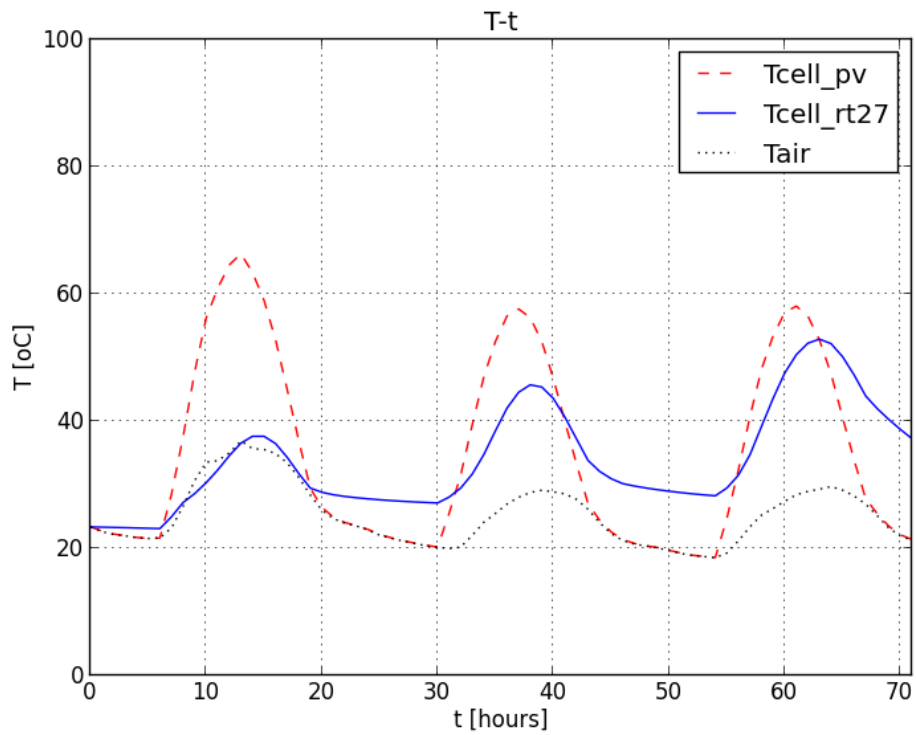


Διάγραμμα Β.6.2.9: Θερμοκρασία κυψέλης, PV-RT27, 1-3/5.

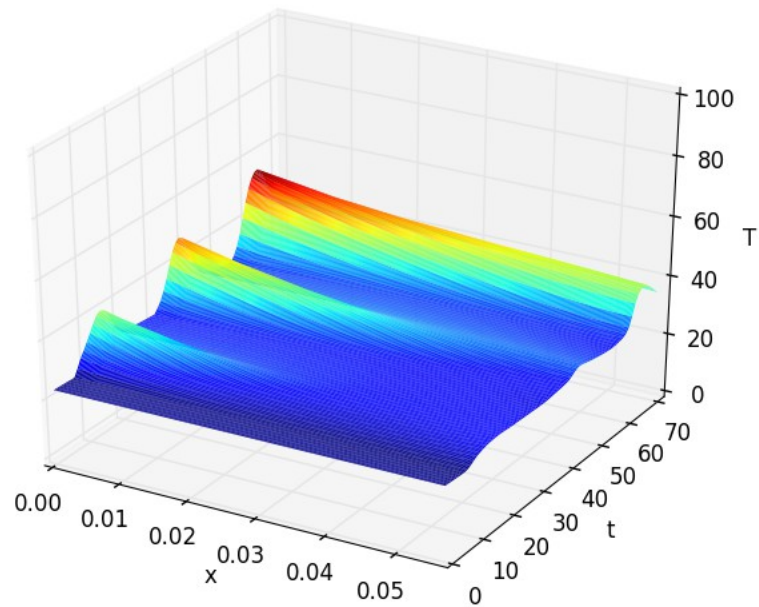


Διάγραμμα Β.6.2.10: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT27, 1-3/5.

Περίοδος:1-3/6

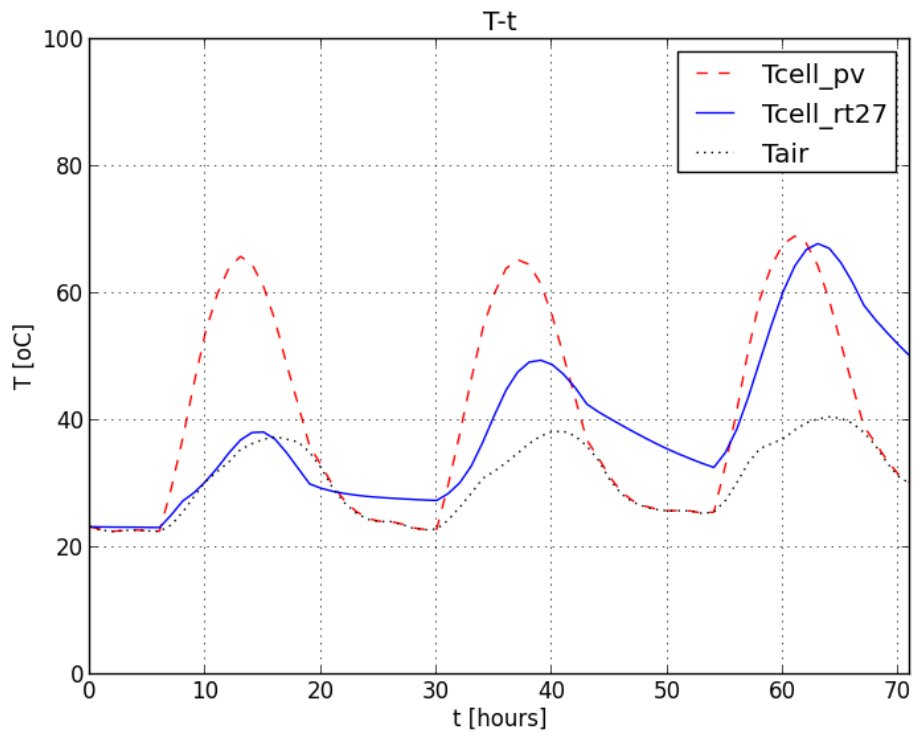


Διάγραμμα Β.6.2.11: Θερμοκρασία κυψέλης, PV-RT27, 1-3/6.

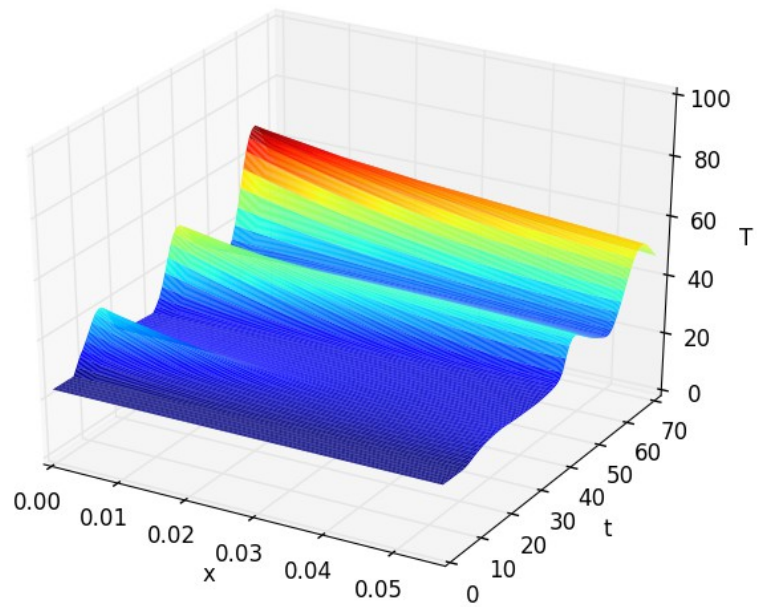


Διάγραμμα Β.6.2.12: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT27, 1-3/6.

Περίοδος:1-3/7

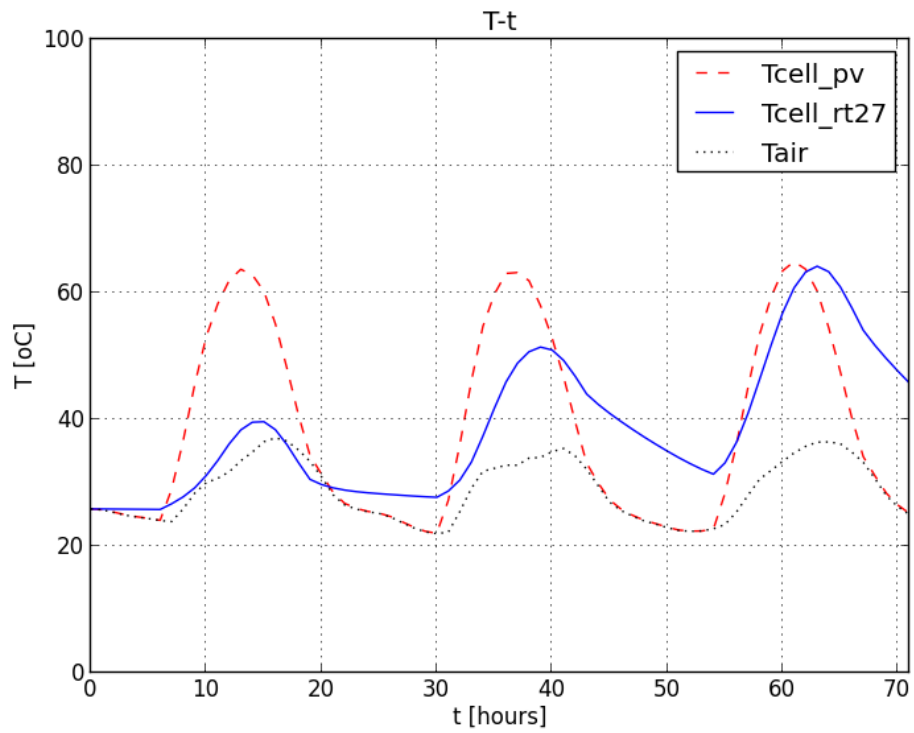


Διάγραμμα Β.6.2.13: Θερμοκρασία κυψέλης, PV-RT27, 1-3/7.

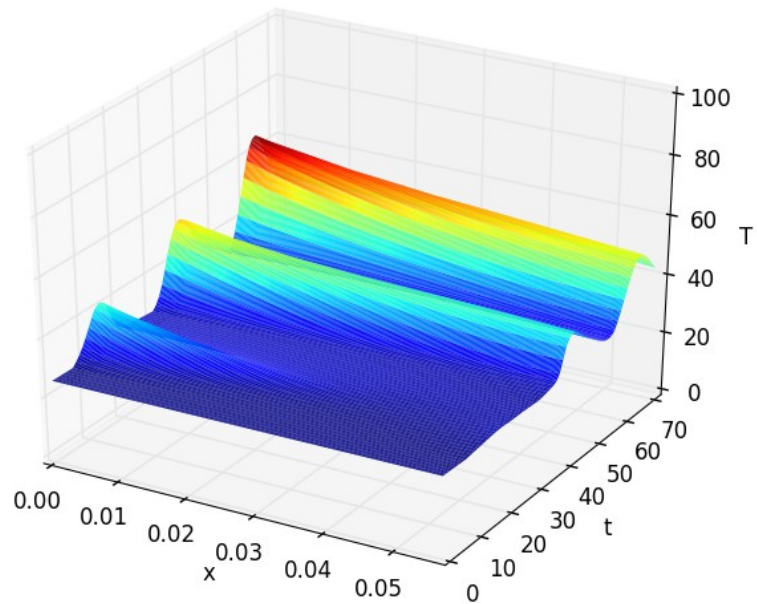


Διάγραμμα Β.6.2.14: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT27, 1-3/7.

Περίοδος:1-3/8



Διάγραμμα Β.6.2.15: Θερμοκρασία κυψέλης, PV-RT27, 1-3/8.

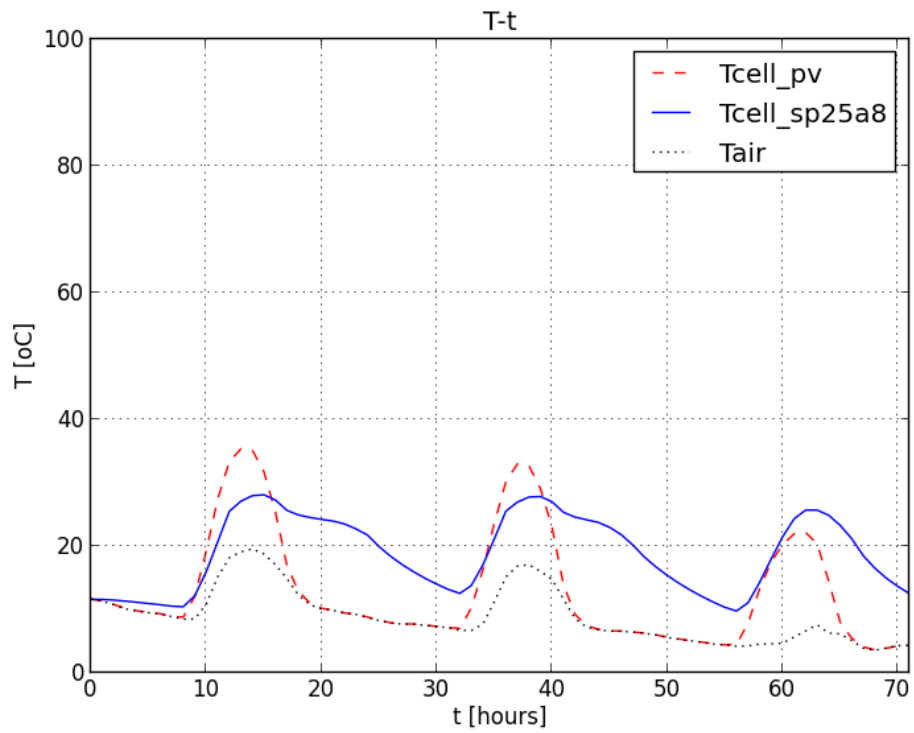


Διάγραμμα Β.6.2.16: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-RT27, 1-3/8.

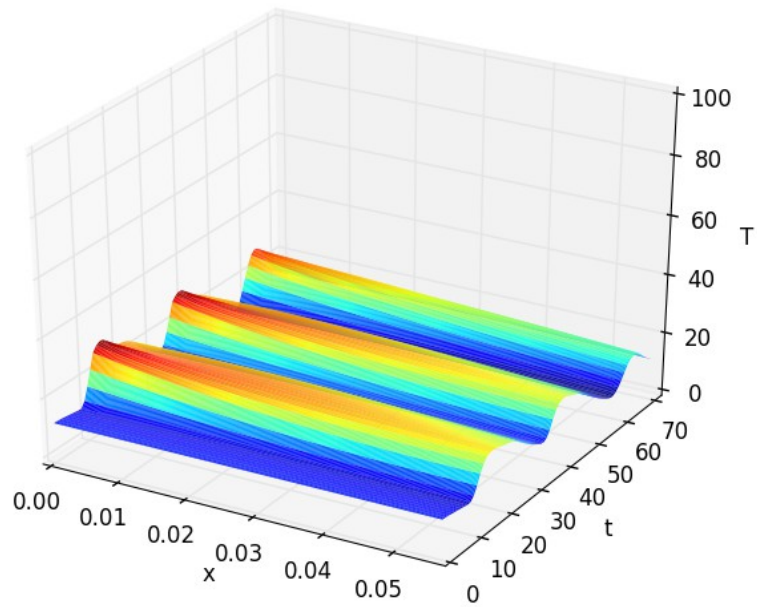
B.6.3-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-SP25A8

Για το σύστημα PV-SP25A8 ο αντίστοιχος κώδικας `implicit_pv_sp25a8.py` (Γ.6.3) με τις ρουτίνες επίλυσης τριδιαγώνιων συστημάτων (`thomas_algorithm.py`, Γ.8.1) και υπολογισμού της ενεργού θερμοχωρητικότητας του υλικού SP25A8 (`effective_heat_capacity_sp25a8.py`, Γ.8.4), δίνει το μητρώο θερμοκρασίας κυψέλης (`tcellarray_pv_sp25a8.txt`), τα παρακάτω διαγράμματα, καθώς και τα διαγράμματα θερμοκρασίας στο εσωτερικό του PCM, που θα χρησιμοποιηθούν στη συνέχεια (διαγράμματα B.7.3, B.7.7, B.7.11, B.7.15, B.7.19, B.7.23, B.7.27, B.7.31)

Περίοδος:1-3/1

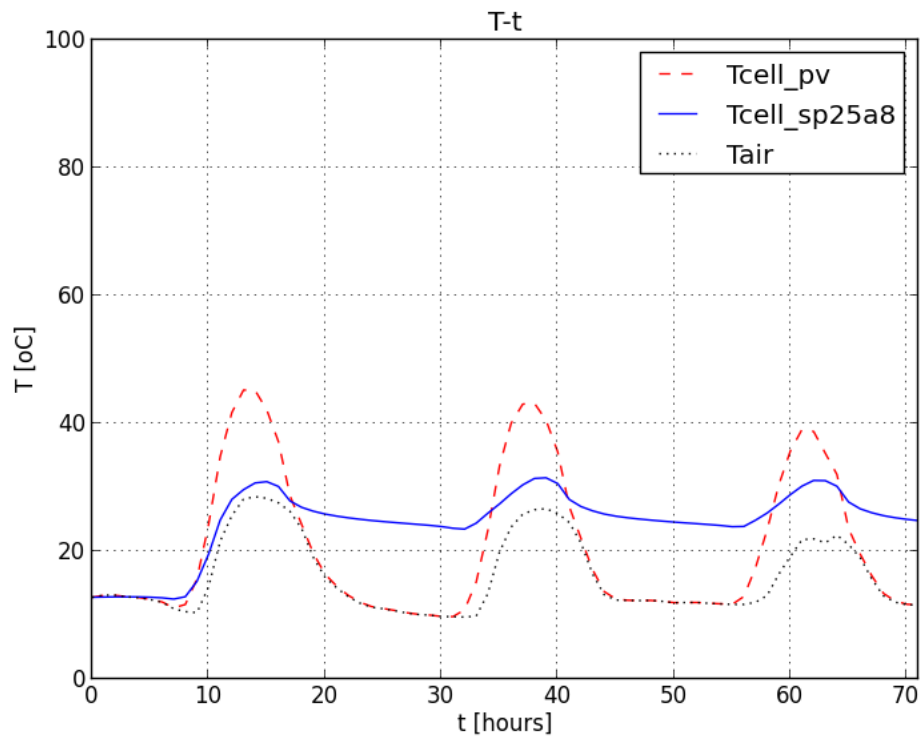


Διάγραμμα Β.6.3.1: Θερμοκρασία κυψέλης, PV-SP25A8, 1-3/1.

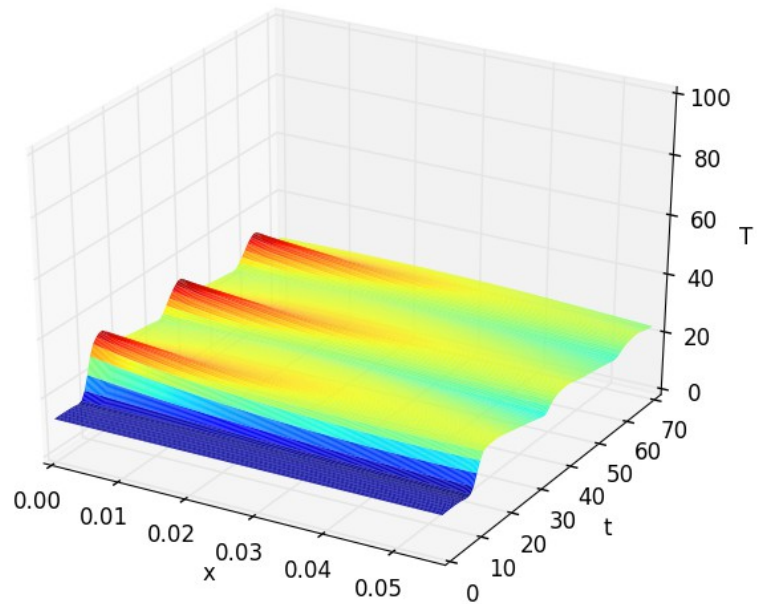


Διάγραμμα Β.6.3.2: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, SP25A8, 1-3/1.

Περίοδος:1-3/2

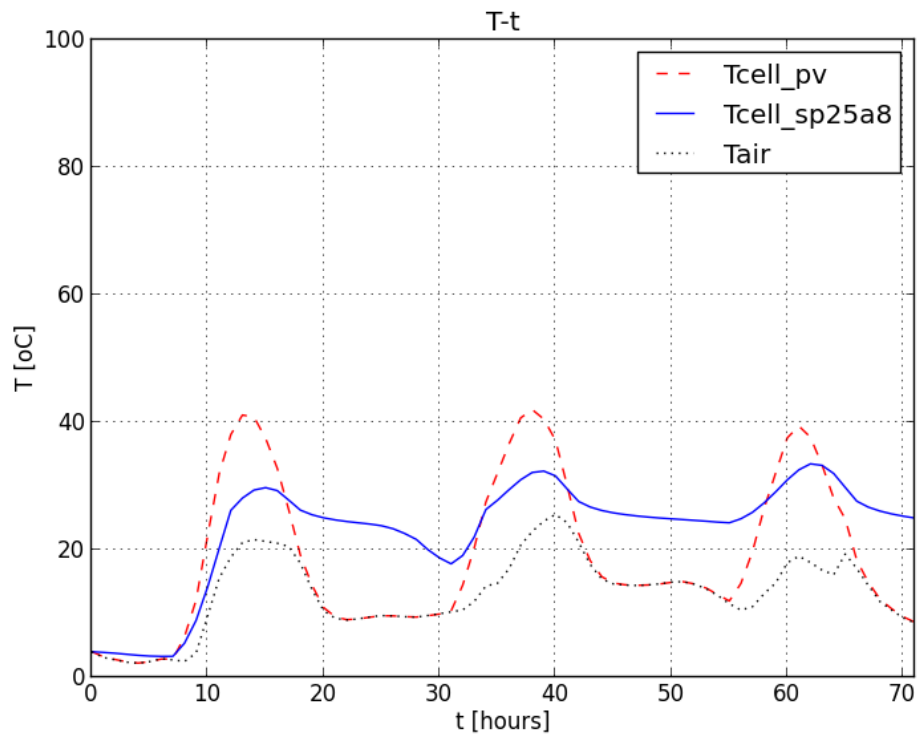


Διάγραμμα Β.6.3.3: Θερμοκρασία κυψέλης, PV-SP25A8, 1-3/2.

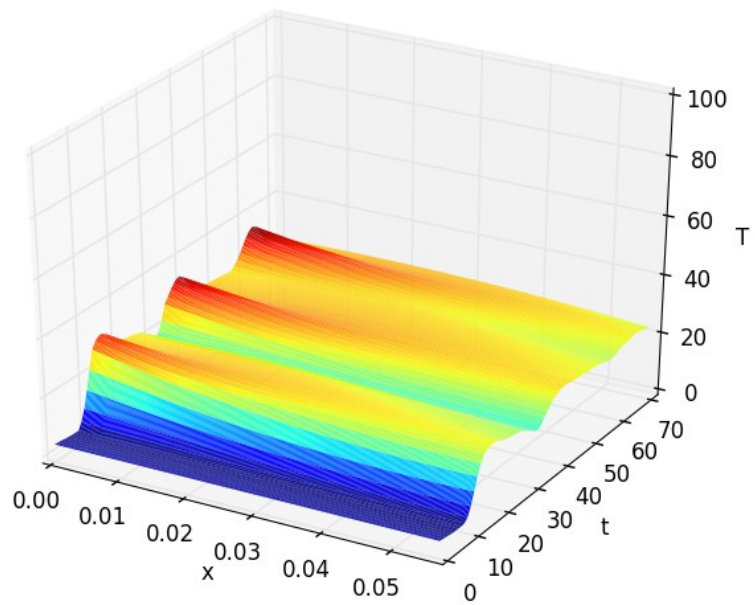


Διάγραμμα Β.6.3.4: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-SP25A8, 1-3/2.

Περίοδος:1-3/3

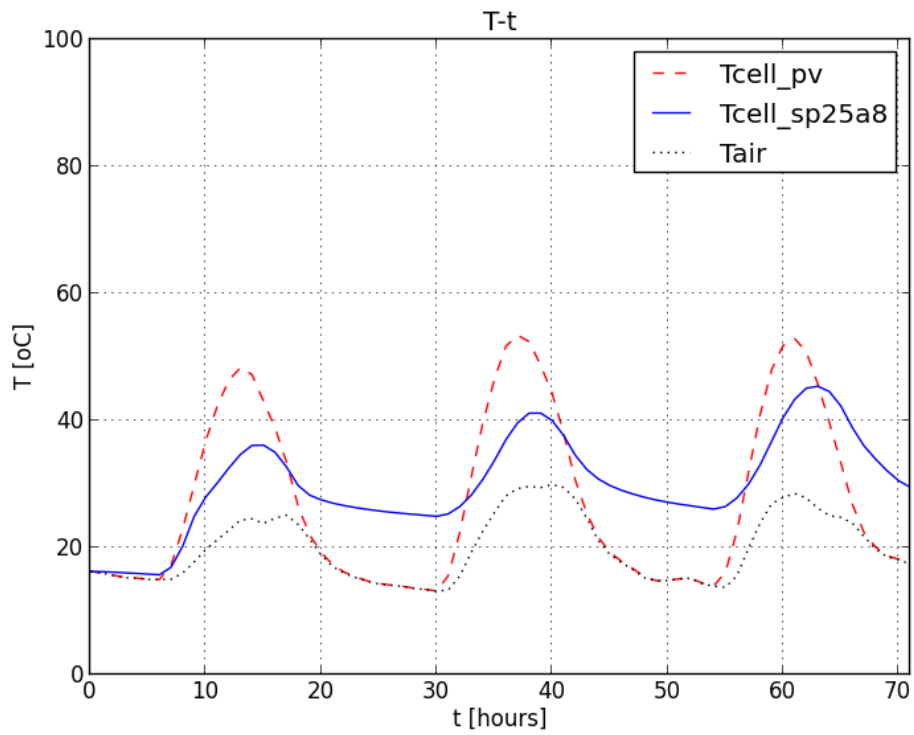


Διάγραμμα Β.6.3.5: Θερμοκρασία κυψέλης, PV-SP25A8, 1-3/3.

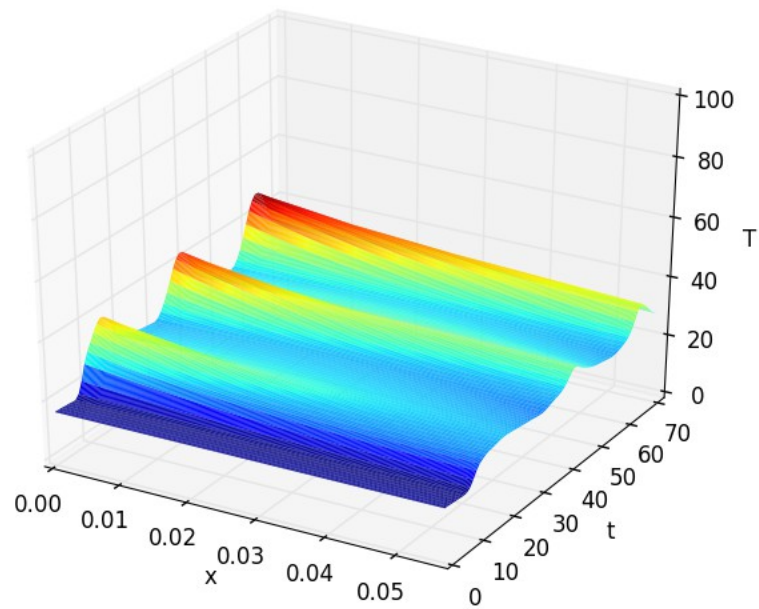


Διάγραμμα Β.6.3.6: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-SP25A8, 1-3/3.

Περίοδος:1-3/4

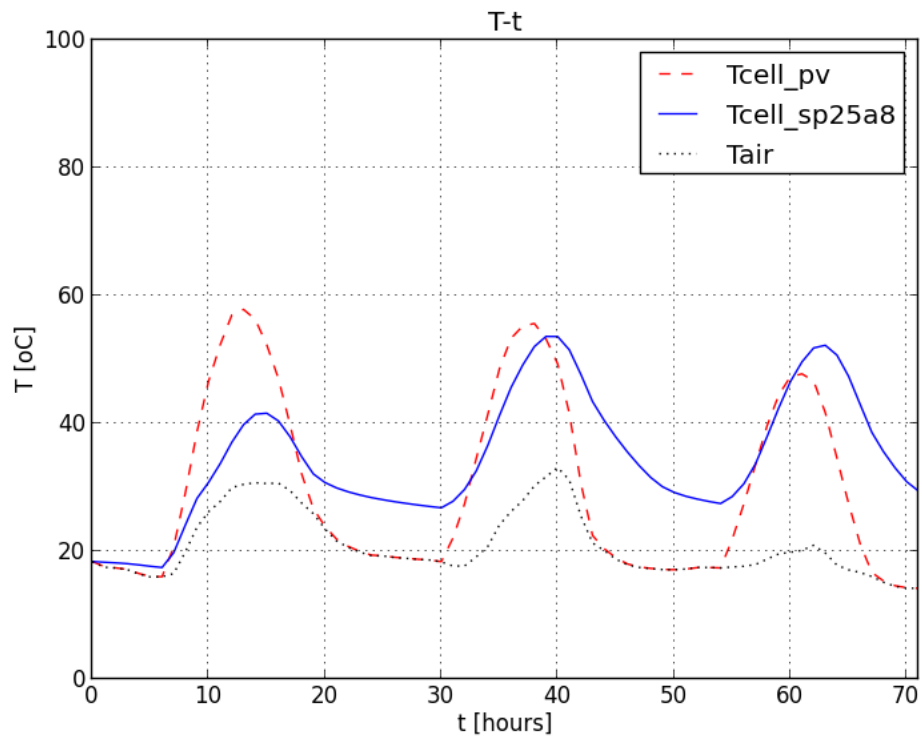


Διάγραμμα Β.6.3.7: Θερμοκρασία κυψέλης, PV-SP25A8, 1-3/4.

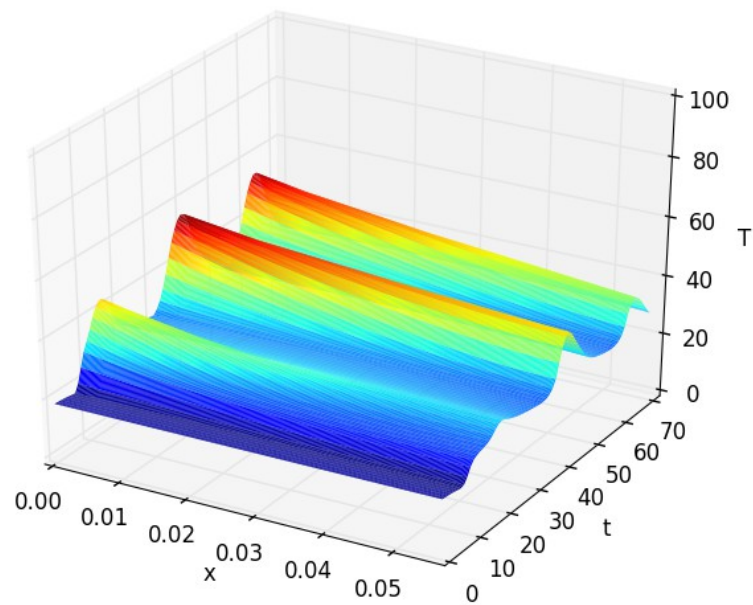


Διάγραμμα Β.6.3.8: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-SP25A8, 1-3/4.

Περίοδος:1-3/5

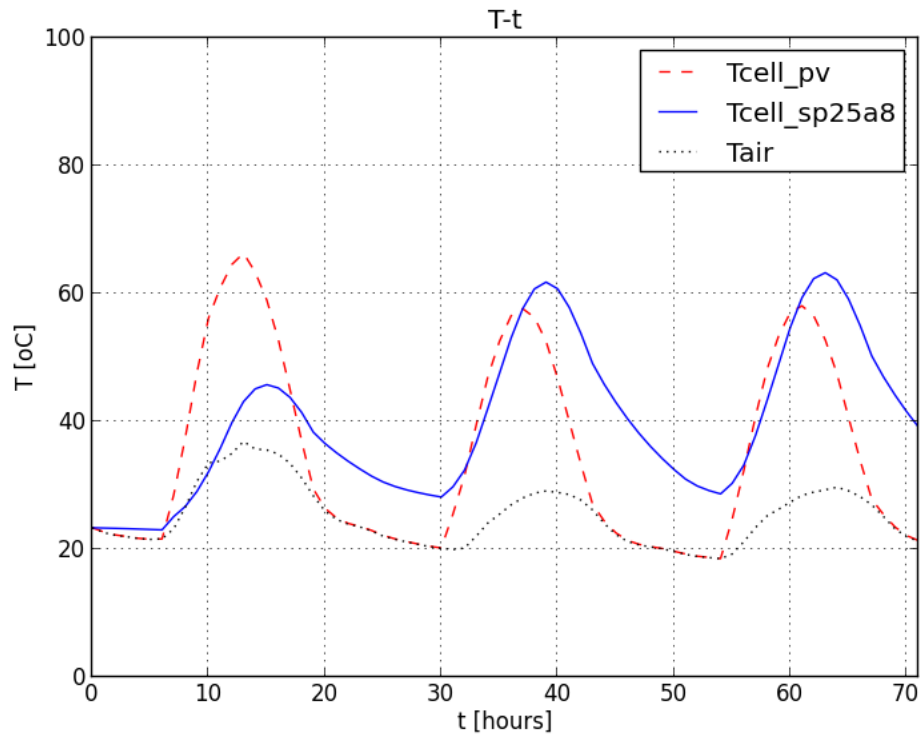


Διάγραμμα Β.6.3.9: Θερμοκρασία κυψέλης, PV-SP25A8, 1-3/5.

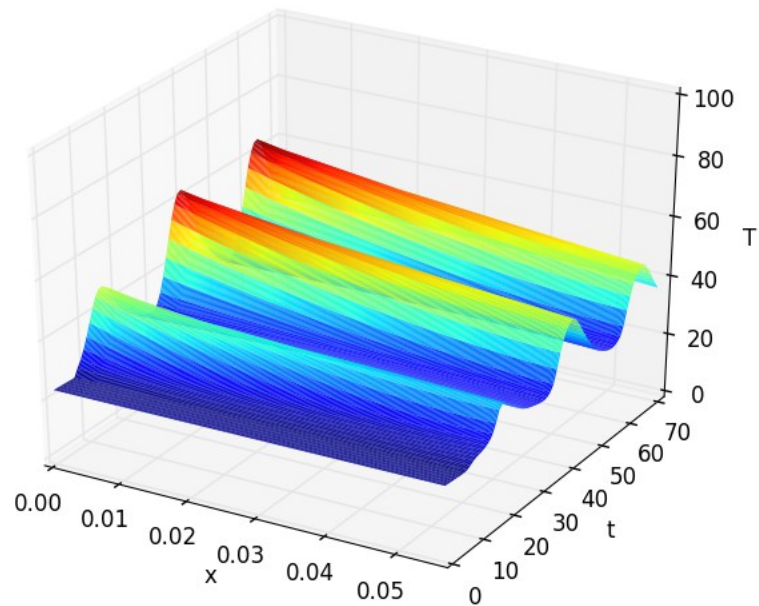


Διάγραμμα Β.6.3.10: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-SP25A8, 1-3/5.

Περίοδος:1-3/6

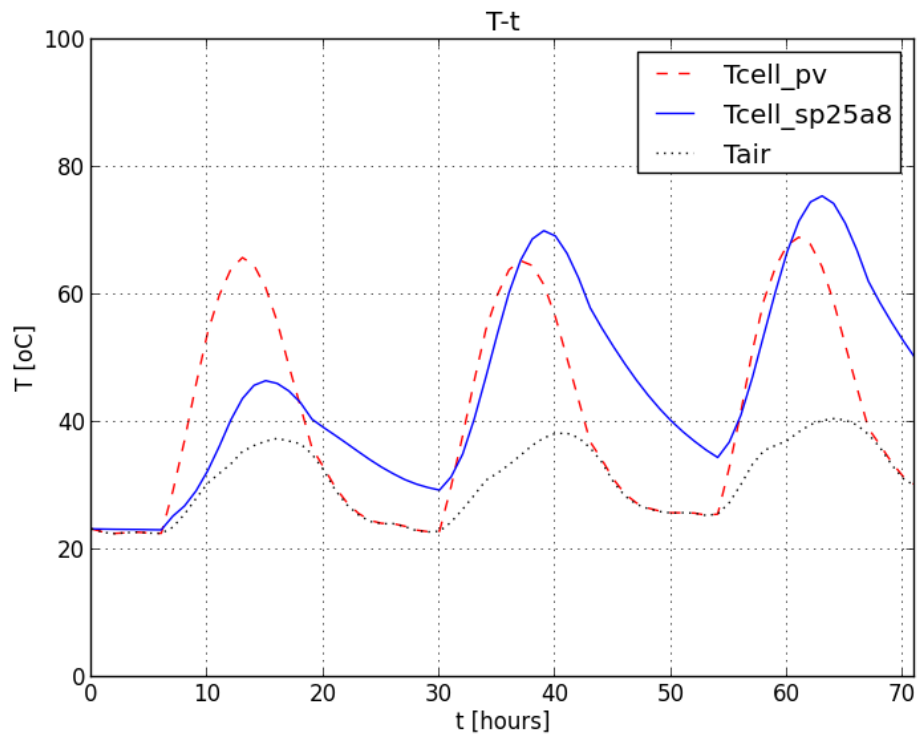


Διάγραμμα Β.6.3.11: Θερμοκρασία κυψέλης, PV-SP25A8, 1-3/6.

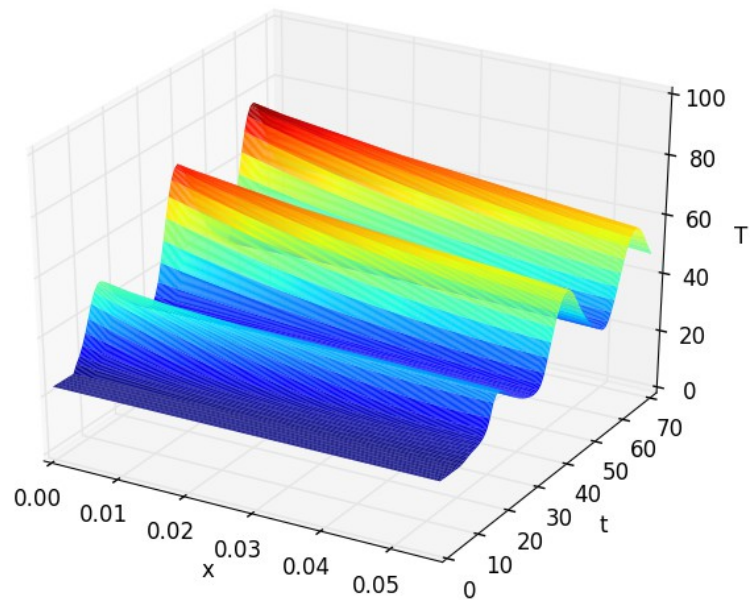


Διάγραμμα Β.6.3.12: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-SP25A8, 1-3/6.

Περίοδος:1-3/7

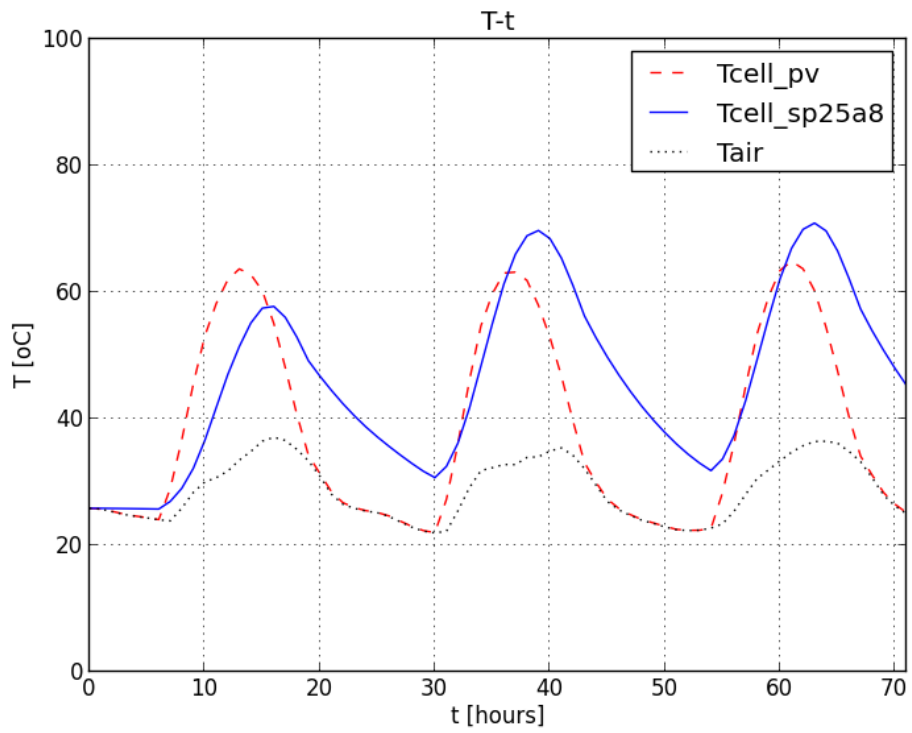


Διάγραμμα Β.6.3.13: Θερμοκρασία κυψέλης, PV-SP25A8, 1-3/7.

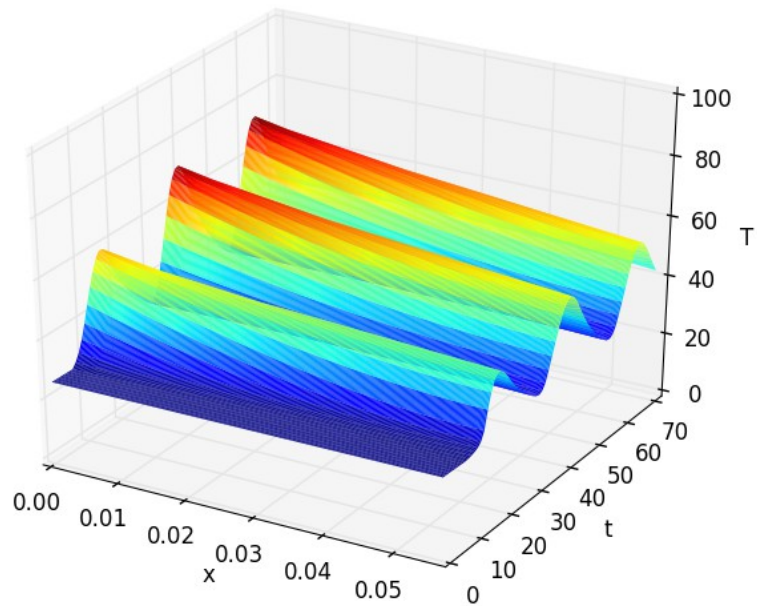


Διάγραμμα Β.6.3.14: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-SP25A8, 1-3/7.

Περίοδος:1-3/8



Διάγραμμα Β.6.3.15: Θερμοκρασία κυψέλης, PV-SP25A8, 1-3/8.

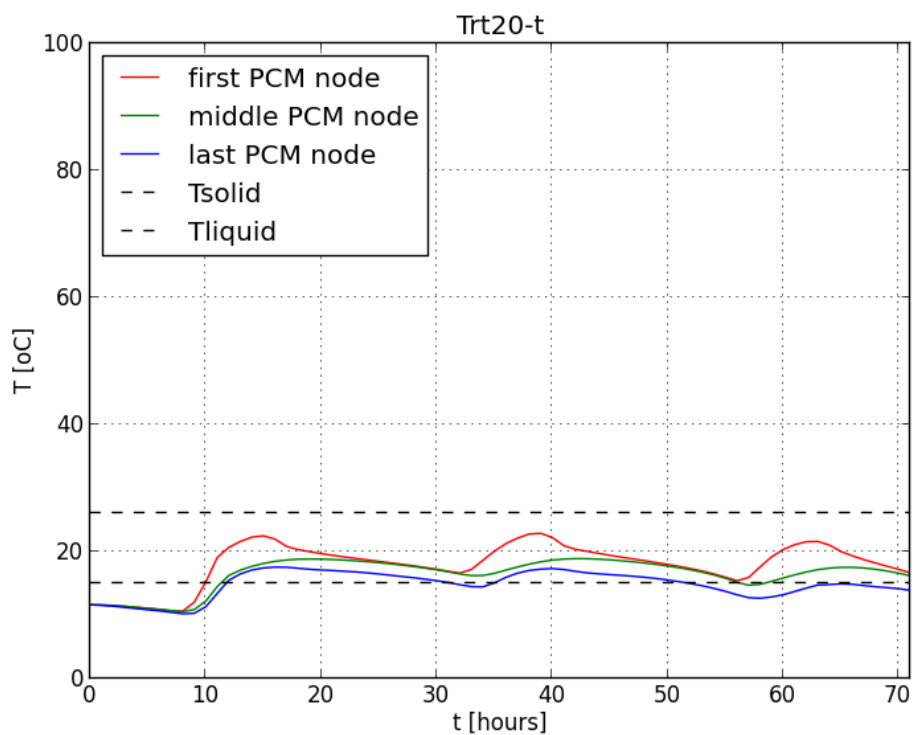


Διάγραμμα Β.6.3.16: Θερμοκρασιακή κατανομή συναρτήσει του χρόνου, PV-SP25A8, 1-3/8.

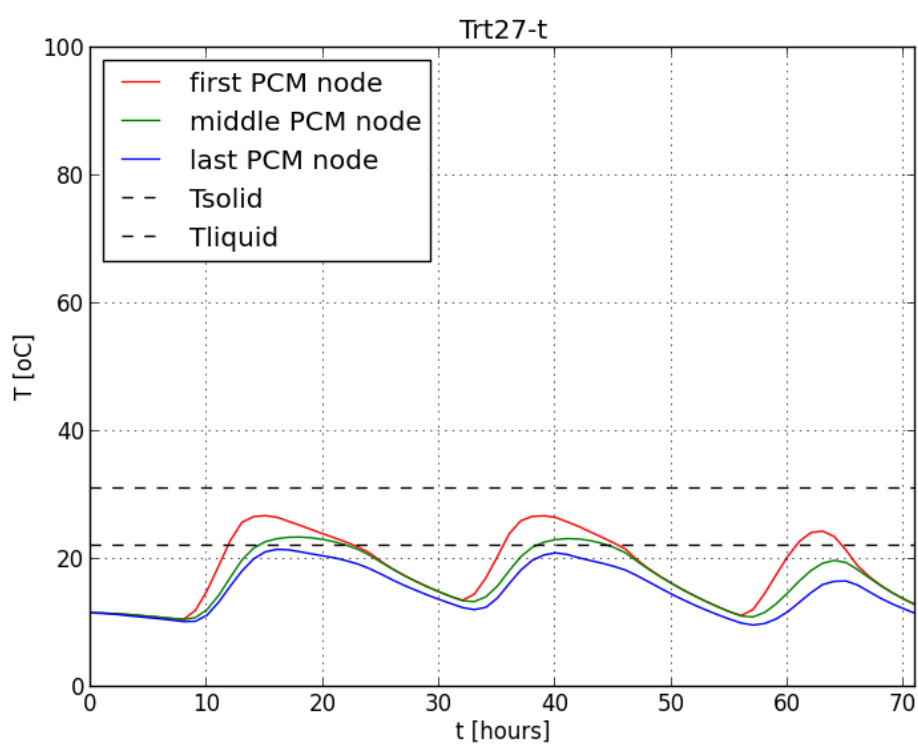
B.7-ΣΥΓΚΡΙΤΙΚΑ ΔΙΑΓΡΑΜΜΑΤΑ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ

Εκτελώντας τον κώδικα `comparison.py` (Γ.7) για τα μητρώα θερμοκρασιών κυψέλης `tcellarray_pn.txt`, `tcellarray_pn_rt20.txt`, `tcellarray_pn_rt27.txt` και `tcellarray_pn_sp25a8.txt` που αντιστοιχούν σε κάθε χρονική περίοδο παίρνουμε τα παρακάτω συγκριτικά διαγράμματα. Επίσης παρατίθενται και τα διαγράμματα θερμοκρασιών στο εσωτερικό των υλικών αλλαγής φάσης, που προέκυψαν από τους προηγούμενους κώδικες (`implicit_pn_rt20.py` (Γ.6.1), `implicit_pn_rt27.py` (Γ.6.2), `implicit_pn_sp25a8.py` (Γ.6.3)).

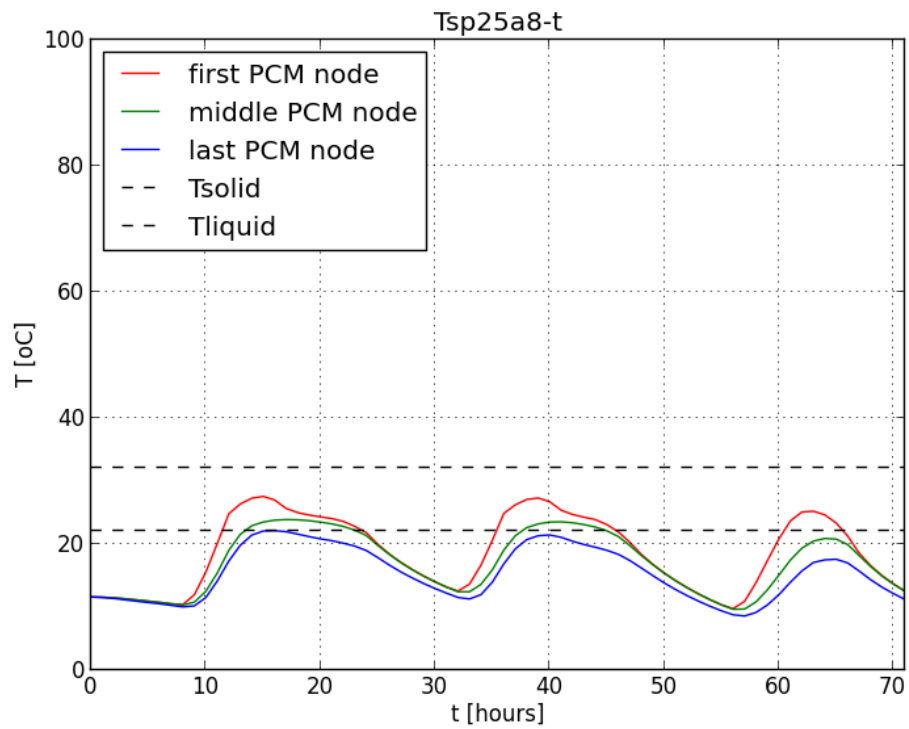
Περίοδος:1-3/1



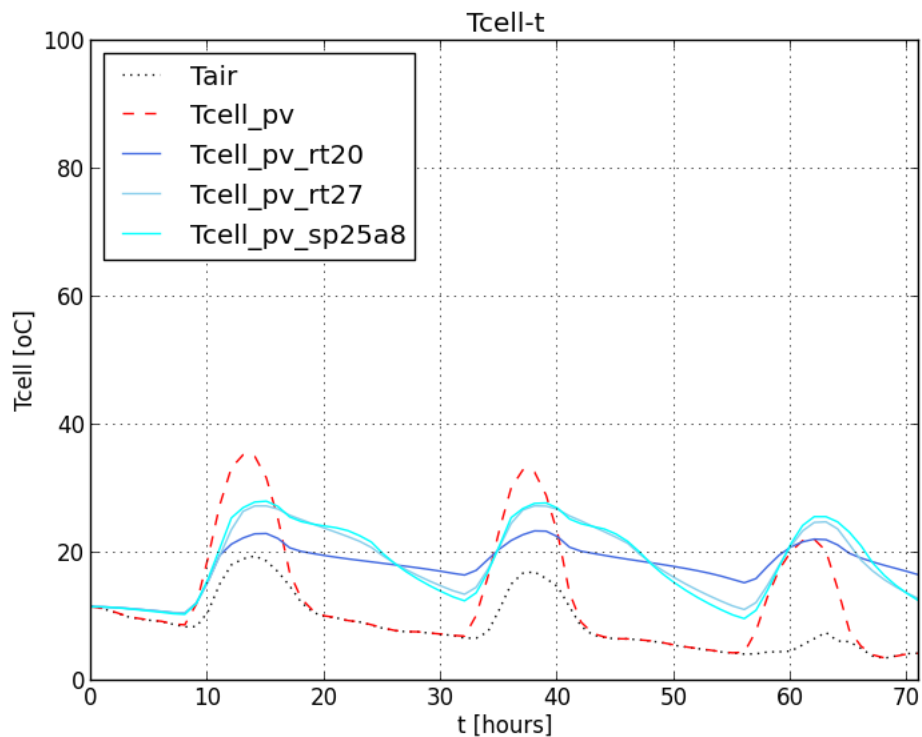
Διάγραμμα Β.7.1: Θερμοκρασίες στο εσωτερικό του RT20, 1-3/1.



Διάγραμμα Β.7.2: Θερμοκρασίες στο εσωτερικό του RT27, 1-3/1.

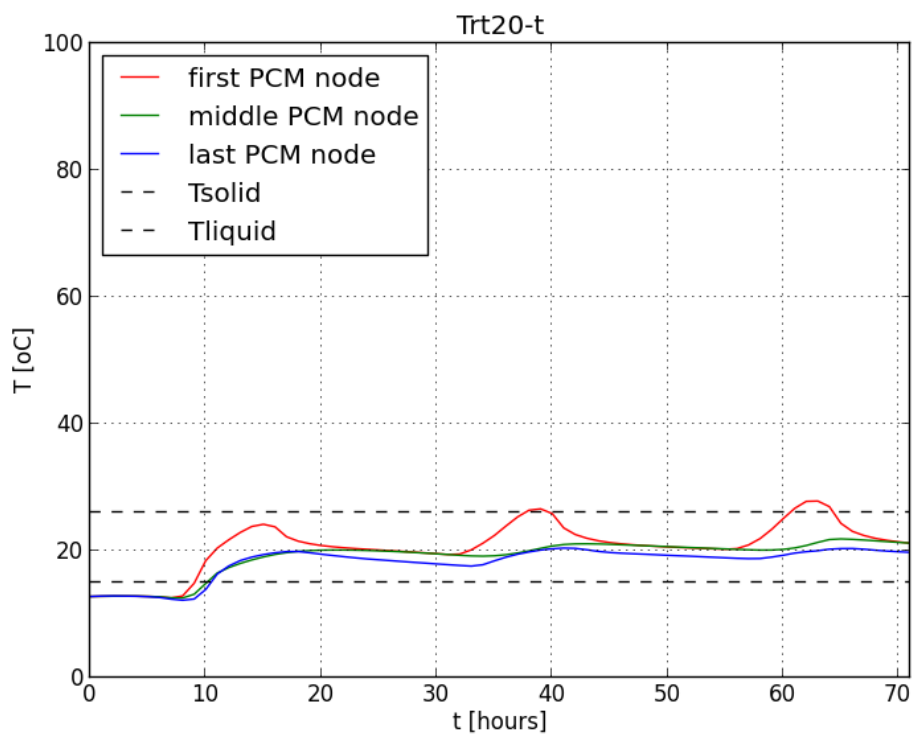


Διάγραμμα Β.7.3: Θερμοκρασίες στο εσωτερικό του SP25A8, 1-3/1.

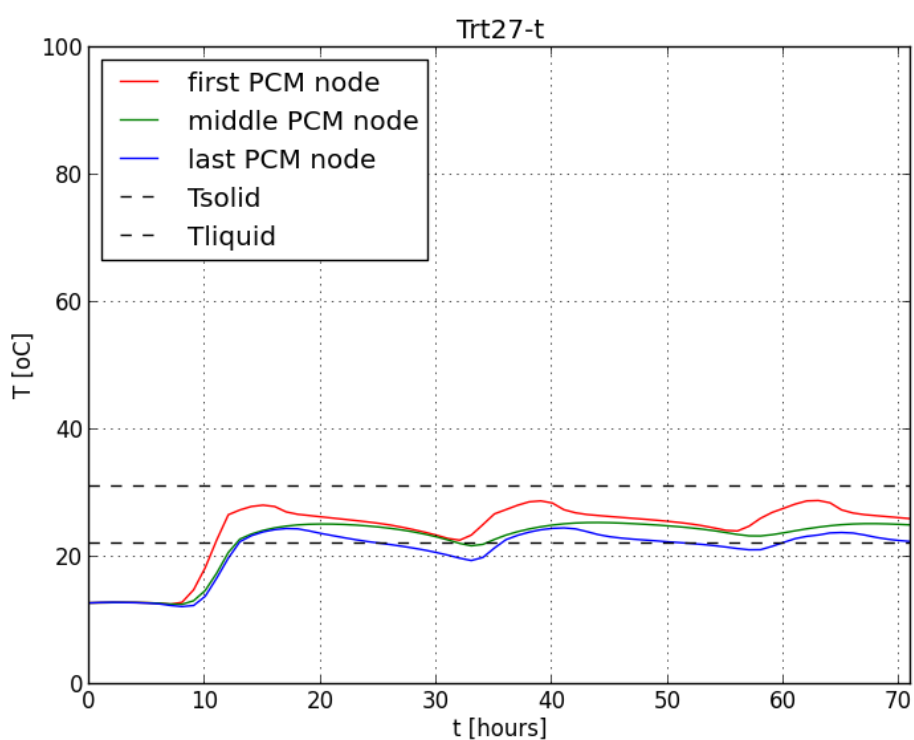


Διάγραμμα Β.7.4: Συγκριτικό διάγραμμα θερμοκρασιών κυψέλης για τις περιπτώσεις PV, PV-RT20, PV-RT27, PV-SP25A8, 1-3/1.

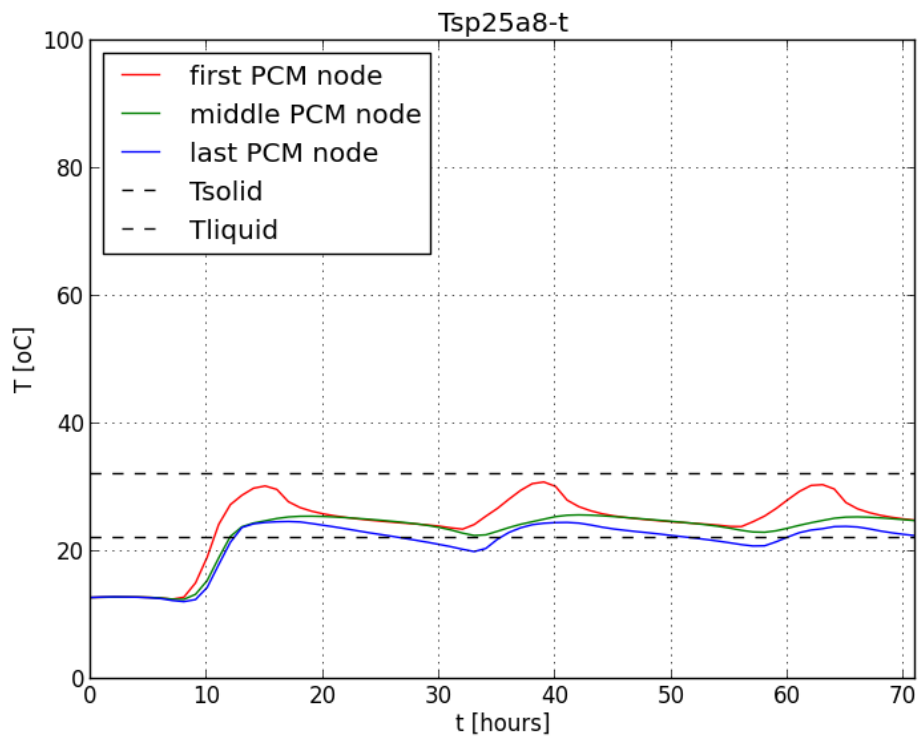
Περίοδος:1-3/2



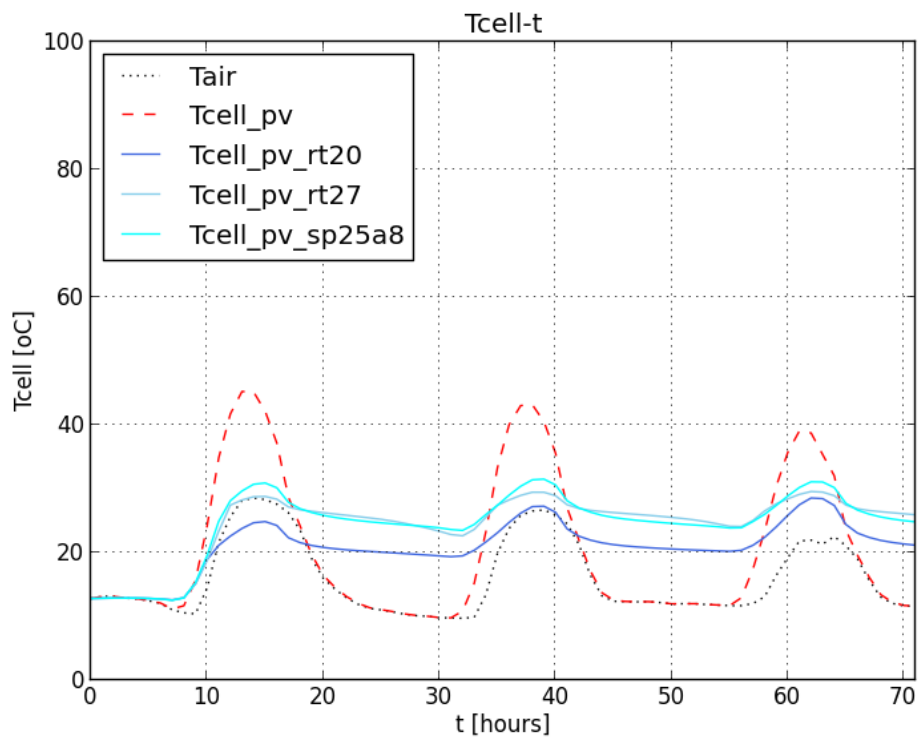
Διάγραμμα Β.7.5: Θερμοκρασίες στο εσωτερικό του RT20, 1-3/2.



Διάγραμμα Β.7.6: Θερμοκρασίες στο εσωτερικό του RT27, 1-3/2.

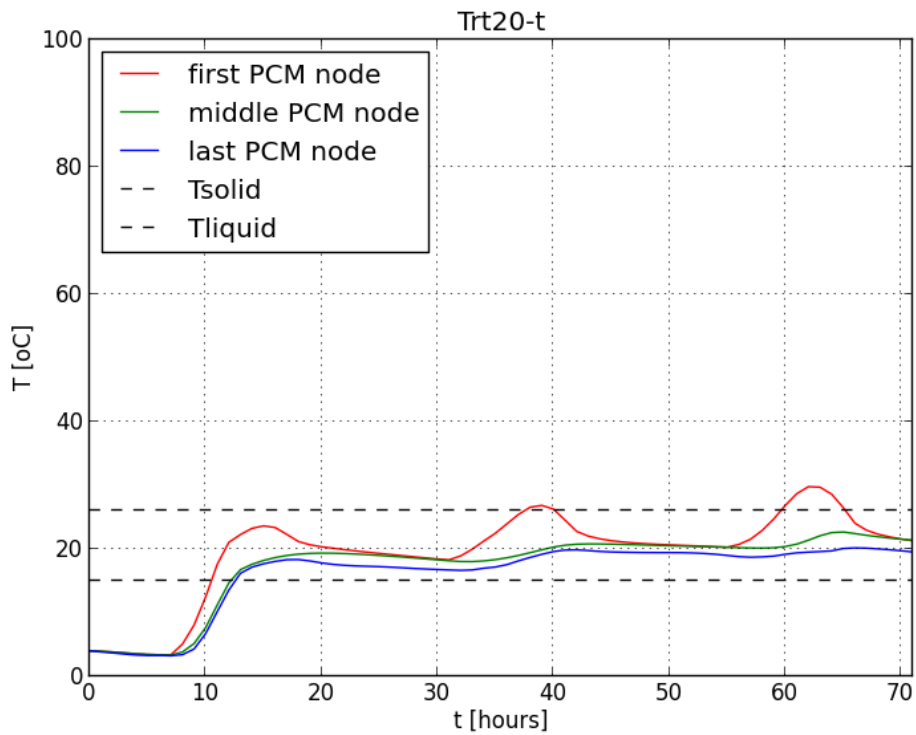


Διάγραμμα Β.7.7: Θερμοκρασίες στο εσωτερικό του SP25A8, 1-3/2.

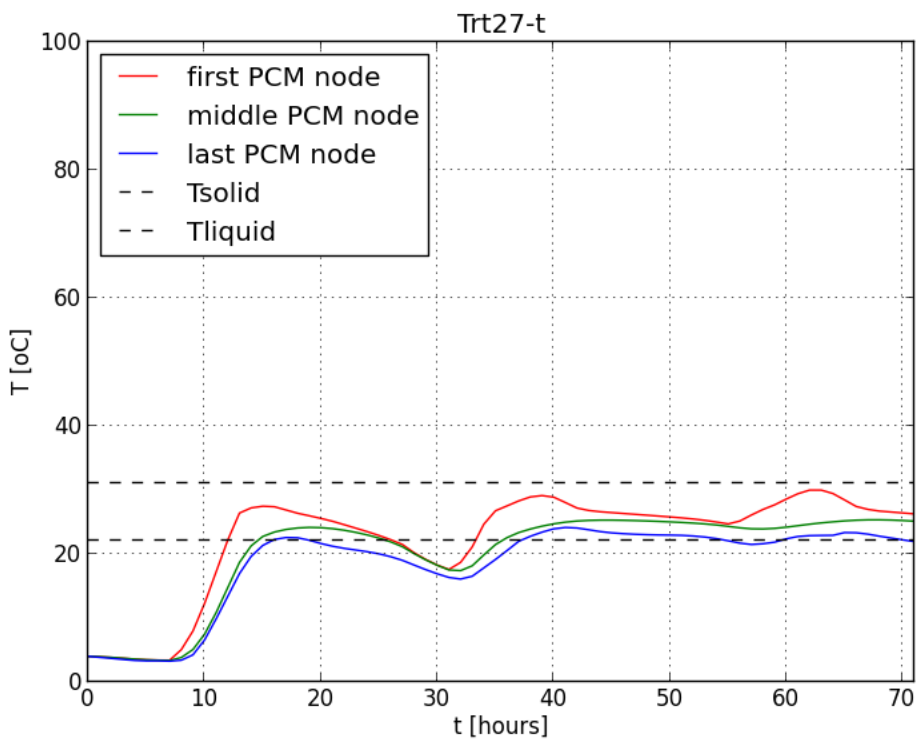


Διάγραμμα Β.7.8: Συγκριτικό διάγραμμα θερμοκρασιών κυψέλης για τις περιπτώσεις PV, PV-RT20, PV-RT27, PV-SP25A8, 1-3/2.

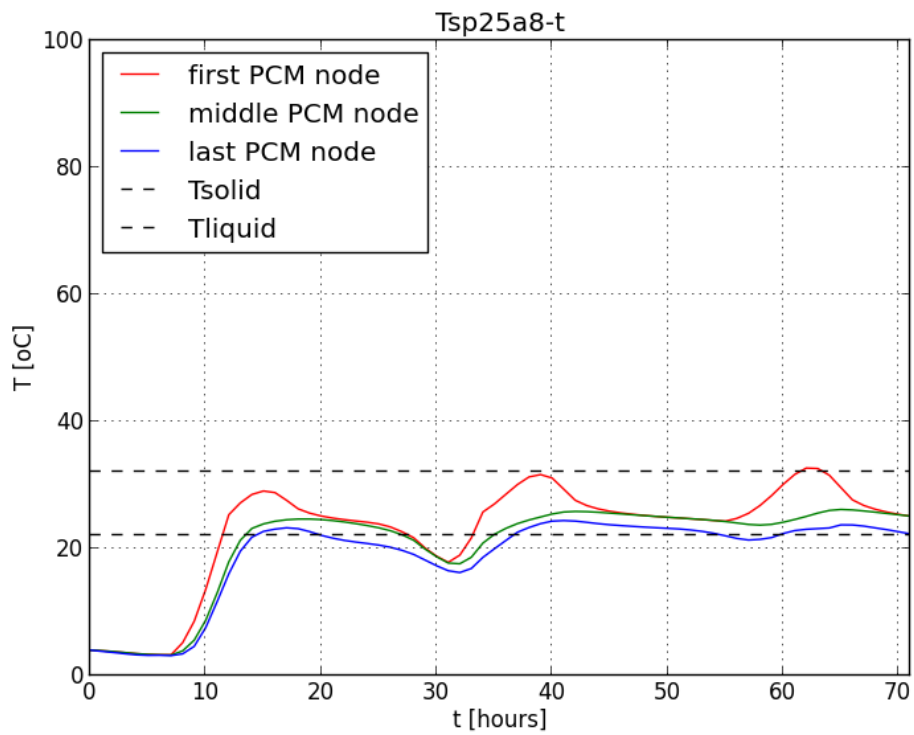
Περίοδος:1-3/3



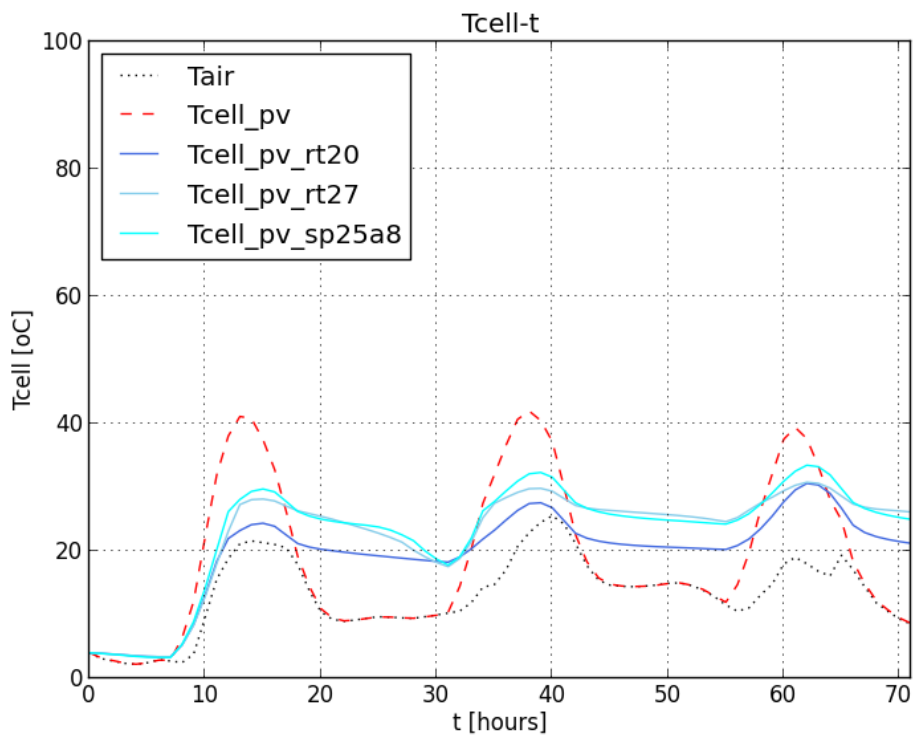
Διάγραμμα Β.7.9: Θερμοκρασίες στο εσωτερικό του RT20, 1-3/3.



Διάγραμμα Β.7.10: Θερμοκρασίες στο εσωτερικό του RT27, 1-3/3.

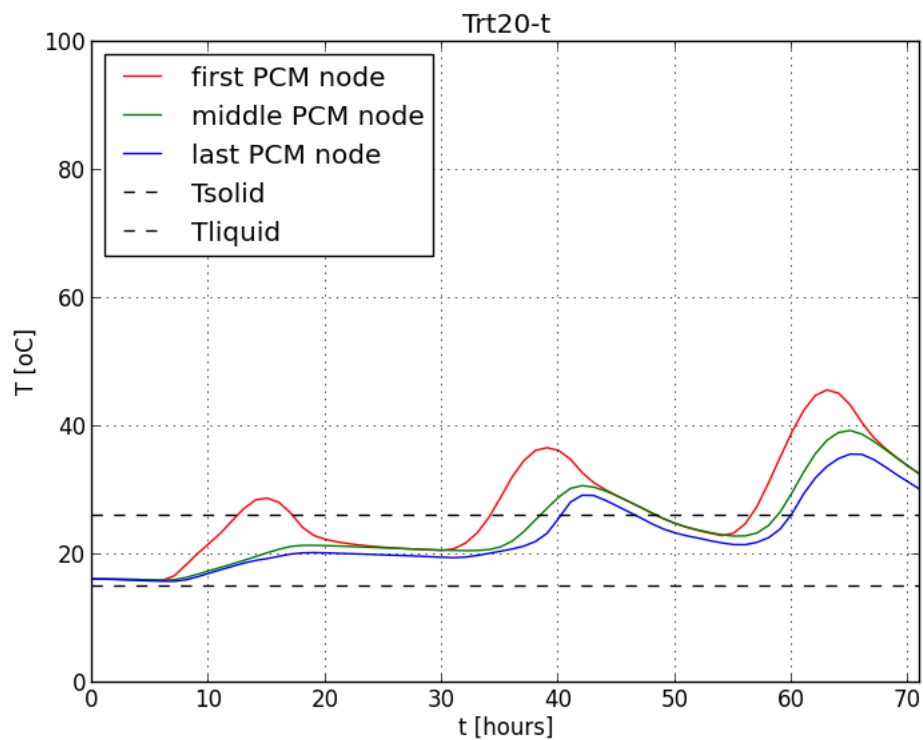


Διάγραμμα Β.7.11: Θερμοκρασίες στο εσωτερικό του SP25A8, 1-3/3.

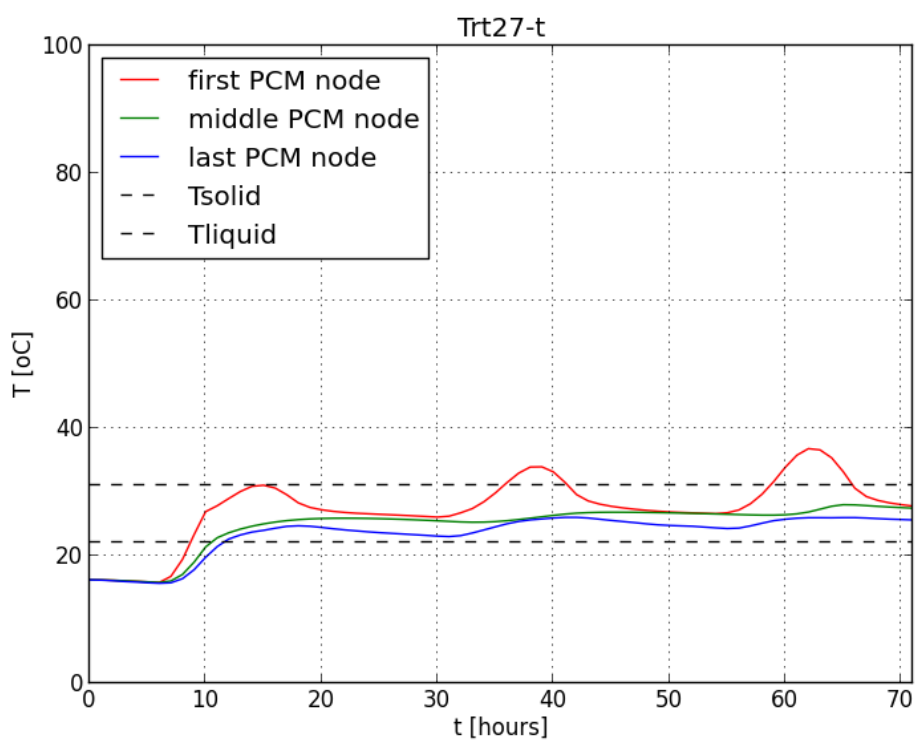


Διάγραμμα Β.7.12: Συγκριτικό διάγραμμα θερμοκρασιών κυψέλης για τις περιπτώσεις PV, PV-RT20, PV-RT27, PV-SP25A8, 1-3/3.

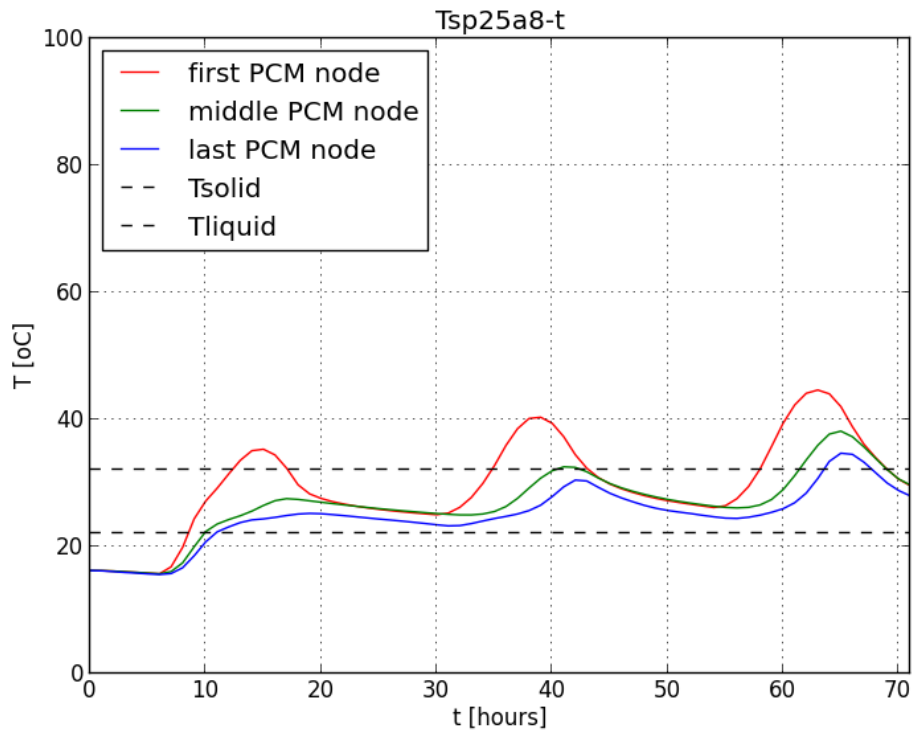
Περίοδος:1-3/4



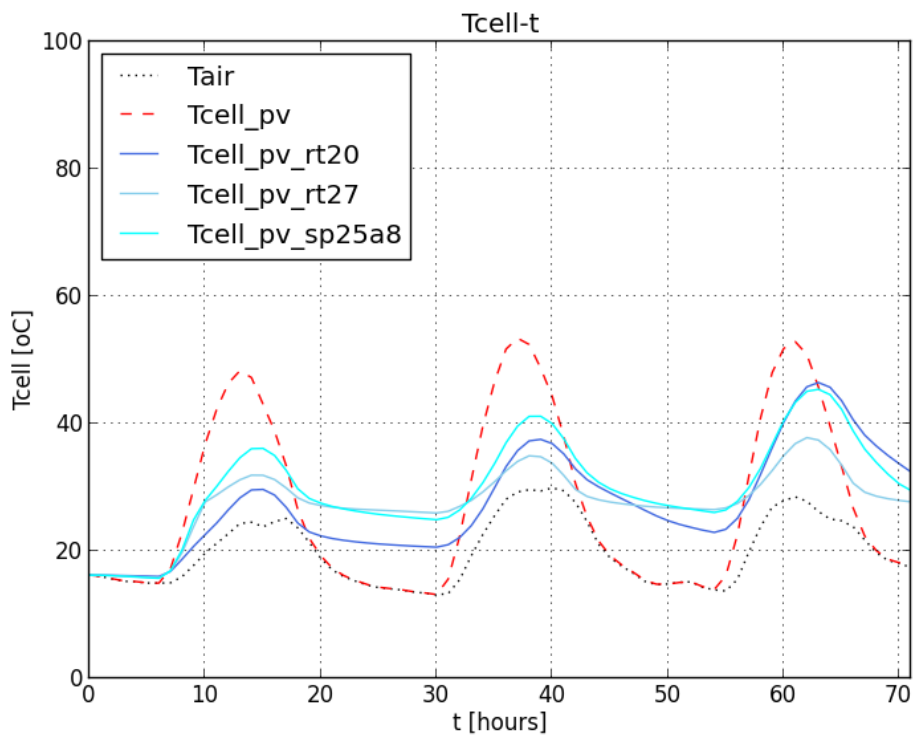
Διάγραμμα Β.7.13: Θερμοκρασίες στο εσωτερικό του RT20, 1-3/4.



Διάγραμμα Β.7.14: Θερμοκρασίες στο εσωτερικό του RT27, 1-3/4.

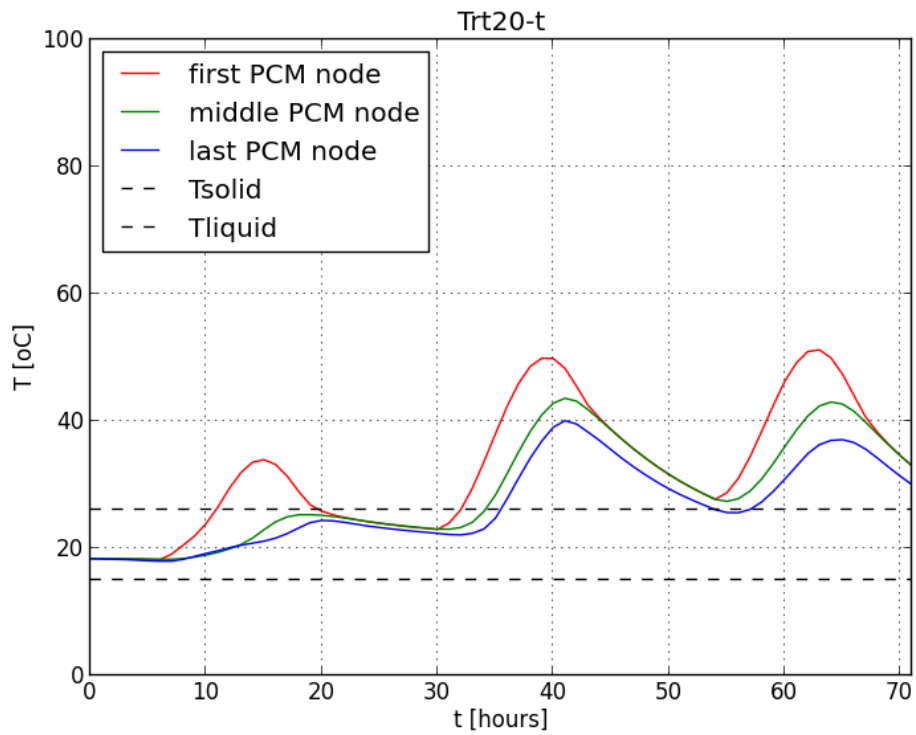


Διάγραμμα Β.7.15: Θερμοκρασίες στο εσωτερικό του SP25A8, 1-3/4.

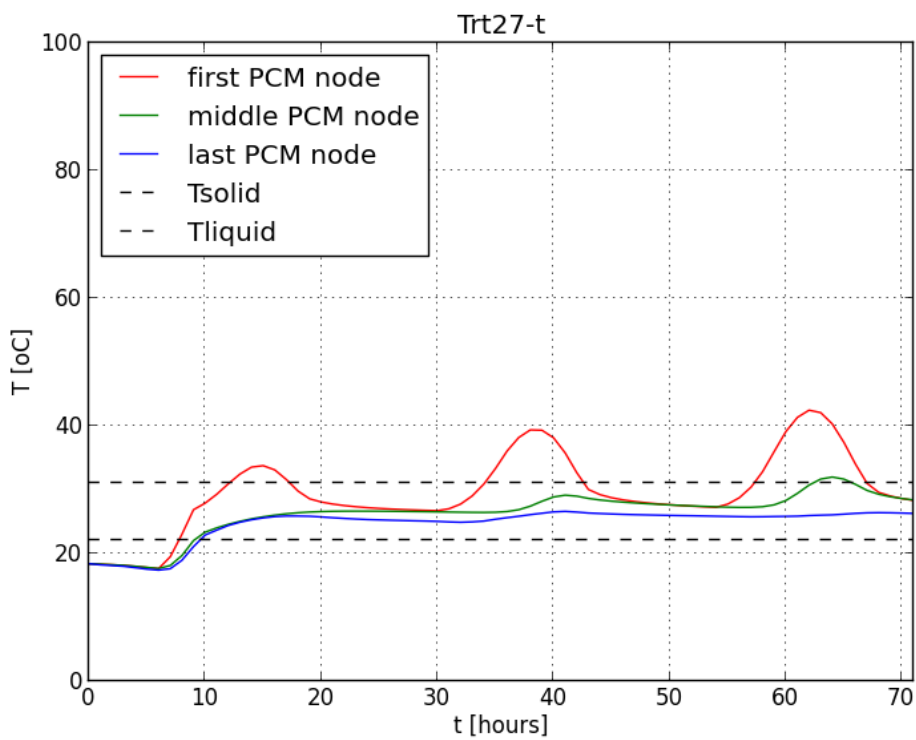


Διάγραμμα Β.7.16: Συγκριτικό διάγραμμα θερμοκρασιών κυψέλης για τις περιπτώσεις PV, PV-RT20, PV-RT27, PV-SP25A8, 1-3/4.

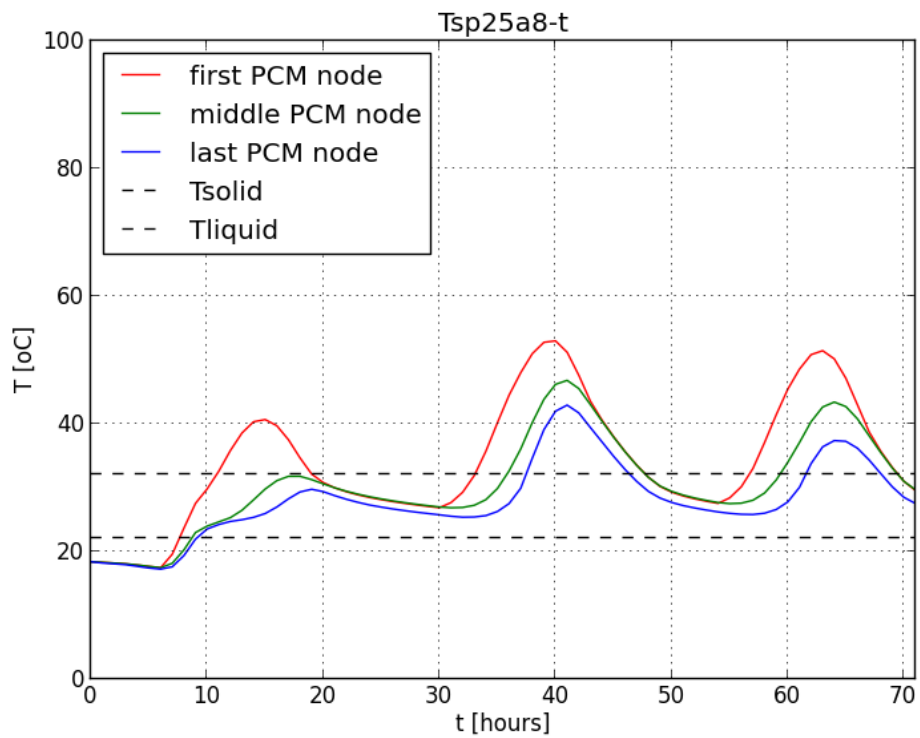
Περίοδος:1-3/5



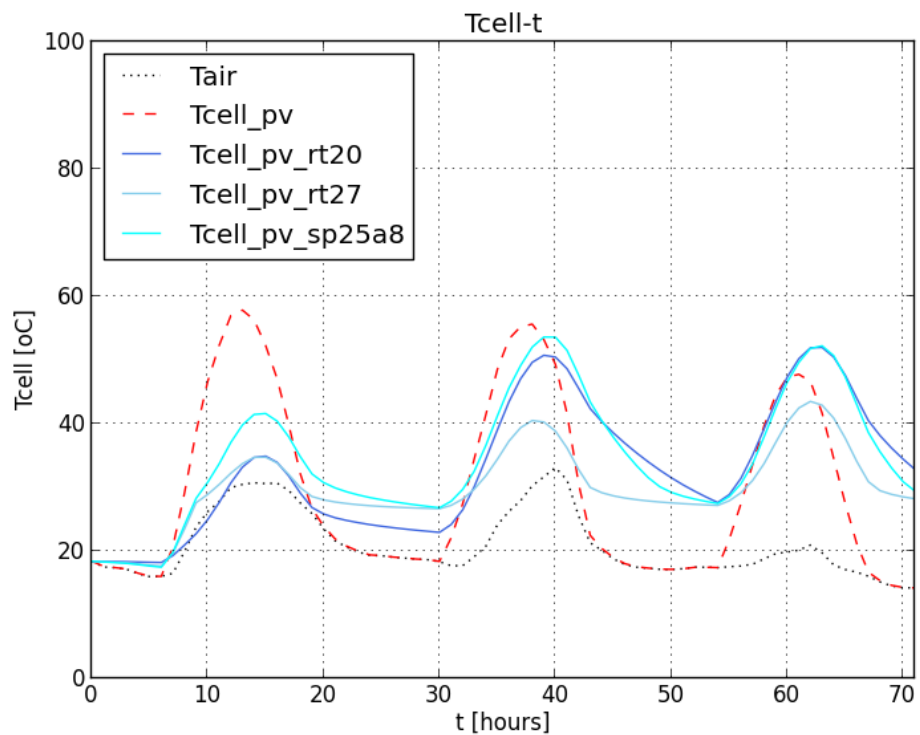
Διάγραμμα Β.7.17: Θερμοκρασίες στο εσωτερικό του RT20, 1-3/5.



Διάγραμμα Β.7.18: Θερμοκρασίες στο εσωτερικό του RT27, 1-3/5.

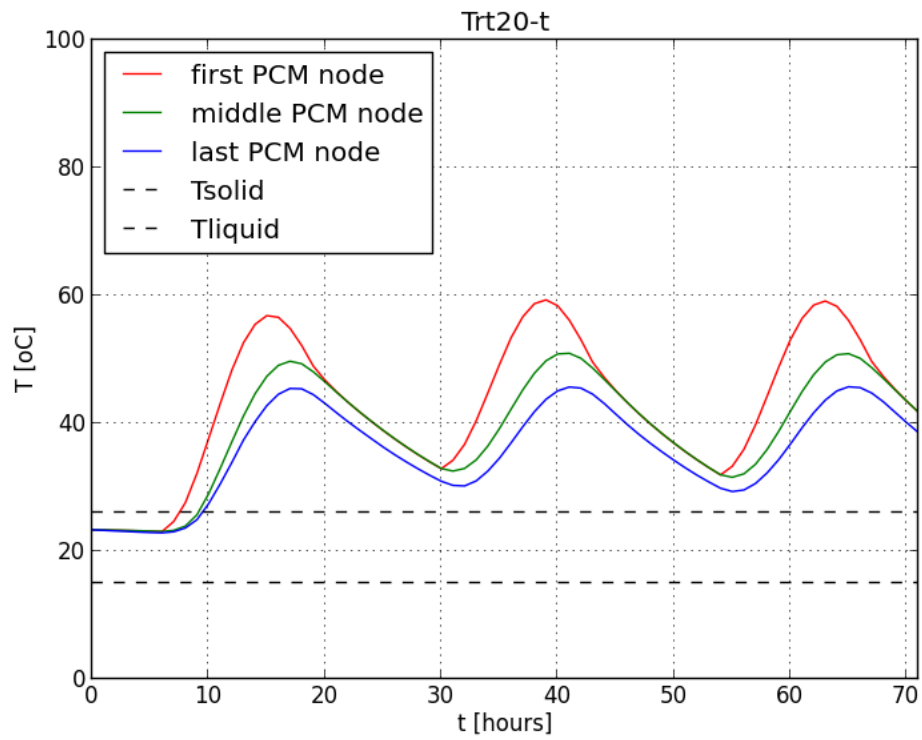


Διάγραμμα Β.7.19: Θερμοκρασίες στο εσωτερικό του SP25A8, 1-3/5.

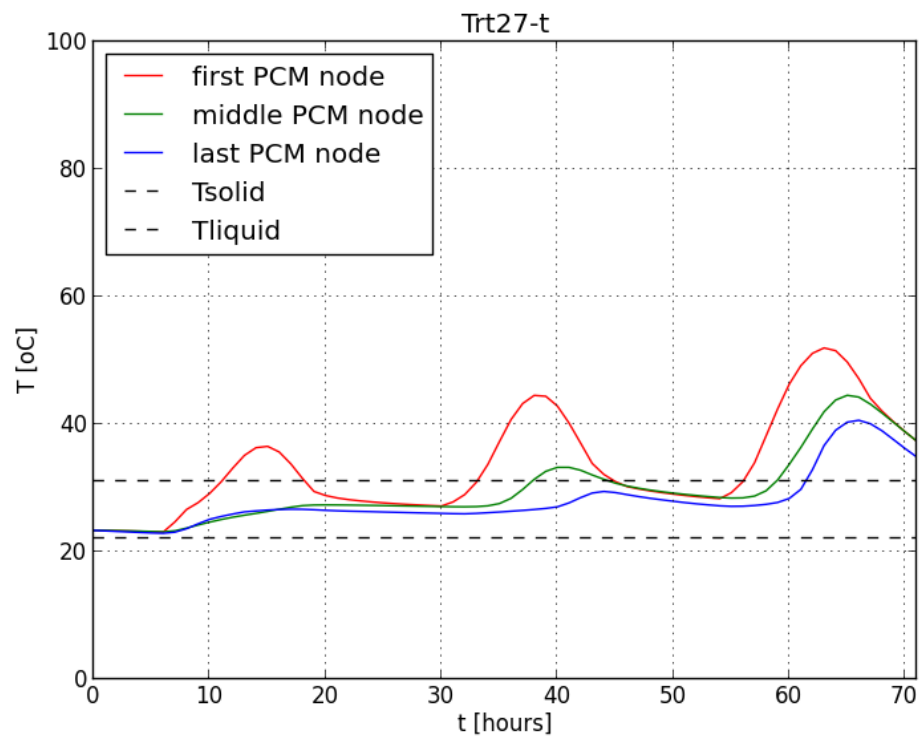


Διάγραμμα Β.7.20: Συγκριτικό διάγραμμα θερμοκρασιών κυψέλης για τις περιπτώσεις PV, PV-RT20, PV-RT27, PV-SP25A8, 1-3/5.

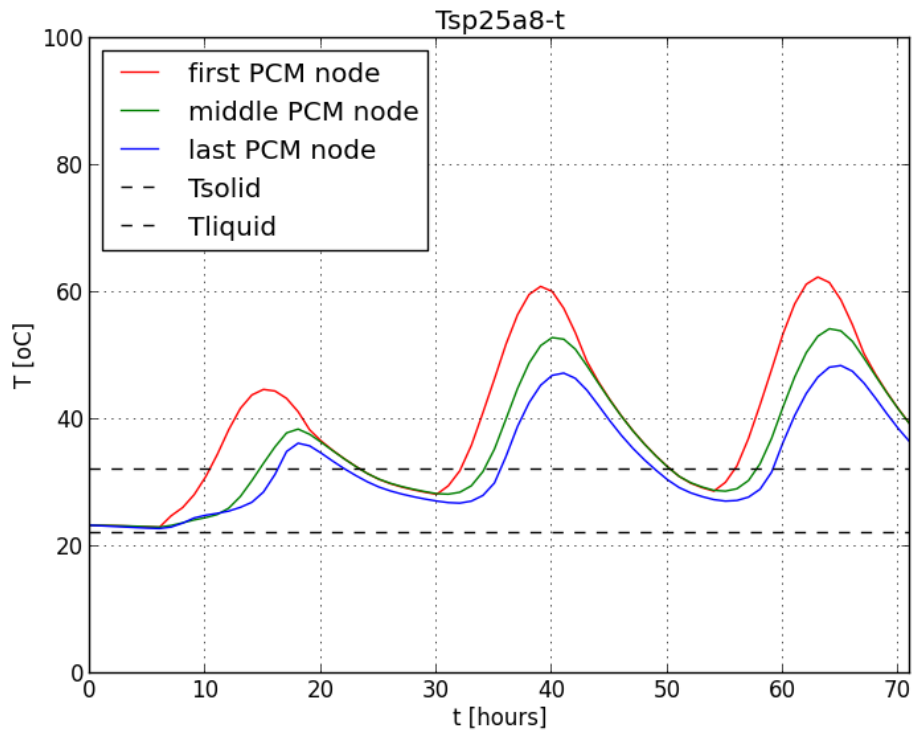
Περίοδος:1-3/6



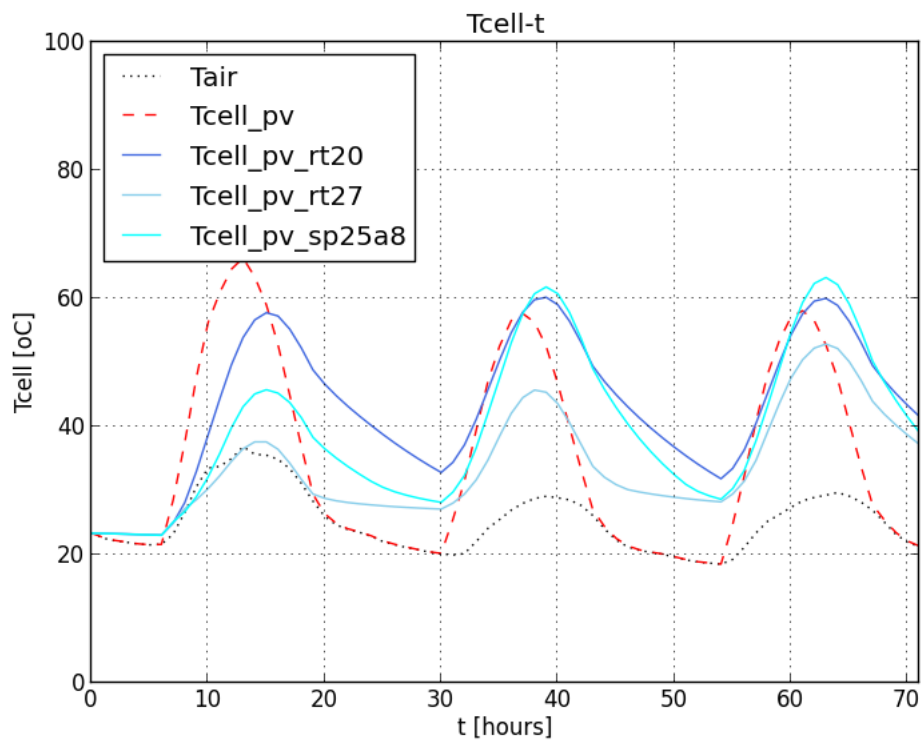
Διάγραμμα Β.7.21: Θερμοκρασίες στο εσωτερικό του RT20, 1-3/6.



Διάγραμμα Β.7.22: Θερμοκρασίες στο εσωτερικό του RT27, 1-3/6.

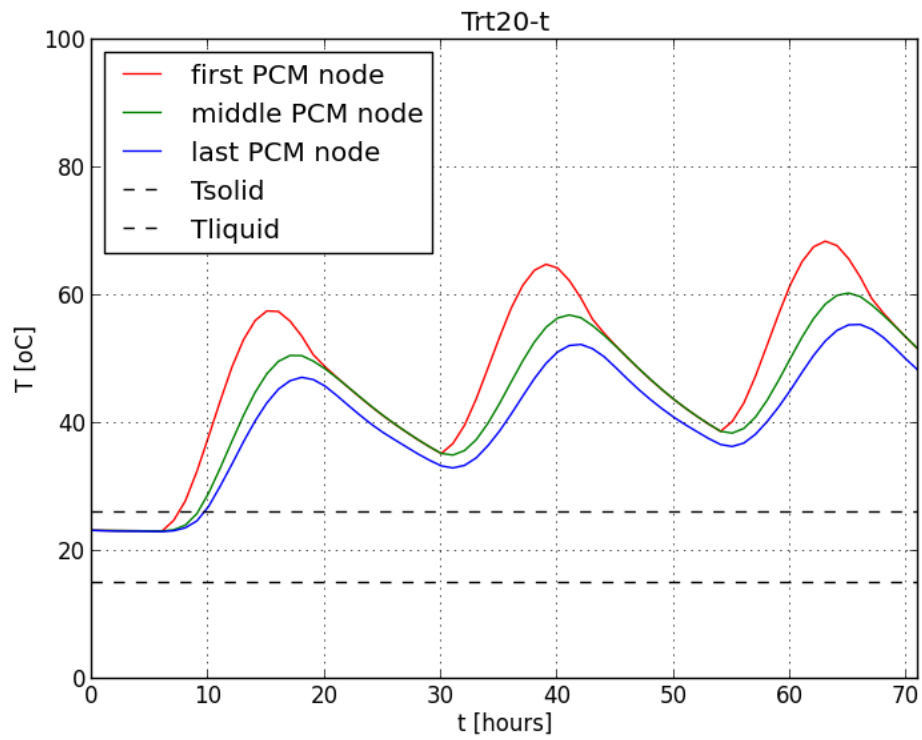


Διάγραμμα Β.7.23: Θερμοκρασίες στο εσωτερικό του SP25A8, 1-3/6.

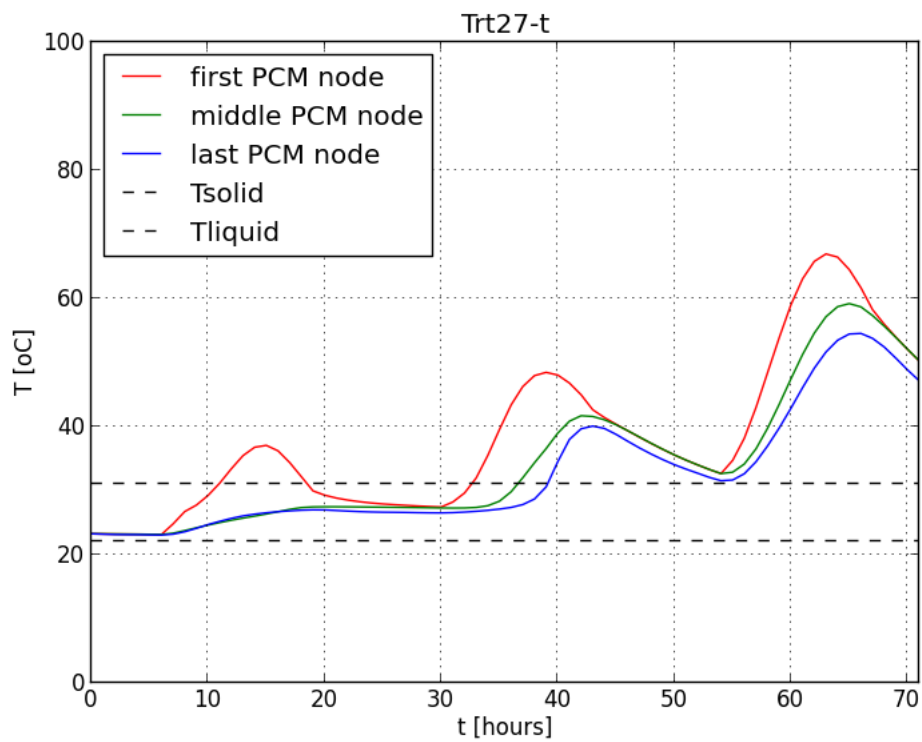


Διάγραμμα Β.7.24: Συγκριτικό διάγραμμα θερμοκρασιών κυψέλης για τις περιπτώσεις PV, PV-RT20, PV-RT27, PV-SP25A8, 1-3/6.

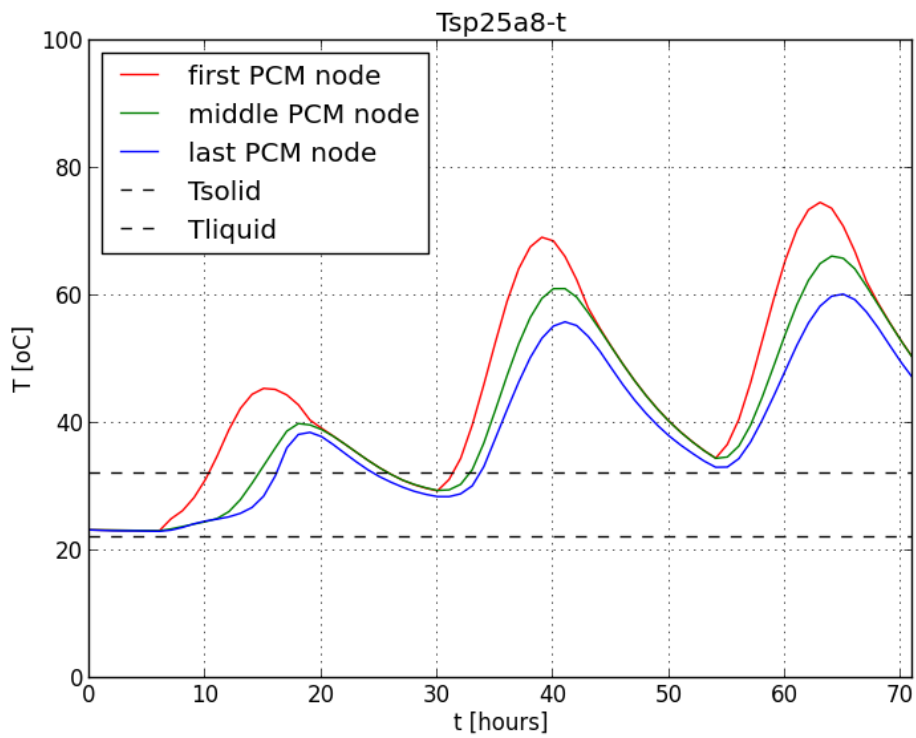
Περίοδος:1-3/7



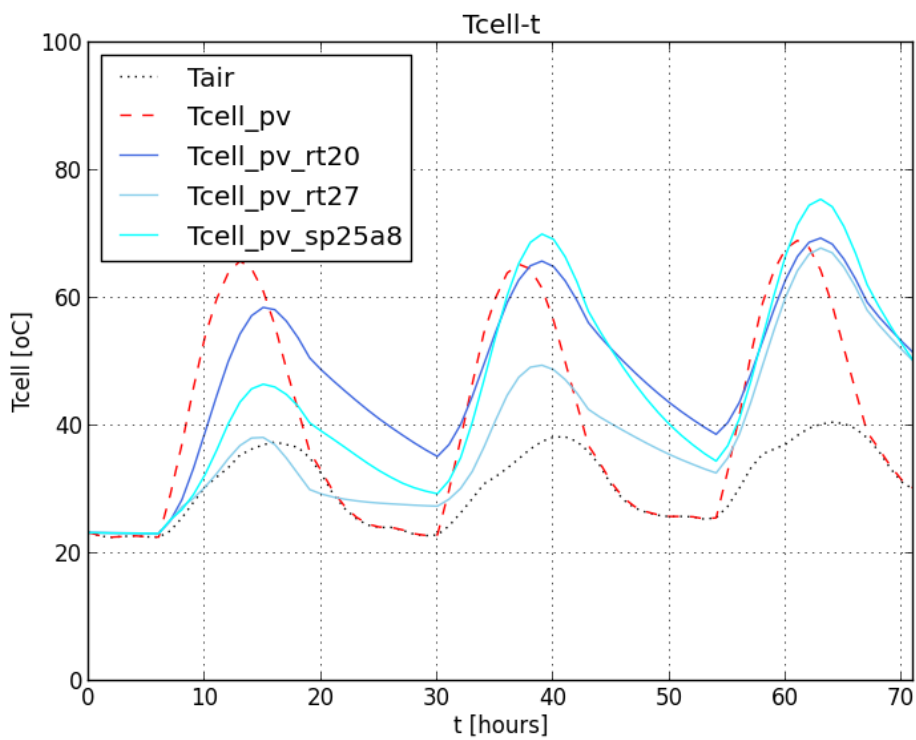
Διάγραμμα Β.7.25: Θερμοκρασίες στο εσωτερικό του RT20, 1-3/7.



Διάγραμμα Β.7.26: Θερμοκρασίες στο εσωτερικό του RT27, 1-3/7.

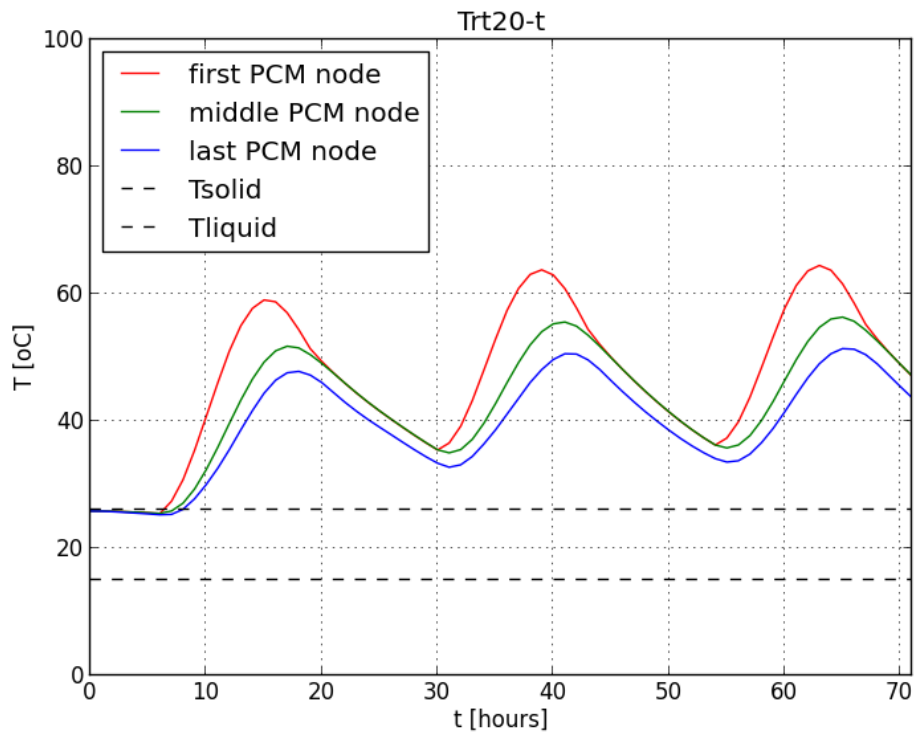


Διάγραμμα Β.7.27: Θερμοκρασίες στο εσωτερικό του SP25A8, 1-3/7.

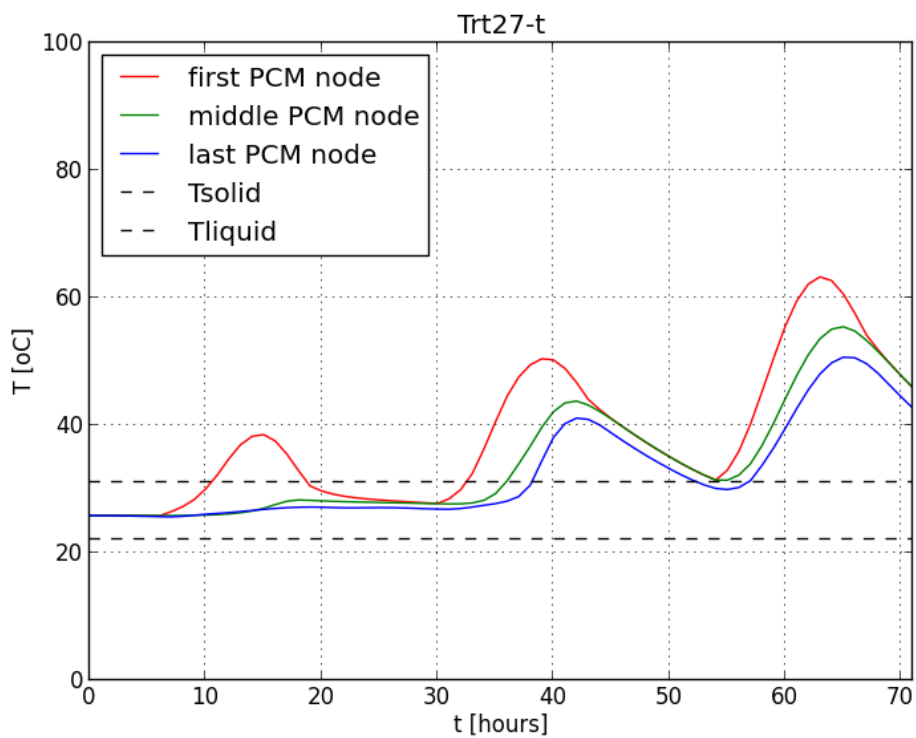


Διάγραμμα Β.7.28: Συγκριτικό διάγραμμα θερμοκρασιών κυψέλης για τις περιπτώσεις PV, PV-RT20, PV-RT27, PV-SP25A8, 1-3/7.

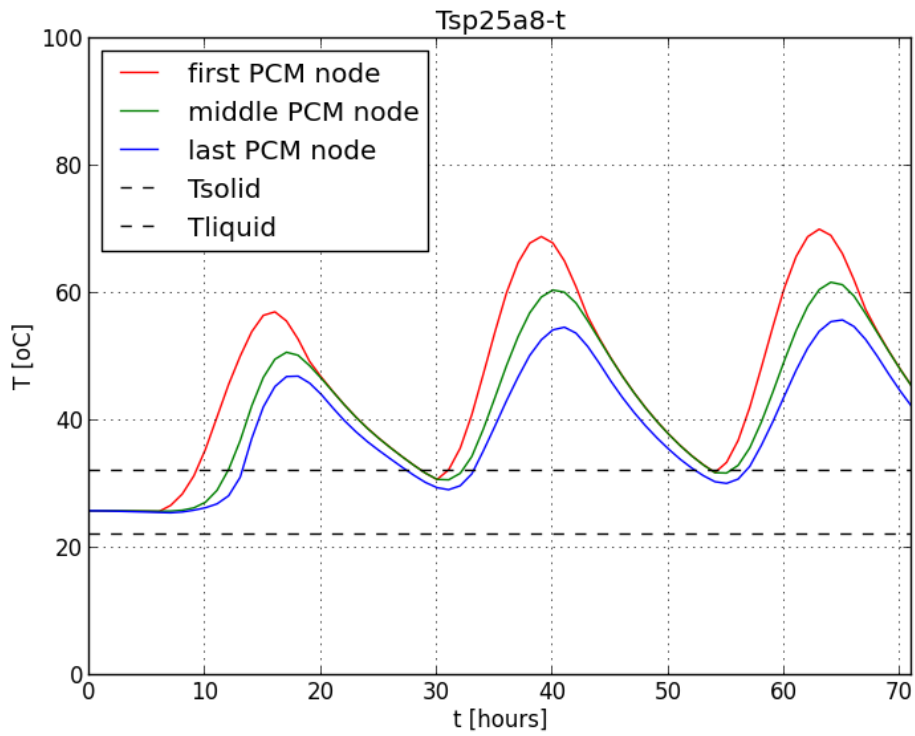
Περίοδος:1-3/8



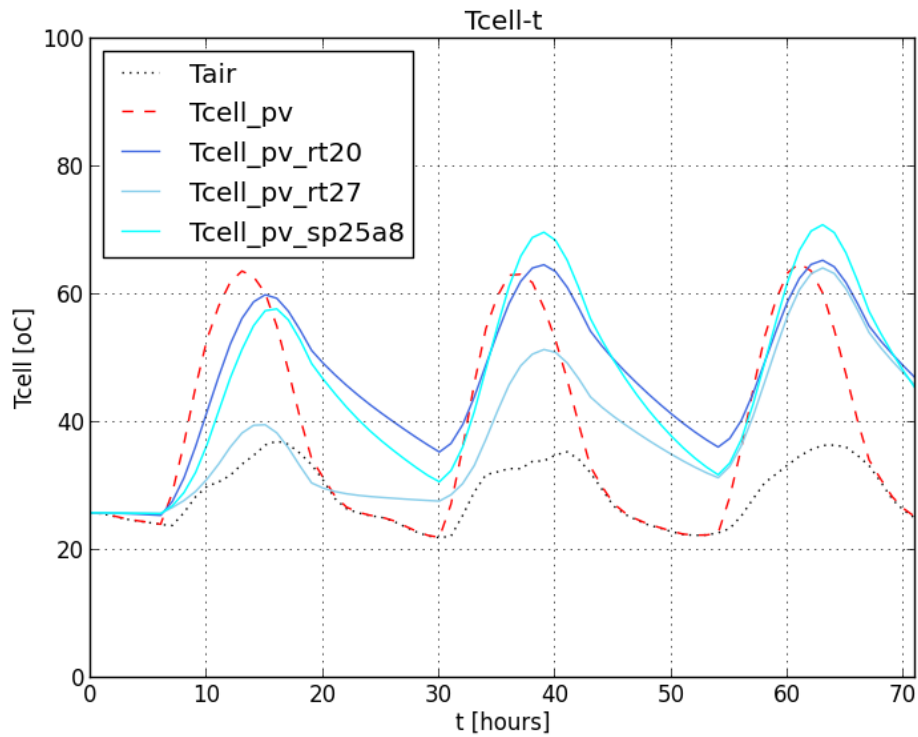
Διάγραμμα Β.7.29: Θερμοκρασίες στο εσωτερικό του RT20, 1-3/8.



Διάγραμμα Β.7.30: Θερμοκρασίες στο εσωτερικό του RT27, 1-3/8.



Διάγραμμα Β.7.31: Θερμοκρασίες στο εσωτερικό του SP25A8, 1-3/8.



Διάγραμμα Β.7.32: Συγκριτικό διάγραμμα θερμοκρασιών κυψέλης για τις περιπτώσεις PV, PV-RT20, PV-RT27, PV-SP25A8, 1-3/8.

B.8-ΣΥΜΠΕΡΑΣΜΑΤΑ

Στο σημείο αυτό θα ήταν χρήσιμο να υπενθυμίσουμε τις παραδοχές που έχουν γίνει. Η γεωμετρία της φωτοβολταϊκής μονάδα θεωρήθηκε απλή, δηλαδή ενισχυμένο γυαλί και πλαστικό υπόστρωμα (PET panel) πάνω στο οποίο βρίσκονται εντυπωμένες οι ηλιακές κυψέλες. Λόγω της γεωμετρικής απλότητας και του μεγάλου λόγου πάχους/εμπρόσθιας επιφάνειας της μονάδας (της τάξης του 0.03 m^{-1}), θεωρήθηκε μονοδιάστατη μεταφορά θερμότητας. Στην περίπτωση του συστήματος PV-PCM, η επαφή μεταξύ υποστρώματος και πλαστικής συσκευασίας PCM θεωρήθηκε τέλεια. Με άλλα λόγια, τυχόν στρώμα αέρα μεταξύ τους λόγω ατελούς επαφής αγνοήθηκε. Οι εξισώσεις που περιγράφουν το πρόβλημα λύθηκαν με τη μέθοδο πεπερασμένων διαφορών και συγκεκριμένα με την άρρητη έκφραση της μεθόδου. Ο λόγος που προτιμήθηκε η άρρητη έκφραση οφείλεται στην άνευ όρων ευστάθεια της μεθόδου και στα επιτρεπόμενα μεγάλα χρονικά βήματα. Παρά το γεγονός ότι οδηγεί στην επίλυση συστήματος αλγεβρικών εξισώσεων, κάτι που αυξάνει τις προγραμματιστικές απαιτήσεις, η τριδιαγώνια μορφή του συστήματος και η δυνατότητα λήψης μεγάλων χρονικών βημάτων, dt , οδηγούν σε εξοικονόμηση υπολογιστικού χρόνου [13]. Επειδή το σύστημα PV-PCM αποτελείται από πολλά στρώματα η διακριτοποίηση χώρου σε κάθε στρώμα έχει ληφθεί έτσι ώστε κανένας κόμβος να μη βρίσκεται σε διεπιφάνεια αλλαγής υλικού (σχήματα A.4.3 και A.7.2). Σε αντίθετη περίπτωση, η εξίσωση θερμότητας δεν ερμηνεύεται φυσικά (οι συντελεστές που περιγράφουν τις θερμοφυσικές ιδιότητες του κόμβου αφορούν ταυτόχρονα δύο υλικά). Χρησιμοποιήθηκαν περίπου 5 κόμβοι ανά mm υλικού, ήτοι $n_1=15$ (γυαλί), $n_2=5$ (PET), $n_3=5$ (πλαστική συσκευασία), $n_4=250$ (PCM), $n_5=5$ (πλαστική συσκευασία) και χρονικό βήμα $dt=3600 \text{ s}$. Τέλος, σε ό,τι αφορά τις απώλειες θερμότητας λόγω ακτινοβολίας και συναγωγής, υποτέθηκε για λόγους απλότητας συνδυαστικός συντελεστής μεταφοράς θερμότητας, ο οποίος λαμβάνει υπ' όψη και τους δύο μηχανισμούς μεταφοράς θερμότητας. Ο προσδιορισμός της συνάρτησης της ειδικής συναγωγιμότητας h_c είναι αποτέλεσμα πολύπλοκης μαθηματικής ή πειραματικής διαδικασίας [9]. Παράλληλα, η εισαγωγή άγνωστης θερμοκρασίας στη τέταρτη δύναμη (εξίσωση θερμότητας λόγω ακτινοβολίας) θα μετέτρεπε το σύστημα σε μη γραμμικό. Για τις ανάγκες της παρούσας εργασίας ο συνδυαστικός αυτός συντελεστής μεταφοράς θερμότητας λόγω ελεύθερης συναγωγής και ακτινοβολίας θεωρήθηκε ίδιος για άνω και κάτω επιφάνεια και σταθερός για κάθε χρονική περίοδο: $h=h_r+h_c=4 \text{ W/m}^2\text{K}$ (B.4).

Η μοντελοποίηση των υλικών αλλαγής φάσης βασίστηκε στη μέθοδο της ενεργού θερμοχωρητικότητας και συγκεκριμένα στη δημοσίευση [17]. Στην δημοσίευση αυτή, προσδιορίστηκαν πειραματικά οι συναρτήσεις των ενεργών θερμοχωρητικοτήτων πέντε υλικών της "οικογένειας" των παραφινών. Λαμβάνοντας υπόψη τα λειτουργικά όρια των υλικών, σε συνδυασμό το κλίμα της Αθήνας (περίπου $10-40 \text{ }^\circ\text{C}$ κατά τη διάρκεια του έτους, B.1) και το γεγονός ότι οι φωτοβολταϊκές μονάδες αποδίδουν καλύτερα κάτω από θερμοκρασία κυψέλης $T_{\text{cell}}=25 \text{ }^\circ\text{C}$, επιλέχθηκαν τα εξής υλικά: RT20 ($15-26 \text{ }^\circ\text{C}$), RT27 ($22-31 \text{ }^\circ\text{C}$) και SP25A8 ($22-32 \text{ }^\circ\text{C}$). Από το διάγραμμα B.4.1 το υλικό που παρουσιάζει την υψηλότερη ενεργό θερμοχωρητικότητα κατά την αλλαγή φάσης είναι το RT27. Τα διαγράμματα του κεφαλαίου B.7 (διαγράμματα B.7.1-B.7.32) μας δίνουν μια συγκριτική εικόνα των θερμοκρασιακών επιπέδων της ηλιακής κυψέλης σε κάθε διακριτή χρονική περίοδο, καθώς και τα θερμοκρασιακά επίπεδα που λαμβάνουν χώρα εντός των υλικών αλλαγής φάσης, δηλαδή το κατά πόσο γίνονται εκμεταλλεύσιμα. Παρατηρούμε ότι, σε διαφορετικές εποχές, διαφορετικά υλικά αποδίδουν καλύτερα.

Για παράδειγμα, στην πρώτη χρονική περίοδο (1-3/1) το υλικό που αποδίδει καλύτερα και παρουσιάζει χαμηλότερα θερμοκρασιακά επίπεδα είναι το RT20 (διάγραμμα B.7.4). Αυτό αντικατοπτρίζεται στα διαγράμματα B.7.1, B.7.2 και B.7.3. Τα διαγράμματα αυτά δείχνουν τη θερμοκρασία του πρώτου, του τελευταίου και ενός ενδιάμεσου κόμβου στο εσωτερικό του κάθε PCM. Με βάση αυτά το RT20 γίνεται περισσότερο εκμεταλλεύσιμο, καθώς εισέρχεται στη διφασική περιοχή και εξέρχεται αυτής σχεδόν εξ' ολοκλήρου σε κάθε 24-ωρο. Όμοια, χρησιμοποιώντας τα διαγράμματα του κεφαλαίου B.7 (διαγράμματα B.7.1-B.7.32), μπορούμε να

συμπεράνουμε πιο υλικό αποδίδει σε κάθε περίπτωση καλύτερα.

Πιο συγκεκριμένα, για τις περιόδους 1-3/1, 1-3/2 και 1-3/3 το υλικό RT20 παρέχει χαμηλότερες θερμοκρασίες κυψέλης (διαγράμματα B.7.1-B.7.12). Την περίοδο 1-3/4 (διαγράμματα B.7.13-B.7.16), αν και το RT20 ξεκινά από χαμηλότερες θερμοκρασίες, το RT27 αποδίδει συνολικά καλύτερα. Αυτό αντικατοπτρίζεται και από το διάγραμμα B.7.13. Μετά από κάποιο σημείο το RT20 παραμένει ρευστό, δηλαδή δε γίνεται εκμεταλλεύσιμο. Ομοίως συμβαίνει και την περίοδο 1-3/5. Παρά το γεγονός ότι το λειτουργικό θερμοκρασιακό εύρος του SP25A8 είναι παρόμοιο με του RT27, αποδίδει χειρότερα, κάτι που οφείλεται στη σχεδόν διπλάσια πυκνότητα του (πίνακας A.6.1) και στη χαμηλότερη ενεργό θερμοχωρητικότητα (διάγραμμα B.5.1). Στις υπόλοιπες περιόδους (1-3/6, 1-3/7 και 1-3/8) το RT27 επικρατεί σε όρους απόδοσης (διαγράμματα B.7.17-B.7.32). Παρατηρείται όμως το εξής φαινόμενο: σε βάθος χρόνου και καθώς η θερμοκρασία του αέρα ξεπερνά τα λειτουργικά όρια των τριών υλικών (26, 31 και 32 °C), τα επίπεδα των θερμοκρασιών σταδιακά αυξάνονται. Ειδικά τις θερινές περιόδους (1-3/6, 1-3/7 και 1-3/8) η θερμοκρασία κυψέλης του συστήματος PV-PCM καταλήγει να είναι υψηλότερη σε σχέση με τη φωτοβολταϊκή μονάδα χωρίς PCM. Αυτό αντικατοπτρίζεται και στα διαγράμματα θερμοκρασίας στο εσωτερικό των PCM, καθώς δεν επανέρχονται στη στερεή φάση. Το γεγονός αυτό σε συνδυασμό με τη χαμηλή θερμική αγωγιμότητα των υλικών (RT20 και RT27: 0.2 W/mK, SP25A8: 0.6 W/mK) τα κάνει να λειτουργούν, κατά κάποιο τρόπο, ως θερμικές μονώσεις. Αυτό θα μπορούσε να αντιμετωπιστεί εν μέρει με εξαναγκασμένη συναγωγή ή με την είσοδο μεταλλικού πλέγματος μορφής κυψέλης στο εσωτερικό των PCM, κάτι που θα βελτίωνε συνολικά την θερμική αγωγιμότητα και, κατά συνέπεια, το ρυθμό αποβολής της αποθηκευμένης θερμότητας.

Γ-ΚΩΔΙΚΑΣ PYTHON

Γ.1-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΑΕΡΑ

meteorological_data.py

```
#####  
#1.)COMMENTS  
  
#Module that produces hourly air temperature and insolation files (txt format),  
#based on historic data acquired from:  
#http://openmeteo.org/stations/d/1334/  
#####  
#2.)MODULE IMPORTATION  
  
import numpy as np  
import math  
import matplotlib.pyplot as plt  
#####  
#3.)DATA  
  
#Parameters  
startingdate='2000-01-01' #starting date, year-month-day format  
period=3 #total period of time in days  
m=period*24 #total number of hours  
  
tarray=np.linspace(0,m-1,num=m,endpoint=True).reshape(-1,1) #temporal array  
  
#Creating blank arrays to store data  
Tairarray=np.zeros((m,1),dtype=float)  
Garray=np.zeros((m,1),dtype=float)  
#-----#  
#Air temperature file, Tair  
f=open('hourly_temperature.txt','r')  
  
startingpoint=0  
linecounter=0  
  
for line in f: #finding out where desired data begin  
    startingpoint=startingpoint+1  
    words=line.split(',')  
    if startingdate in words[0]:  
        break  
  
f.seek(0,0) #rewind' file  
  
for line in f: #storing desired data  
    linecounter=linecounter+1  
    if linecounter>=startingpoint and linecounter<=startingpoint+(m-1):  
        data=line.split(',')
```

```

    Tairarray[linecounter-startingpoint,0]=data[1]

f.close()
#-----#
#Insolation file, G
g=open('hourly_insolation.txt','r')

startingpoint=0
linecounter=0

for line in g: #finding out where desired data begin
    startingpoint=startingpoint+1
    words=line.split(',')
    if startingdate in words[0]:
        break

g.seek(0,0) #rewind' file

for line in g: #storing desired data
    linecounter=linecounter+1
    if linecounter>=startingpoint and linecounter<=startingpoint+(m-1):
        data=line.split(',')
        Garray[linecounter-startingpoint,0]=data[1]

g.close()
#-----#
#"""
#Tair plot
plt.figure(1)
plt.axis([0.0,m-1,0.0,50.0])
plt.title('Tair-t')
plt.xlabel('t [hours]')
plt.ylabel('Tair [oC]')
plt.plot(tarray,Tairarray,color='black')
plt.grid(True)
plt.savefig('air_temperature_over_time.png')
plt.show()
#"""
#"""
plt.figure(2)
plt.axis([0.0,m-1,0.0,1.0])
plt.title('G-t')
plt.xlabel('t [hours]')
plt.ylabel('G [kW/m2]')
plt.plot(tarray,Garray,color='red')
plt.grid(True)
plt.savefig('insolation_over_time.png')
plt.show()
#"""

```

```
#Saving arrays in txt format
np.savetxt('garray.txt',Garray)
np.savetxt('tairarray.txt',Tairarray)
#*****#
```

Γ.2-ΥΠΟΛΟΓΙΣΜΟΣ ΣΤΙΓΜΙΑΙΑΣ ΟΛΙΚΗΣ ΗΛΙΑΚΗΣ ΑΚΤΙΝΟΒΟΛΙΑΣ
ΚΕΚΛΙΜΕΝΟΥ ΕΠΙΠΕΔΟΥ

insolation.py

```
#####  
#1.)COMMENTS  
#  
#Module that calculates total insolation, GT [W/m2] of a tilted surface, based  
#on statistically determined equations.  
#For different periods of time, lines 20-21 are to be changed accordingly.  
#  
#####nighttime#####|#####daytime#####|###nighttime###|day##  
#00 01 02 03 04 05 06|07 08 09 10 11 12 13 14 15 16 17 18|19 20 21 22 23|n  
#24 25 26 27 28 29 30|31 32 33 34 35 36 37 38 39 40 41 42|43 44 45 46 47|n+1  
#48 49 50 51 52 53 54|55 56 57 58 59 60 61 62 63 64 65 66|67 68 69 70 71|n+2  
#...  
#  
#Serial number n of days of the year  
#January February March April May June  
#i 31+i 59+i 90+i 120+i 151+i  
#July August September October November December  
#181+i 212+i 243+i 273+i 304+i 334+i  
#  
#NOTES:temp*:temporary values/math's trigonometric functions receive angles in  
#rads rather than degrees.  
#####  
#2.)MODULE IMPORTATION  
  
import numpy as np  
import math  
import matplotlib.pyplot as plt  
#####  
#3.)DATA  
  
firstday=1 #selected day:1-365 <--Setting starting day  
period=3 #selected period of days <--Setting period of days  
m=period*24 #total number of hours  
  
fi=37.97 #f [degrees],geographic latitude of Athens  
beta=fi #b [degrees],surface angle,maximum solar insolation (fi=b)  
gamma=0.0 #g [degrees],azimuth,surface facing south  
ro=0.2 #r [-],ground reflectivity  
  
#Reading thermal radiation coefficients from file  
f=open("insolation_coefficients.txt","rb")  
t=np.genfromtxt(f,dtype=float,skiprows=1,usecols=0).reshape(12,-1)  
f.seek(0,0) #rewind  
Mt=np.genfromtxt(f,dtype=float,skiprows=1,usecols=1).reshape(12,-1)  
f.seek(0,0) #rewind
```

```

At=np.genfromtxt(f,dtype=float,skiprows=1,usecols=2).reshape(12,-1)
f.seek(0,0) #rewind
Pt=np.genfromtxt(f,dtype=float,skiprows=1,usecols=3).reshape(12,-1)
f.seek(0,0) #rewind
mt=np.genfromtxt(f,dtype=float,skiprows=1,usecols=4).reshape(12,-1)
f.seek(0,0) #rewind
at=np.genfromtxt(f,dtype=float,skiprows=1,usecols=5).reshape(12,-1)
f.seek(0,0) #rewind
pt=np.genfromtxt(f,dtype=float,skiprows=1,usecols=6).reshape(12,-1)
f.seek(0,0) #rewind

```

#Creating blank arrays

```

tarray=np.zeros((m,1),dtype=float)
Mtarray=np.zeros((m,1),dtype=float)
Atarray=np.zeros((m,1),dtype=float)
Ptarray=np.zeros((m,1),dtype=float)
mtarray=np.zeros((m,1),dtype=float)
atarray=np.zeros((m,1),dtype=float)
ptarray=np.zeros((m,1),dtype=float)

```

```

Gbarray=np.zeros((m,1),dtype=float)
Gdarray=np.zeros((m,1),dtype=float)

```

```

deltaarray=np.zeros((m,1),dtype=float)

```

```

omegaarray=np.zeros((m,1),dtype=float)

```

```

cosfiarray=np.zeros((m,1),dtype=float)
cosfizarray=np.zeros((m,1),dtype=float)

```

```

Rbarray=np.zeros((m,1),dtype=float)

```

```

GTarray=np.zeros((m,1),dtype=float)

```

```

#*****#

```

#3.)MAIN CODE

#Creating arrays

```

timearray=np.linspace(0,m,num=m,endpoint=False).reshape(-1,1) #temporal array

```

#Creating day array

```

dayarray=np.zeros((m,1),dtype=float)

```

```

for i in range(0,period): #day loop
    for j in range(0,24): #hour loop
        k=j+(i*24)
        dayarray[k,0]=firstday+(i*1)

```

#Creating thermal radiation coefficient arrays

```

for i in range(0,period):

```

```

for j in range(0,24):
    if j>=7 and j<=18:
        k=j+(i*24)
        tarray[k,0]=t[j-7,0]
        Mtarray[k,0]=Mt[j-7,0]
        Aarray[k,0]=At[j-7,0]
        Ptarray[k,0]=Pt[j-7,0]
        mtarray[k,0]=mt[j-7,0]
        aarray[k,0]=at[j-7,0]
        ptarray[k,0]=pt[j-7,0]

```

```

#Calculating average values of direct insolation Gb [kW/m2] and
#diffuse insolation Gd [kW/m2], for horizontal surfaces
#math.sin receives rads as input

```

```

for p in range(0,m):
    temp1=(((360.0/365.0)*dayarray[p,0])-Ptarray[p,0])*(math.pi/180.0)
    Gbarray[p,0]=Mtarray[p,0]+(Aarray[p,0]*math.sin(temp1))
    temp2=(((360.0/365.0)*dayarray[p,0])-ptarray[p,0])*(math.pi/180.0)
    Gdarray[p,0]=mtarray[p,0]+(aarray[p,0]*math.sin(temp2))
    (((360.0/365.0)*dayarray[p,0])-Ptarray[p,0])*(math.pi/180)

```

```

#-----
#Next part "ignores" negative parts of the curves

```

```

for p in range(0,m):
    if Gbarray[p,0]<0.0:
        Gbarray[p,0]=0.0
    if Gdarray[p,0]<0.0:
        Gdarray[p,0]=0.0

```

```

#-----

```

```

#Calculating solar declination, delta [degrees]

```

```

for p in range(0,m):
    temp3=(((360.0/365.0)*(284+dayarray[p,0]))*(math.pi/180.0)
    deltaarray[p,0]=23.45*math.sin(temp3)

```

```

#Calculating solar hour angle, omega [degrees]

```

```

for i in range(0,period): #day loop
    for j in range(0,24): #hour loop
        k=j+(i*24)
        omegaarray[k,0]=0.25*60*(j-12)

```

```

#Calculating cosinefi, cosinefiz

```

```

for p in range(0,m):
    sinfi=math.sin(fi*(math.pi/180.0)) #sin(f)
    cosfi=math.cos(fi*(math.pi/180.0)) #cos(f)
    sinbeta=math.sin(beta*(math.pi/180.0)) #sin(b)
    cosbeta=math.cos(beta*(math.pi/180.0)) #cos(b)
    singamma=math.sin(gamma*(math.pi/180.0)) #sin(g)
    cosgamma=math.cos(gamma*(math.pi/180.0)) #cos(g)
    sindelta=math.sin(deltaarray[p,0]*(math.pi/180.0)) #sin(d)

```

```

cosdelta=math.cos(deltaarray[p,0]*(math.pi/180.0)) #cos(d)
sinomega=math.sin(omegaarray[p,0]*(math.pi/180.0)) #sin(o)
cosomega=math.cos(omegaarray[p,0]*(math.pi/180.0)) #cos(o)

cosfiarray[p,0]=(sindelta*sinfi*cosbeta)-\
    (sindelta*cosfi*sinbeta*cosgamma)+\
    (cosdelta*cosfi*cosbeta*cosomega)+\
    (cosdelta*sinfi*sinbeta*cosgamma*cosomega)+\
    (cosdelta*sinbeta*singamma*sinomega)

cosfizarray[p,0]=(sindelta*sinfi)+\
    (cosdelta*cosfi*cosomega)

#Calculating Rb=GbT/Gb ratio
#direct radiation of tilted surface to direct radiation of horizontal surface
for p in range(0,m):
    if tarray[p,0]!=0: #solar radiation only during daytime (7am-6pm)
        Rbarray[p,0]=cosfiarray[p,0]/cosfizarray[p,0]

#Calculating solar radiation of tilted surface, GT [W/m2].
for p in range(0,m):
    temp1=(1+math.cos(beta*(math.pi/180)))/2
    temp2=(1-math.cos(beta*(math.pi/180)))/2
    GTarray[p,0]=(Gbarray[p,0]*Rbarray[p,0])+ \
        (Gdarray[p,0]*temp1)+ \
        ((Gbarray[p,0]+Gdarray[p,0])*ro*temp2)

#Printing results
#"""
print('Gb:',Gbarray,Gbarray.shape)
print('Gd:',Gdarray,Gdarray.shape)
print('GT:',GTarray,GTarray.shape)
#"""

#Saving GT array to file
np.savetxt('insolation.txt',GTarray)

#Plotting chart
#"""
plt.figure(1)
plt.axis([0.0,m-1,0.0,1.0])
plt.xlabel('t [hour(s)]')
plt.ylabel('G [kW/m2]')
plt.title('G-t')
plt.plot(timearray,Gbarray,color='green',label='GbT')
plt.plot(timearray,Gdarray,color='blue',label='GdT')
plt.plot(timearray,GTarray,color='red',label='GT')
plt.legend(loc='upper right')
plt.savefig('insolation.png')
plt.show()

```

```

#""
#*****#
"""
print('*'*80)
print('Parameters')
print('Geographic latitude: fi=%.2f [degrees]' %(fi))
print('Incident angle: beta=%.2f [degrees]' %(beta))
print('Surface azimuth: gamma=%.2f [degrees]' %(gamma))
print('Ground reflectance: ro=%.2f [-]' %(ro))
print('-'*80)
print('period:%i, firstday: %i, total hours: %i' %(period,firstday,m))
print('*'*80)
"""
"""
print('*'*80)
print('t:',t,t.shape)
print('Mt:',Mt,Mt.shape)
print('At:',At,At.shape)
print('Pt:',Pt,Pt.shape)
print('mt:',mt,mt.shape)
print('at:',at,at.shape)
print('pt:',pt,pt.shape)
print('-'*80)
print('timearray:',timearray,timearray.shape)
print('dayarray:',dayarray,dayarray.shape)
print('*'*80)
"""
"""
print('*'*80)
print('tarray:',tarray,tarray.shape)
print('Mtarray:',Mtarray,Mtarray.shape)
print('Atarray:',Atarray,Atarray.shape)
print('Ptarray:',Ptarray,Ptarray.shape)
print('mtarray:',mtarray,mtarray.shape)
print('atarray:',atarray,atarray.shape)
print('ptarray:',ptarray,ptarray.shape)
print('-'*80)
print('deltaarray:',deltaarray,deltaarray.shape)
print('omegaarray:',omegaarray,omegaarray.shape)
print('cosfiarray:',cosfiarray,cosfiarray.shape)
print('cosfizarray:',cosfizarray,cosfizarray.shape)
print('Rbarray:',Rbarray,Rbarray.shape)
print('-'*80)
print('Gb:',Gbarray,Gbarray.shape)
print('Gd:',Gdarray,Gdarray.shape)
print('GT:',GTarray,GTarray.shape)
print('*'*80)
"""
#*****#

```


insolation_coefficients.txt

t[hrs]	Mt[kW/m2]	At[kW/m2]	Pt[degrees]	mt[kW/m2]	at[kW/m2]	pt[degrees]
07.00	0.007	0.046	86.18	0.048	0.111	81.23
08.00	0.057	0.068	87.31	0.134	0.127	82.45
09.00	0.119	0.104	87.94	0.208	0.133	83.94
10.00	0.191	0.129	88.31	0.261	0.129	85.60
11.00	0.258	0.153	89.35	0.283	0.117	86.05
12.00	0.319	-0.181	-88.88	0.271	0.099	82.71
13.00	0.342	-0.194	-89.18	0.243	0.081	81.95
14.00	0.325	0.183	87.21	0.210	0.083	87.23
15.00	0.269	0.166	86.22	0.176	0.087	89.53
16.00	0.183	0.147	82.10	0.141	-0.085	-88.19
17.00	0.078	0.131	79.85	0.104	-0.079	-88.97
18.00	-0.025	0.114	77.36	0.051	0.091	86.79

Γ.3-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ

implicit_pv.py

```
#####  
#1.)COMMENTS  
#  
#Module that calculates the temperature distribution of a pv over time.  
#Present code uses 2 external txt files as input:  
#insolation.txt ->GT [kW/m2]  
#tairarray.txt ->Tair [oC]  
#  
# L2,dx2  
# -----  
# | 1 | 2 |  
# | glass |PET|  
# -----  
# L1,dx1  
#  
#System of equations  
#  
# | d0 ud0 | | T0,p+1 | | b0 |  
# | ld0 d1 ud1 | | T1,p+1 | | b1 |  
# | ld1 d2 ud2 | | T2,p+1 | | b2 |  
# | . . . | | . | | . |  
# | . . . | X | . = | . |  
# | . . . | | . | | . |  
# | ldn-3 dn-2 udn-2 | | Tn-3,p+1 | | bn-2 |  
# | ldn-3 dn-2 udn-2 | | Tn-2,p+1 | | bn-2 |  
# | ldn-2 dn-1 | | Tn-1,p+1 | | bn-1 |  
#  
#Spacial index:0,1,2,...,i,...,n-1,n-2  
#Temporal index:0,1,2,...,p,...,m-2,m-1  
#####  
#2.)MODULE IMPORTATION  
  
import numpy as np  
import matplotlib.pyplot as plt  
from mpl_toolkits.mplot3d import Axes3D  
from thomas_algorithm import thomas  
import sys  
import math  
#####  
#3.)DATA INPUT  
  
#PROPERTIES  
#1:tempered (toughened) soda-lime glass  
L1=0.0032 #length, [m]  
R1=2440.0 #density, [kg/m3]  
K1=1.05 #thermal conductivity, [W/mk]
```

```

Cp1=0.72 #specific heat capacity, [kJ/kgK]

#2:PET panel (polyethylene terephthalate)
L2=0.001 #length, [m]
R2=1470.0 #density, [kg/m3]
K2=0.195 #thermal conductivity, [W/mk]
Cp2=1.075 #specific heat capacity, [kJ/kgK]

H1=4.0 #combined convection-radiation coefficient, upper surface, [W/m2K]
H2=4.0 #combined convection-radiation coefficient, lowesurface, [W/m2K]

aglass=0.96 #glass absorptivity, [-]

NOCT=47.0 #Nominal operating temperature, [oC]
#-----#
#Importing Tair array
Tairarray=np.loadtxt('tairarray.txt',dtype=float).reshape(-1,1)
Tairarraysize=Tairarray.size

#Importing GT array
GTarray=np.loadtxt('insolation.txt',dtype=float).reshape(-1,1)
GTarraysize=GTarray.size

#Verifying that Tair and GT arrays have the same size
if Tairarraysize!=GTarraysize:
    print("Tair, GT arrays are not of equal size!")
    sys.exit()
#-----#
#Temporal array, tarray
m=Tairarraysize #number of temporal nodes
dt=3600.0 # temporal step, [s]

tarray=np.linspace(0,m-1,num=m,endpoint=True).reshape(-1,1)
#-----#
#Spatial array, xarray
n1=15 #number of nodes, material 1 (glass)
n2=5 #number of nodes, material 2 (PET)
n=n1+n2 #total number of nodes
L=L1+L2 #total length, [m]
dx1=(2*L1)/((2*n1)-1) #spatial step, material 1 (glass), [m]
dx2=(2*L2)/((2*n2)-1) #spatial step, material 2 (PET), [m]
i12=n1 #layer change 1-2 (glass-PET)

dx=np.zeros((n-1,1),dtype=float) #spatial step, [m]
for i in range(0,n-1):
    if i<(i12-1): #material 1 (glass)
        dx[i,0]=dx1
    if i==(i12-1): #layer change 1-2 (glass-PET)
        dx[i,0]=(dx1/2)+(dx2/2)
    if i>(i12-1): #material 2 (PET)

```

```

dx[i,0]=dx2

xarray=np.zeros((n,1),dtype=float)
for i in range(1,n):
    xarray[i,0]=xarray[i-1,0]+dx[i-1,0]
#*****#
#MAIN CODE

#Create 'blank' arrays
K=np.zeros((n,1),dtype=float) #thermal conductivity per node, [W/mK]
R=np.zeros((n,1),dtype=float) #density per node, [kg/m3]
Cp=np.zeros((n,1),dtype=float) #specific heat capacity per node, [kJ/kgK]

ld=np.zeros((n-1,1),dtype=float) #lower-diagonal elements, ld
d=np.zeros((n,1),dtype=float) #diagonal elements, d
ud=np.zeros((n-1,1),dtype=float) #upper-diagonal elements, ud
b=np.zeros((n,1),dtype=float) #right side elements, b

Tarray=np.zeros((n,1),dtype=float) #Temperature distribution, [oC]

Tcellarray=np.zeros((m,1),dtype=float) #Cell temperature over time, [oC]

Tcellapproxarray=np.zeros((m,1),dtype=float) #Cell temperature approxmtn, [oC]

Tsurfacearray=np.zeros((m,n),dtype=float) #Surface temperature, [oC]
#-----#
#Initial conditions
for i in range(0,n):
    Tarray[i,0]=Tairarray[0,0] #initial temperature for all nodes
#-----#
#Finding out initial temperature distribution
ci=10 #estimated number of iterations for convergence

for p in range(0,ci):

    #K,R,Cp
    for i in range(0,n):
        if i<i12: #material 1 (glass)
            K[i,0]=K1
            R[i,0]=R1
            Cp[i,0]=Cp1
        if i>=i12 : #material 2 (PET)
            K[i,0]=K2
            R[i,0]=R2
            Cp[i,0]=Cp2

    #ld
    for i in range(1,n-1):
        temp1=(K[i,0]*dt)
        temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))

```

```

temp3=(temp1/temp2)*dx[i,0]
ld[i-1,0]=-temp3
ld[n-2,0]=-(K2/dx2) #BC at x=L

```

```
#d
```

```

for i in range(1,n-1):
    temp1=(K[i,0]*dt)
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
    temp3=1+((temp1/temp2)*(dx[i-1,0]+dx[i,0]))
    d[i,0]=temp3
d[0,0]=(K1/dx1)+H1 #BC at x=0
d[n-1,0]=(K2/dx2)+H2 #BC at x=L

```

```
#ud
```

```

for i in range(1,n-1):
    temp1=(K[i,0]*dt)
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
    temp3=(temp1/temp2)*dx[i-1,0]
    ud[i,0]=-temp3
ud[0,0]=-(K1/dx1) #BC at x=0

```

```
#b
```

```

for i in range(1,n-1):
    b[i,0]=(Tarray[i,0]+273.15)
b[0,0]=((Tairarray[0,0]+273.15)*H1)+(aglass*GTarray[0,0]*1000.0) #BC at x=0
b[n-1,0]=((Tairarray[0,0]+273.15)*H2) #BC at x=L

```

```
#Solving system using Thomas algorithm (TDMA)
```

```
Tarray=thomas(ld,d,ud,b,n)
```

```
#Reverting temperature back to Celsius
```

```

for i in range(0,n):
    Tarray[i,0]=(Tarray[i,0]-273.15)

```

```
#-----#
```

```
#Temporal loop
```

```
for p in range(0,m):
```

```
#K,R,Cp
```

```

for i in range(0,n):
    if i<i12: #material 1 (glass)
        K[i,0]=K1
        R[i,0]=R1
        Cp[i,0]=Cp1
    if i>=i12 : #material 2 (PET)
        K[i,0]=K2
        R[i,0]=R2
        Cp[i,0]=Cp2

```

```
#ld
```

```
for i in range(1,n-1):
```

```

temp1=(K[i,0]*dt)
temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
temp3=(temp1/temp2)*dx[i,0]
ld[i-1,0]=-temp3
ld[n-2,0]=-(K2/dx2) #BC at x=L

#d
for i in range(1,n-1):
    temp1=(K[i,0]*dt)
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
    temp3=1+((temp1/temp2)*(dx[i-1,0]+dx[i,0]))
    d[i,0]=temp3
d[0,0]=(K1/dx1)+H1 #BC at x=0
d[n-1,0]=(K2/dx2)+H2 #BC at x=L

#ud
for i in range(1,n-1):
    temp1=(K[i,0]*dt)
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
    temp3=(temp1/temp2)*dx[i-1,0]
    ud[i,0]=-temp3
ud[0,0]=-(K1/dx1) #BC at x=0

#b
for i in range(1,n-1):
    b[i,0]=(Tarray[i,0]+273.15)
b[0,0]=((Tairarray[p,0]+273.15)*H1)+(aglass*GTarray[p,0]*1000.0) #BC at x=0
b[n-1,0]=((Tairarray[p,0]+273.15)*H2) #BC at x=L

#Solving system using Thomas algorithm (TDMA)
Tarray=thomas(ld,d,ud,b,n)

#Reverting temperature back to Celsius
for i in range(0,n):
    Tarray[i,0]=(Tarray[i,0]-273.15)

#Cell temperature distribution over time
Tcellarray[p,0]=Tarray[i12,0]

#Cell temperature approximation
Tcellapproxarray[p,0]=Tairarray[p,0]+\
    ((NOCT-20.0)*((GTarray[p,0]*1000.0)/800.0))

for i in range(0,n): #surface plot, temperature distribution over x, over t
    Tsurfacearray[p,i]=Tarray[i,0]

'''
#Plotting temperature distribution at each time step
plt.clf() #clears' previous figures
plt.figure(1)

```

```

plt.title('time:=%i,Tair:=%0.2f,GT:=%0.2f' %(p,Tairarray[p,0],GTarray[p,0]))
plt.xlabel('x [m]')
plt.ylabel('T [oC]')
plt.axis([0,L,0,100.0])
plt.plot(xarray,Tarray,color='red')
plt.axvline(x=L1,color='blue',ls='dotted')
plt.grid(True)
plt.savefig('%0.3i.png' %(p))
'''

'''
print('*'*80)
print('Time:= %i [hours]' %(p))
print('  Tair:= %0.2f [oC]' %(Tairarray[p,0]))
print('  GT:= %0.2f [kW/m^2]' %(GTarray[p,0]))
print('*'*80)
print('Tarray:')
print(Tarray,Tarray.shape)
print('Tcell:= %0.2f' %(Tcellarray[p,0]))
print('*'*80)
'''

np.savetxt('tcellarray_pv.txt',Tcellarray) #save Tcell array into txt file
#-----#
'''
plt.figure(2)
plt.title('T-t')
plt.xlabel('t [hours]')
plt.ylabel('T [oC]')
plt.axis([0.0,m-1,0.0,100.0])
plt.plot(tarray,Tcellarray,color='red',label='Tcell')
plt.plot(tarray,Tairarray,color='black',ls='dotted',label='Tair')
#plt.plot(tarray,Tcellapproxarray,color='magenta',ls='dashdot')
plt.legend(loc='upper right')
plt.grid(True)
plt.savefig('tcell_pv.png')
plt.show()
'''

'''
#Surface plot, T-x over t
fig=plt.figure(3)
ax=fig.add_subplot(111,projection='3d')
sx,st=np.meshgrid(xarray,tarray)
ax.plot_surface(sx,st,Tsurfacearray,rstride=1,cstride=1,cmap='jet',linewidth=0)
ax.set_ylim(0.0,L)
ax.set_ylim(0.0,m-1)
ax.set_zlim(0.0,100.0)
ax.set_xlabel('x')
ax.set_ylabel('t')

```

```

ax.set_zlabel('T')
plt.savefig('surface_pv.png')
plt.show()
#"""
#-----#
"""
#System matrices
systemarray=np.zeros((n,4),dtype=float)
for i in range(0,n-1):
    systemarray[i+1,0]=ld[i,0]
    systemarray[i,2]=ud[i,0]
for i in range(0,n):
    systemarray[i,1]=d[i,0]
    systemarray[i,3]=b[i,0]
for i in range(0,n):
    temp0=systemarray[i,0]
    temp1=systemarray[i,1]
    temp2=systemarray[i,2]
    temp3=systemarray[i,3]
    print("%10.2f %10.2f %10.2f %10.2f" %(temp0,temp1,temp2,temp3))
"""
#*****#

```


Γ.4-ΥΠΟΛΟΓΙΣΜΟΣ ΣΥΝΤΕΛΕΣΤΗ ΜΕΤΑΦΟΡΑΣ ΘΕΡΜΟΤΗΤΑΣ

heat_transfer_coefficient.py

```
#####  
#1.)COMMENTS  
#Module that compares cell temperature arrays for 10 different heat transfer  
#coefficients (h1,h2), for the warmest period of time (1-3/7)  
#####  
#2.)MODULE IMPORTATION  
  
import numpy as np  
import matplotlib.pyplot as plt  
#####  
#3.)ARRAY IMPORTATION  
#Importing Tcell arrays  
Tairarray=np.loadtxt('tairarray.txt',dtype=float).reshape(-1,1)  
  
Tcellarray1=np.loadtxt('tcellarray_pv_1.txt',dtype=float).reshape(-1,1)  
Tcellarray2=np.loadtxt('tcellarray_pv_2.txt',dtype=float).reshape(-1,1)  
Tcellarray3=np.loadtxt('tcellarray_pv_3.txt',dtype=float).reshape(-1,1)  
Tcellarray4=np.loadtxt('tcellarray_pv_4.txt',dtype=float).reshape(-1,1)  
Tcellarray5=np.loadtxt('tcellarray_pv_5.txt',dtype=float).reshape(-1,1)  
Tcellarray6=np.loadtxt('tcellarray_pv_6.txt',dtype=float).reshape(-1,1)  
Tcellarray7=np.loadtxt('tcellarray_pv_7.txt',dtype=float).reshape(-1,1)  
Tcellarray8=np.loadtxt('tcellarray_pv_8.txt',dtype=float).reshape(-1,1)  
Tcellarray9=np.loadtxt('tcellarray_pv_9.txt',dtype=float).reshape(-1,1)  
Tcellarray10=np.loadtxt('tcellarray_pv_10.txt',dtype=float).reshape(-1,1)  
#####  
#4.)MAIN CODE  
  
#Temporal array  
m=Tcellarray1.size  
tarray=np.linspace(0,m-1,num=m,endpoint=True).reshape(-1,1)  
  
#air temperature difference of 30 oC  
Tairarrayplus30=np.zeros((m,1),dtype=float)  
print(Tairarrayplus30)  
  
for p in range(0,m):  
    Tairarrayplus30[p,0]=Tairarray[p,0]+30.0  
  
#Comparative plot  
plt.figure(2)  
plt.title("Tcell-t")  
plt.xlabel('t [hours]')  
plt.ylabel("Tcell [oC]")  
plt.axis([0.0,m-1,0.0,100.0])  
plt.plot(tarray,Tairarray,color='black',ls='dotted',label='Tair')  
plt.plot(tarray,Tairarrayplus30,color='black',ls='dashed',label='Tair+30oC')
```

```
plt.plot(tarray,Tcellarray1,color='red',label='hcr=1 [W/m2K]')
plt.plot(tarray,Tcellarray2,color='orange',label='hcr=2 [W/m2K]')
plt.plot(tarray,Tcellarray3,color='yellow',label='hcr=3 [W/m2K]')
plt.plot(tarray,Tcellarray4,color='green',label='hcr=4 [W/m2K]')
plt.plot(tarray,Tcellarray5,color='blue',label='hcr=5 [W/m2K]')
plt.plot(tarray,Tcellarray6,color='purple',label='hcr=6 [W/m2K]')
plt.plot(tarray,Tcellarray7,color='magenta',label='hcr=7 [W/m2K]')
plt.plot(tarray,Tcellarray8,color='pink',label='hcr=8 [W/m2K]')
plt.plot(tarray,Tcellarray9,color='brown',label='hcr=9 [W/m2K]')
plt.plot(tarray,Tcellarray10,color='gray',label='hcr=10 [W/m2K]')
plt.legend(loc='upper right')
plt.grid(True)
plt.savefig('heat_transfer_coefficient.png')
plt.show()
#*****#
```

**Γ.5-ΣΥΓΚΡΙΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΕΝΕΡΓΟΥ ΘΕΡΜΟΧΩΡΗΤΙΚΟΤΗΤΑΣ ΓΙΑ ΤΑ ΥΛΙΚΑ
ΑΛΛΑΓΗΣ ΦΑΣΗΣ RT20, RT27, SP25A8**

effective_heat_capacity_comparison_plot.py

```
#####  
#1.)COMMENTS  
#Plotting comparison plot of Cpeff=Cpeff(T) for 3 different PCMs.  
#####  
#2.)MODULE IMPORTATION  
import numpy as np  
import matplotlib.pyplot as plt  
import effective_heat_capacity_rt20 as rt20  
import effective_heat_capacity_rt27 as rt27  
import effective_heat_capacity_sp25a8 as sp25a8  
#####  
#3.)MAIN CODE  
  
Tstart=0.0 #starting temperature value  
Tstop=50.0 #ending temperature value  
n=101 #number of temperature values  
  
T=np.linspace(Tstart,Tstop,num=101,endpoint=True).reshape(-1,1)  
  
cpeff_rt20=np.zeros((n,1),dtype=float)  
cpeff_rt27=np.zeros((n,1),dtype=float)  
cpeff_sp25a8=np.zeros((n,1),dtype=float)  
  
for i in range(0,n):  
    Tc=T[i,0]  
    cpeff_rt20[i,0]=rt20.cpeff(Tc)  
    cpeff_rt27[i,0]=rt27.cpeff(Tc)  
    cpeff_sp25a8[i,0]=sp25a8.cpeff(Tc)  
  
#####  
#4.)PLOT  
plt.figure(1)  
plt.axis([0.0,50.0,0.0,100.0])  
plt.xlabel("T [oC]")  
plt.ylabel("Cpeff [kJ/kgC]")  
plt.title("Cpeff=Cpeff(T)")  
plt.plot(T,cpeff_rt20,color='red',label='RT20')  
plt.plot(T,cpeff_rt27,color='green',label='RT27')  
plt.plot(T,cpeff_sp25a8,color='blue',label='SP25A8')  
plt.legend(loc='upper right')  
plt.axvline(x=15,color='red',ls='dashed')  
plt.axvline(x=26,color='red',ls='dashed')  
plt.axvline(x=21.9,color='green',ls='dashed')  
plt.axvline(x=31,color='green',ls='dashed')  
plt.axvline(x=22.1,color='blue',ls='dashed')
```

```
plt.axvline(x=32,color='blue',ls='dashed')
plt.grid(True)
plt.savefig('cpeff_t_rt20_rt27_sp25a8')
plt.show()
#*****#
#"
print("cpeff_rt20:",cpeff_rt20,cpeff_rt20.shape)
print("cpeff_rt27:",cpeff_rt27,cpeff_rt27.shape)
print("cpeff_sp25a8:",cpeff_sp25a8,cpeff_sp25a8.shape)
print("T:",T,T.shape)
#"

```

Γ.6-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-ΥΛΙΚΟΥ ΑΛΛΑΓΗΣ ΦΑΣΗΣ

Γ.6.1-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-RT20

implicit_pv_rt20.py

```
#####  
#1.)COMMENTS  
#  
#Module that calculates the temperature distribution of a pv over time.  
#Present code uses 2 external txt files as input:  
#insolation.txt ->GT [kW/m2]  
#tairarray.txt ->Tair [oC]  
#  
# L2,dx2 L4,dx4  
# -----  
# | 1 | 2 | 3 | 4 | 5 |  
# | glass | PET | bag | PCM | bag |  
# -----  
# L1,dx1 L3,dx3 L5,dx5  
#  
#System of equations  
#  
# | d0 ud0 | | T0,p+1 | | b0 |  
# | ld0 d1 ud1 | | T1,p+1 | | b1 |  
# | ld1 d2 ud2 | | T2,p+1 | | b2 |  
# | . . . | | . | | . |  
# | . . . | X | . | = | . |  
# | . . . | | . | | . |  
# | ldn-3 dn-2 udn-2 | | Tn-3,p+1 | | bn-2 |  
# | ldn-3 dn-2 udn-2 | | Tn-2,p+1 | | bn-2 |  
# | ldn-2 dn-1 | | Tn-1,p+1 | | bn-1 |  
#  
#Spacial index:0,1,2,...,i,...,n-1,n-2  
#Temporal index:0,1,2,...,p,...,m-2,m-1  
#####  
#2.)MODULE IMPORTATION  
  
import numpy as np  
import matplotlib.pyplot as plt  
from mpl_toolkits.mplot3d import Axes3D  
from thomas_algorithm import thomas  
import effective_heat_capacity_rt20 as rt20  
import sys  
import math  
#####  
#3.)DATA INPUT
```

#PROPERTIES

#1:tempered (toughened) soda-lime glass

L1=0.0032 #length, [m]

R1=2440.0 #density, [kg/m3]

K1=1.05 #thermal conductivity, [W/mk]

Cp1=0.72 #specific heat capacity, [kJ/kgK]

#2:PET panel (polyethylene terephthalate)

L2=0.001 #length, [m]

R2=1470.0 #density, [kg/m3]

K2=0.195 #thermal conductivity, [W/mk]

Cp2=1.075 #specific heat capacity, [kJ/kgK]

#3:plastic bag (LDPE-low density polyethylene)

L3=0.001 #length [m]

R3=920.0 #density [kg/m3]

K3=0.36 #thermal conductivity [W/mk]

Cp3=2.3 #specific heat capacity [kJ/kgK]

#4:phase change material (paraffin RT20)

L4=0.05 #length [m]

R4=880.0 #density [kg/m3]

K4=0.2 #thermal conductivity [W/mk]

Cps=1.9 #specific heat capacity,solid state [kj/kgK or kj/kgC]

Cpl=2.5 #specific heat capacity,liquid state [kj/kgK or kj/kgC]

Ts=15.0 #T_solid [oC]

Tl=26.0 #T_liquid [oC]

#3:plastic bag (LDPE-low density polyethylene)

L5=0.001 #length [m]

R5=920.0 #density [kg/m3]

K5=0.36 #thermal conductivity [W/mk]

Cp5=2.3 #specific heat capacity [kJ/kgK]

H1=4.0 #combined convection-radiation coefficient, upper surface, [W/m2K]

H5=4.0 #combined convection-radiation coefficient, lowesurface, [W/m2K]

aglass=0.96 #glass absorptivity, [-]

NOCT=47.0 #Nominal operating temperature, [oC]

#-----#

#Importing Tair array

Tairarray=np.loadtxt('tairarray.txt',dtype=float).reshape(-1,1)

Tairarraysize=Tairarray.size

#Importing GT array

GTarray=np.loadtxt('insolation.txt',dtype=float).reshape(-1,1)

GTarraysize=GTarray.size

#Verifying that Tair and GT arrays have the same size

```

if Tairarraysize!=GTarraysize:
    print("Tair, GT arrays are not of equal size!")
    sys.exit()

#Importing array
Tcellarray_pvn=np.loadtxt('tcellarray_pvn.txt',dtype=float).reshape(-1,1)
#-----#
#Temporal array, tarray
m=Tairarraysize #number of temporal nodes
dt=3600.0 # temporal step, [s]

tarray=np.linspace(0,m-1,num=m,endpoint=True).reshape(-1,1)
#-----#
#Spatial array, xarray
n1=15 #number of nodes, material 1 (glass)
n2=5 #number of nodes, material 2 (PET)
n3=5 #number of nodes, material 3 (plastic bag)
n4=250 #number of nodes, material 4 (PCM)
n5=5 #number of nodes, material 3 (plastic bag)
n=n1+n2+n3+n4+n5 #total number of nodes
L=L1+L2+L3+L4+L5 #total length, [m]
dx1=(2*L1)/((2*n1)-1) #spatial step, material 1 (glass), [m]
dx2=(L2/n2) #spatial step, material 1 (glass), [m]
dx3=(L3/n3) #spatial step, material 1 (glass), [m]
dx4=(L4/n4) #spatial step, material 1 (glass), [m]
dx5=(2*L5)/((2*n5)-1) #spatial step, material 1 (glass), [m]
i12=n1 #layer change 1-2 (glass-PET)
i23=n1+n2 #layer change 2-3 (PET-plastic bag)
i34=n1+n2+n3 #layer change 3-4 (plastic bag-PCM)
i45=n1+n2+n3+n4 #layer change 4-5 (PCM-plastic bag)

dx=np.zeros((n-1,1),dtype=float) #spatial step, [m]
for i in range(0,n-1):
    if i<(i12-1): #material 1 (glass)
        dx[i,0]=dx1
    if i==(i12-1): #layer change 1-2 (glass-PET)
        dx[i,0]=(dx1/2)+(dx2/2)
    if i>(i12-1) and i<(i23-1): #material 2 (PET)
        dx[i,0]=dx2
    if i==(i23-1): #layer change 2-3 (PET-plastic bag)
        dx[i,0]=(dx2/2)+(dx3/2)
    if i>(i23-1) and i<(i34-1): #material 3 (plastic bag)
        dx[i,0]=dx3
    if i==(i34-1): #layer change 3-4 (plastic bag-PCM)
        dx[i,0]=(dx3/2)+(dx4/2)
    if i>(i34-1) and i<(i45-1): #material 4 (PCM)
        dx[i,0]=dx4
    if i==(i45-1): #layer change 4-5 (PCM-plastic bag)
        dx[i,0]=(dx4/2)+(dx5/2)
    if i>(i45-1): #material 5 (plastic bag)

```

```

dx[i,0]=dx5

xarray=np.zeros((n,1),dtype=float)
for i in range(1,n):
    xarray[i,0]=xarray[i-1,0]+dx[i-1,0]
#*****#
#MAIN CODE

#Create 'blank' arrays
K=np.zeros((n,1),dtype=float) #thermal conductivity per node, [W/mK]
R=np.zeros((n,1),dtype=float) #density per node, [kg/m3]
Cp=np.zeros((n,1),dtype=float) #specific heat capacity per node, [kJ/kgK]

ld=np.zeros((n-1,1),dtype=float) #lower-diagonal elements, ld
d=np.zeros((n,1),dtype=float) #diagonal elements, d
ud=np.zeros((n-1,1),dtype=float) #upper-diagonal elements, ud
b=np.zeros((n,1),dtype=float) #right side elements, b

Tarray=np.zeros((n,1),dtype=float) #Temperature distribution, [oC]

Tcellarray=np.zeros((m,1),dtype=float) #Cell temperature over time, [oC]

Tcellapproxarray=np.zeros((m,1),dtype=float) #Cell temperature approx, [oC]

Tsurfacearray=np.zeros((m,n),dtype=float) #Surface temperature, [oC]

Tdiffarray=np.zeros((m,1),dtype=float) #Cell-air temperature difference, [oC]

Tpcmstart=np.zeros((m,1),dtype=float) #PCM temperature, first node [oC]
Tpcmmid=np.zeros((m,1),dtype=float) #PCM temperature, middle node [oC]
Tpcmend=np.zeros((m,1),dtype=float) #PCM temperature, first node [oC]
#-----#
#Initial conditions
for i in range(0,n):
    Tarray[i,0]=Tairarray[0,0] #initial temperature for all nodes
#-----#
#Finding out initial temperature distribution
ci=10 #estimated number of iterations for convergence

for p in range(0,ci):

    #K,R,Cp
    for i in range(0,n):
        if i<i12: #material 1 (glass)
            K[i,0]=K1
            R[i,0]=R1
            Cp[i,0]=Cp1
        if i>=i12 and i<i23: #material 2 (PET)
            K[i,0]=K2
            R[i,0]=R2

```



```

    Cp[i,0]=Cp2
    if i>=i23 and i<i34: #material 3 (plastic bag)
        K[i,0]=K3
        R[i,0]=R3
        Cp[i,0]=Cp3
    if i>=i34 and i<i45: #material 4 (PCM)
        Cp4=rt20.cpeff(Tarray[i,0])
        K[i,0]=K4
        R[i,0]=R4
        Cp[i,0]=Cp4
    if i>=i45: #material 5 (plastic bag)
        K[i,0]=K5
        R[i,0]=R5
        Cp[i,0]=Cp5

#ld
for i in range(1,n-1):
    temp1=(K[i,0]*dt)
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
    temp3=(temp1/temp2)*dx[i,0]
    ld[i-1,0]=-temp3
ld[n-2,0]=-(K5/dx5) #BC at x=L

#d
for i in range(1,n-1):
    temp1=(K[i,0]*dt)
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
    temp3=1+((temp1/temp2)*(dx[i-1,0]+dx[i,0]))
    d[i,0]=temp3
d[0,0]=(K1/dx1)+H1 #BC at x=0
d[n-1,0]=(K5/dx5)+H5 #BC at x=L

#ud
for i in range(1,n-1):
    temp1=(K[i,0]*dt)
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
    temp3=(temp1/temp2)*dx[i-1,0]
    ud[i,0]=-temp3
ud[0,0]=-(K1/dx1) #BC at x=0

#b
for i in range(1,n-1):
    b[i,0]=(Tarray[i,0]+273.15)
b[0,0]=((Tairarray[0,0]+273.15)*H1)+(aglass*GTarray[0,0]*1000.0) #BC at x=0
b[n-1,0]=((Tairarray[0,0]+273.15)*H5) #BC at x=L

#Solving system using Thomas algorithm (TDMA)
Tarray=thomas(ld,d,ud,b,n)

#Reverting temperature back to Celsius

```

```

for i in range(0,n):
    Tarray[i,0]=(Tarray[i,0]-273.15)
#-----#
#Temporal loop
for p in range(0,m):

    #K,R,Cp
    for i in range(0,n):
        if i<i12: #material 1 (glass)
            K[i,0]=K1
            R[i,0]=R1
            Cp[i,0]=Cp1
        if i>=i12 and i<i23: #material 2 (PET)
            K[i,0]=K2
            R[i,0]=R2
            Cp[i,0]=Cp2
        if i>=i23 and i<i34: #material 3 (plastic bag)
            K[i,0]=K3
            R[i,0]=R3
            Cp[i,0]=Cp3
        if i>=i34 and i<i45: #material 4 (PCM)
            Cp4=rt20.cpeff(Tarray[i,0])
            K[i,0]=K4
            R[i,0]=R4
            Cp[i,0]=Cp4
        if i>=i45: #material 5 (plastic bag)
            K[i,0]=K5
            R[i,0]=R5
            Cp[i,0]=Cp5

    #ld
    for i in range(1,n-1):
        temp1=(K[i,0]*dt)
        temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
        temp3=(temp1/temp2)*dx[i,0]
        ld[i-1,0]=-temp3
    ld[n-2,0]=-(K5/dx5) #BC at x=L

    #d
    for i in range(1,n-1):
        temp1=(K[i,0]*dt)
        temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
        temp3=1+((temp1/temp2)*(dx[i-1,0]+dx[i,0]))
        d[i,0]=temp3
    d[0,0]=(K1/dx1)+H1 #BC at x=0
    d[n-1,0]=(K5/dx5)+H5 #BC at x=L

    #ud
    for i in range(1,n-1):
        temp1=(K[i,0]*dt)

```

```

temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
temp3=(temp1/temp2)*dx[i-1,0]
ud[i,0]=-temp3
ud[0,0]=-(K1/dx1) #BC at x=0

#b
for i in range(1,n-1):
    b[i,0]=(Tarray[i,0]+273.15)
b[0,0]=((Tairarray[p,0]+273.15)*H1)+(aglass*GTarray[p,0]*1000.0) #BC at x=0
b[n-1,0]=((Tairarray[p,0]+273.15)*H5) #BC at x=L

#Solving system using Thomas algorithm (TDMA)
Tarray=thomas(ld,d,ud,b,n)

#Reverting temperature back to Celsius
for i in range(0,n):
    Tarray[i,0]=(Tarray[i,0]-273.15)

#Cell temperature distribution over time
Tcellarray[p,0]=Tarray[i12,0]

#Cell temperature approximaton
Tcellapproxarray[p,0]=Tairarray[p,0]+\
    ((NOCT-20.0)*((GTarray[p,0]*1000.0)/800.0))

for i in range(0,n): #surface plot, temperature distribution over x,over t
    Tsurfacearray[p,i]=Tarray[i,0]

#Cell-air temperature difference
Tdiffarray[p,0]=Tarray[i12,0]-Tairarray[p,0]

#PCM temperature
middle=math.floor((i34+i45)/2)
Tpcmstart[p,0]=Tarray[i34,0] #first PCM node
Tpcmmid[p,0]=Tarray[middle,0] #middle PCM node
Tpcmend[p,0]=Tarray[i45,0] #last PCM node

'''
#Plotting temperature distribution at each time step
plt.clf() #clears' previous figures
plt.figure(1)
plt.title('time:=%i,Tair:=%0.2f,GT:=%0.2f'%(p,Tairarray[p,0],GTarray[p,0]))
plt.xlabel('x [m]')
plt.ylabel('T [oC]')
plt.axis([0,L,0,100.0])
plt.plot(xarray,Tarray,color='red')
plt.axvline(x=L1,color='blue',ls='dotted')
plt.grid(True)
plt.savefig('%0.3i.png'%p)
'''

```

```

"""
print('*'*80)
print('Time:= %i [hours]' %(p))
print('  Tair:= %.2f [oC]' %(Tairarray[p,0]))
print('  GT:= %.2f [kW/m^2]' %(GTarray[p,0]))
print('-'*80)
print('Tarray:')
print(Tarray,Tarray.shape)
print('Tcell:= %.2f' %(Tcellarray[p,0]))
print('*'*80)
"""

np.savetxt('tcellarray_pv_rt20.txt',Tcellarray) #save Tcell array into txt file
#-----#
"""
plt.figure(2)
plt.title('T-t')
plt.xlabel('t [hours]')
plt.ylabel('T [oC]')
plt.axis([0.0,m-1,0.0,100.0])
plt.plot(tarray,Tcellarray_pv,color='red',ls='dashed',label='Tcell_pv')
plt.plot(tarray,Tcellarray,color='blue',label='Tcell_rt20')
plt.plot(tarray,Tairarray,color='black',ls='dotted',label='Tair')
#plt.plot(tarray,Tcellapproxarray,color='magenta',ls='dashdot',label='Tclaprx')
#plt.plot(tarray,Tdiffarray,color='green',ls='dashdot',label='Tdiff')
plt.legend(loc='upper right')
plt.grid(True)
plt.savefig('tcell_pv_rt20.png')
plt.show()
"""

"""
plt.figure(3)
plt.title('Trt20-t')
plt.xlabel('t [hours]')
plt.ylabel('T [oC]')
plt.axis([0.0,m-1,0.0,100.0])
plt.plot(tarray,Tpcmstart,color='red',label='first PCM node')
plt.plot(tarray,Tpcmmid,color='green',label='middle PCM node')
plt.plot(tarray,Tpcmend,color='blue',label='last PCM node')
plt.axhline(y=Ts,color='black',ls='dashed',label='Tsolid')
plt.axhline(y=Tl,color='black',ls='dashed',label='Tliquid')
plt.legend(loc='upper left')
plt.grid(True)
plt.savefig('tpcm_rt20.png')
plt.show()
"""

"""

```

```

#Surface plot, T-x over t
fig=plt.figure(4)
ax=fig.add_subplot(111,projection='3d')
sx,st=np.meshgrid(xarray,tarray)
ax.plot_surface(sx,st,Tsurfacearray,rstride=1,cstride=1,cmap='jet',linewidth=0)
ax.set_ylim(0.0,L)
ax.set_ylim(0.0,m-1)
ax.set_zlim(0.0,100.0)
ax.set_xlabel('x')
ax.set_ylabel('t')
ax.set_zlabel('T')
plt.savefig('surface_pv_rt20.png')
plt.show()
#"""
#-----#
"""
#System matrices
systemarray=np.zeros((n,4),dtype=float)
for i in range(0,n-1):
    systemarray[i+1,0]=ld[i,0]
    systemarray[i,2]=ud[i,0]
for i in range(0,n):
    systemarray[i,1]=d[i,0]
    systemarray[i,3]=b[i,0]
for i in range(0,n):
    temp0=systemarray[i,0]
    temp1=systemarray[i,1]
    temp2=systemarray[i,2]
    temp3=systemarray[i,3]
    print("%10.2f %10.2f %10.2f %10.2f"%(temp0,temp1,temp2,temp3))
"""
#*****#

```

**Γ.6.2-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ
ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-RT27**

implicit_pv_rt27.py

```
#####  
#1.)COMMENTS  
#  
#Module that calculates the temperature distribution of a pv over time.  
#Present code uses 2 external txt files as input:  
#insolation.txt ->GT [kW/m2]  
#tairarray.txt ->Tair [oC]  
#  
# L2,dx2 L4,dx4  
# -----  
# | 1 | 2 | 3 | 4 | 5 |  
# | glass | PET | bag | PCM | bag |  
# -----  
# L1,dx1 L3,dx3 L5,dx5  
#  
#System of equations  
#  
# | d0 ud0 | | T0,p+1 | | b0 |  
# | ld0 d1 ud1 | | T1,p+1 | | b1 |  
# | ld1 d2 ud2 | | T2,p+1 | | b2 |  
# | . . . | | . | | . |  
# | . . . | X | . | = | . |  
# | . . . | | . | | . |  
# | ldn-3 dn-2 udn-2 | | Tn-3,p+1 | | bn-2 |  
# | ldn-3 dn-2 udn-2 | | Tn-2,p+1 | | bn-2 |  
# | ldn-2 dn-1 | | Tn-1,p+1 | | bn-1 |  
#  
#Spacial index:0,1,2,...,i,...,n-1,n-2  
#Temporal index:0,1,2,...,p,...,m-2,m-1  
#####  
#2.)MODULE IMPORTATION  
  
import numpy as np  
import matplotlib.pyplot as plt  
from mpl_toolkits.mplot3d import Axes3D  
from thomas_algorithm import thomas  
import effective_heat_capacity_rt27 as rt27  
import sys  
import math  
#####  
#3.)DATA INPUT  
  
#PROPERTIES  
#1:tempered (toughened) soda-lime glass  
L1=0.0032 #length, [m]
```

```

R1=2440.0 #density, [kg/m3]
K1=1.05 #thermal conductivity, [W/mk]
Cp1=0.72 #specific heat capacity, [kJ/kgK]

#2:PET panel (polyethylene terephthalate)
L2=0.001 #length, [m]
R2=1470.0 #density, [kg/m3]
K2=0.195 #thermal conductivity, [W/mk]
Cp2=1.075 #specific heat capacity, [kJ/kgK]

#3:plastic bag (LDPE-low density polyethylene)
L3=0.001 #length [m]
R3=920.0 #density [kg/m3]
K3=0.36 #thermal conductivity [W/mk]
Cp3=2.3 #specific heat capacity [kJ/kgK]

#4:phase change material (paraffin RT27)
L4=0.05 #length [m]
R4=880.0 #density [kg/m3]
K4=0.2 #thermal conductivity [W/mk]
Cps=2.0 #specific heat capacity,solid state [kj/kgK or kj/kgC]
Cpl=2.3 #specific heat capacity,liquid state [kj/kgK or kj/kgC]
Ts=22.0 #T_solid [oC]
Tl=31.0 #T_liquid [oC]

#5:plastic bag (LDPE-low density polyethylene)
L5=0.001 #length [m]
R5=920.0 #density [kg/m3]
K5=0.36 #thermal conductivity [W/mk]
Cp5=2.3 #specific heat capacity [kJ/kgK]

H1=4.0 #combined convection-radiation coefficient, upper surface, [W/m2K]
H5=4.0 #combined convection-radiation coefficient, lowesurface, [W/m2K]

aglass=0.96 #glass absorptivity, [-]

NOCT=47.0 #Nominal operating temperature, [oC]
#-----#
#Importing Tair array
Tairarray=np.loadtxt('tairarray.txt',dtype=float).reshape(-1,1)
Tairarraysize=Tairarray.size

#Importing GT array
GTarray=np.loadtxt('insolation.txt',dtype=float).reshape(-1,1)
GTarraysize=GTarray.size

#Verifying that Tair and GT arrays have the same size
if Tairarraysize!=GTarraysize:
    print('Tair, GT arrays are not of equal size!')
    sys.exit()

```

```

#Importing array
Tcellarray_pv=np.loadtxt('tcellarray_pv.txt',dtype=float).reshape(-1,1)
#-----#
#Temporal array, tarray
m=Tairarraysize #number of temporal nodes
dt=3600.0 # temporal step, [s]

tarray=np.linspace(0,m-1,num=m,endpoint=True).reshape(-1,1)
#-----#
#Spatial array, xarray
n1=15 #number of nodes, material 1 (glass)
n2=5 #number of nodes, material 2 (PET)
n3=5 #number of nodes, material 3 (plastic bag)
n4=250 #number of nodes, material 4 (PCM)
n5=5 #number of nodes, material 5 (plastic bag)
n=n1+n2+n3+n4+n5 #total number of nodes
L=L1+L2+L3+L4+L5 #total length, [m]
dx1=(2*L1)/((2*n1)-1) #spatial step, material 1 (glass), [m]
dx2=(L2/n2) #spatial step, material 1 (glass), [m]
dx3=(L3/n3) #spatial step, material 1 (glass), [m]
dx4=(L4/n4) #spatial step, material 1 (glass), [m]
dx5=(2*L5)/((2*n5)-1) #spatial step, material 1 (glass), [m]
i12=n1 #layer change 1-2 (glass-PET)
i23=n1+n2 #layer change 2-3 (PET-plastic bag)
i34=n1+n2+n3 #layer change 3-4 (plastic bag-PCM)
i45=n1+n2+n3+n4 #layer change 4-5 (PCM-plastic bag)

dx=np.zeros((n-1,1),dtype=float) #spatial step, [m]
for i in range(0,n-1):
    if i<(i12-1): #material 1 (glass)
        dx[i,0]=dx1
    if i==(i12-1): #layer change 1-2 (glass-PET)
        dx[i,0]=(dx1/2)+(dx2/2)
    if i>(i12-1) and i<(i23-1): #material 2 (PET)
        dx[i,0]=dx2
    if i==(i23-1): #layer change 2-3 (PET-plastic bag)
        dx[i,0]=(dx2/2)+(dx3/2)
    if i>(i23-1) and i<(i34-1): #material 3 (plastic bag)
        dx[i,0]=dx3
    if i==(i34-1): #layer change 3-4 (plastic bag-PCM)
        dx[i,0]=(dx3/2)+(dx4/2)
    if i>(i34-1) and i<(i45-1): #material 4 (PCM)
        dx[i,0]=dx4
    if i==(i45-1): #layer change 4-5 (PCM-plastic bag)
        dx[i,0]=(dx4/2)+(dx5/2)
    if i>(i45-1): #material 5 (plastic bag)
        dx[i,0]=dx5

xarray=np.zeros((n,1),dtype=float)

```



```

for i in range(1,n):
    xarray[i,0]=xarray[i-1,0]+dx[i-1,0]
#*****#
#MAIN CODE

#Create 'blank' arrays
K=np.zeros((n,1),dtype=float) #thermal conductivity per node, [W/mK]
R=np.zeros((n,1),dtype=float) #density per node, [kg/m3]
Cp=np.zeros((n,1),dtype=float) #specific heat capacity per node, [kJ/kgK]

ld=np.zeros((n-1,1),dtype=float) #lower-diagonal elements, ld
d=np.zeros((n,1),dtype=float) #diagonal elements, d
ud=np.zeros((n-1,1),dtype=float) #upper-diagonal elements, ud
b=np.zeros((n,1),dtype=float) #right side elements, b

Tarray=np.zeros((n,1),dtype=float) #Temperature distribution, [oC]

Tcellarray=np.zeros((m,1),dtype=float) #Cell temperature over time, [oC]

Tcellapproxarray=np.zeros((m,1),dtype=float) #Cell temperature approx, [oC]

Tsurfacearray=np.zeros((m,n),dtype=float) #Surface temperature, [oC]

Tdiffarray=np.zeros((m,1),dtype=float) #Cell-air temperature difference, [oC]

Tpcmstart=np.zeros((m,1),dtype=float) #PCM temperature, first node [oC]
Tpcmmid=np.zeros((m,1),dtype=float) #PCM temperature, middle node [oC]
Tpcmend=np.zeros((m,1),dtype=float) #PCM temperature, first node [oC]
#-----#
#Initial conditions
for i in range(0,n):
    Tarray[i,0]=Tairarray[0,0] #initial temperature for all nodes
#-----#
#Finding out initial temperature distribution
ci=10 #estimated number of iterations for convergence

for p in range(0,ci):

    #K,R,Cp
    for i in range(0,n):
        if i<i12: #material 1 (glass)
            K[i,0]=K1
            R[i,0]=R1
            Cp[i,0]=Cp1
        if i>=i12 and i<i23: #material 2 (PET)
            K[i,0]=K2
            R[i,0]=R2
            Cp[i,0]=Cp2
        if i>=i23 and i<i34: #material 3 (plastic bag)
            K[i,0]=K3

```

```

R[i,0]=R3
Cp[i,0]=Cp3
if i>=i34 and i<i45: #material 4 (PCM)
    Cp4=rt20.cpeff(Tarray[i,0])
    K[i,0]=K4
    R[i,0]=R4
    Cp[i,0]=Cp4
if i>=i45: #material 5 (plastic bag)
    K[i,0]=K5
    R[i,0]=R5
    Cp[i,0]=Cp5

```

```
#ld
```

```

for i in range(1,n-1):
    temp1=(K[i,0]*dt)
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
    temp3=(temp1/temp2)*dx[i,0]
    ld[i-1,0]=-temp3
ld[n-2,0]=-(K5/dx5) #BC at x=L

```

```
#d
```

```

for i in range(1,n-1):
    temp1=(K[i,0]*dt)
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
    temp3=1+((temp1/temp2)*(dx[i-1,0]+dx[i,0]))
    d[i,0]=temp3
d[0,0]=(K1/dx1)+H1 #BC at x=0
d[n-1,0]=(K5/dx5)+H5 #BC at x=L

```

```
#ud
```

```

for i in range(1,n-1):
    temp1=(K[i,0]*dt)
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
    temp3=(temp1/temp2)*dx[i-1,0]
    ud[i,0]=-temp3
ud[0,0]=-(K1/dx1) #BC at x=0

```

```
#b
```

```

for i in range(1,n-1):
    b[i,0]=(Tarray[i,0]+273.15)
b[0,0]=((Tairarray[0,0]+273.15)*H1)+(aglass*GTarray[0,0]*1000.0) #BC at x=0
b[n-1,0]=((Tairarray[0,0]+273.15)*H5) #BC at x=L

```

```
#Solving system using Thomas algorithm (TDMA)
```

```
Tarray=thomas(ld,d,ud,b,n)
```

```
#Reverting temperature back to Celsius
```

```

for i in range(0,n):
    Tarray[i,0]=(Tarray[i,0]-273.15)

```

```
#-----#
```

```
#Temporal loop
for p in range(0,m):
```

```
#K,R,Cp
```

```
for i in range(0,n):
```

```
    if i<i12: #material 1 (glass)
```

```
        K[i,0]=K1
```

```
        R[i,0]=R1
```

```
        Cp[i,0]=Cp1
```

```
    if i>=i12 and i<i23: #material 2 (PET)
```

```
        K[i,0]=K2
```

```
        R[i,0]=R2
```

```
        Cp[i,0]=Cp2
```

```
    if i>=i23 and i<i34: #material 3 (plastic bag)
```

```
        K[i,0]=K3
```

```
        R[i,0]=R3
```

```
        Cp[i,0]=Cp3
```

```
    if i>=i34 and i<i45: #material 4 (PCM)
```

```
        Cp4=rt27.cpeff(Tarray[i,0])
```

```
        K[i,0]=K4
```

```
        R[i,0]=R4
```

```
        Cp[i,0]=Cp4
```

```
    if i>=i45: #material 5 (plastic bag)
```

```
        K[i,0]=K5
```

```
        R[i,0]=R5
```

```
        Cp[i,0]=Cp5
```

```
#ld
```

```
for i in range(1,n-1):
```

```
    temp1=(K[i,0]*dt)
```

```
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
```

```
    temp3=(temp1/temp2)*dx[i,0]
```

```
    ld[i-1,0]=-temp3
```

```
ld[n-2,0]=-(K5/dx5) #BC at x=L
```

```
#d
```

```
for i in range(1,n-1):
```

```
    temp1=(K[i,0]*dt)
```

```
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
```

```
    temp3=1+((temp1/temp2)*(dx[i-1,0]+dx[i,0]))
```

```
    d[i,0]=temp3
```

```
d[0,0]=(K1/dx1)+H1 #BC at x=0
```

```
d[n-1,0]=(K5/dx5)+H5 #BC at x=L
```

```
#ud
```

```
for i in range(1,n-1):
```

```
    temp1=(K[i,0]*dt)
```

```
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
```

```
    temp3=(temp1/temp2)*dx[i-1,0]
```

```
    ud[i,0]=-temp3
```

```

ud[0,0]=-(K1/dx1) #BC at x=0

#b
for i in range(1,n-1):
    b[i,0]=(Tarray[i,0]+273.15)
b[0,0]=((Tairarray[p,0]+273.15)*H1)+(aglass*GTarray[p,0]*1000.0) #BC at x=0
b[n-1,0]=((Tairarray[p,0]+273.15)*H5) #BC at x=L

#Solving system using Thomas algorithm (TDMA)
Tarray=thomas(ld,d,ud,b,n)

#Reverting temperature back to Celsius
for i in range(0,n):
    Tarray[i,0]=(Tarray[i,0]-273.15)

#Cell temperature distribution over time
Tcellarray[p,0]=Tarray[i12,0]

#Cell temperature approximaton
Tcellapproxarray[p,0]=Tairarray[p,0]+\
    ((NOCT-20.0)*((GTarray[p,0]*1000.0)/800.0))

for i in range(0,n): #surface plot, temperature distribution over x,over t
    Tsurfacearray[p,i]=Tarray[i,0]

#Cell-air temperature difference
Tdiffarray[p,0]=Tarray[i12,0]-Tairarray[p,0]

#PCM temperature
middle=math.floor((i34+i45)/2)
Tpcmstart[p,0]=Tarray[i34,0] #first PCM node
Tpcmmid[p,0]=Tarray[middle,0] #middle PCM node
Tpcmend[p,0]=Tarray[i45,0] #last PCM node

'''
#Plotting temperature distribution at each time step
plt.clf() #'clears' previous figures
plt.figure(1)
plt.title('time:=%i,Tair:=%.2f,GT:=%.2f'%(p,Tairarray[p,0],GTarray[p,0]))
plt.xlabel('x [m]')
plt.ylabel('T [oC]')
plt.axis([0,L,0,100.0])
plt.plot(xarray,Tarray,color='red')
plt.axvline(x=L1,color='blue',ls='dotted')
plt.grid(True)
plt.savefig('%i.png'%p)
'''

#'''
print('*'*80)

```

```

print('Time:= %i [hours]' %(p))
print('  Tair:= %.2f [oC]' %(Tairarray[p,0]))
print('  GT:= %.2f [kW/m^2]' %(GTarray[p,0]))
print('-'*80)
print('Tarray:')
print(Tarray,Tarray.shape)
print('Tcell:= %.2f' %(Tcellarray[p,0]))
print('*'*80)
#""

np.savetxt('tcellarray_pv_rt27.txt',Tcellarray) #save Tcell array into txt file
#-----#
#""
plt.figure(2)
plt.title('T-t')
plt.xlabel('t [hours]')
plt.ylabel('T [oC]')
plt.axis([0.0,m-1,0.0,100.0])
plt.plot(tarray,Tcellarray_pv,color='red',ls='dashed',label='Tcell_pv')
plt.plot(tarray,Tcellarray,color='blue',label='Tcell_rt27')
plt.plot(tarray,Tairarray,color='black',ls='dotted',label='Tair')
#plt.plot(tarray,Tcellapproxarray,color='magenta',ls='dashdot',label='Tclaprx')
#plt.plot(tarray,Tdiffarray,color='green',ls='dashdot',label='Tdiff')
plt.legend(loc='upper right')
plt.grid(True)
plt.savefig('tcell_pv_rt27.png')
plt.show()
#""

#""
plt.figure(3)
plt.title('Trt27-t')
plt.xlabel('t [hours]')
plt.ylabel('T [oC]')
plt.axis([0.0,m-1,0.0,100.0])
plt.plot(tarray,Tpcmstart,color='red',label='first PCM node')
plt.plot(tarray,Tpcmmid,color='green',label='middle PCM node')
plt.plot(tarray,Tpcmend,color='blue',label='last PCM node')
plt.axhline(y=Ts,color='black',ls='dashed',label='Tsolid')
plt.axhline(y=Tl,color='black',ls='dashed',label='Tliquid')
plt.legend(loc='upper left')
plt.grid(True)
plt.savefig('tpcm_rt27.png')
plt.show()
#""

#""
#Surface plot, T-x over t
fig=plt.figure(4)
ax=fig.add_subplot(111,projection='3d')

```

```

sx,st=np.meshgrid(xarray,tarray)
ax.plot_surface(sx,st,Tsurfacearray,rstride=1,cstride=1,cmap='jet',linewidth=0)
ax.set_ylim(0.0,L)
ax.set_ylim(0.0,m-1)
ax.set_zlim(0.0,100.0)
ax.set_xlabel('x')
ax.set_ylabel('t')
ax.set_zlabel('T')
plt.savefig('surface_pv_rt27.png')
plt.show()
"""
#-----#
"""
#System matrices
systemarray=np.zeros((n,4),dtype=float)
for i in range(0,n-1):
    systemarray[i+1,0]=ld[i,0]
    systemarray[i,2]=ud[i,0]
for i in range(0,n):
    systemarray[i,1]=d[i,0]
    systemarray[i,3]=b[i,0]
for i in range(0,n):
    temp0=systemarray[i,0]
    temp1=systemarray[i,1]
    temp2=systemarray[i,2]
    temp3=systemarray[i,3]
    print("%10.2f %10.2f %10.2f %10.2f" %(temp0,temp1,temp2,temp3))
"""
#*****#

```

**Γ.6.3-ΥΠΟΛΟΓΙΣΜΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΗΣ ΣΥΣΤΗΜΑΤΟΣ
ΦΩΤΟΒΟΛΤΑΪΚΗΣ ΜΟΝΑΔΑΣ-SP25A8**

implicit_pv_sp25a8.py

```

#####
#1.)COMMENTS
#
#Module that calculates the temperature distribution of a pv over time.
#Present code uses 2 external txt files as input:
#insolation.txt ->GT [kW/m2]
#tairarray.txt ->Tair [oC]
#
# L2,dx2 L4,dx4
# -----
# | 1 | 2 | 3 | 4 | 5 |
# | glass | PET | bag | PCM | bag |
# -----
# L1,dx1 L3,dx3 L5,dx5
#
#System of equations
#
#| d0 ud0 | | T0,p+1 | | b0|
#| ld0 d1 ud1 | | T1,p+1 | | b1|
#| ld1 d2 ud2 | | T2,p+1 | | b2|
#| . . . | | . | | . |
#| . . . | X | . | = | . |
#| . . . | | . | | . |
#| ldn-3 dn-2 udn-2 | | Tn-3,p+1 | | bn-2|
#| ldn-3 dn-2 udn-2 | | Tn-2,p+1 | | bn-2|
#| ldn-2 dn-1 | | Tn-1,p+1 | | bn-1|
#
#Spacial index:0,1,2,...,i,...,n-1,n-2
#Temporal index:0,1,2,...,p,...,m-2,m-1
#####
#2.)MODULE IMPORTATION

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from thomas_algorithm import thomas
import effective_heat_capacity_sp25a8 as sp25a8
import sys
import math
#####
#3.)DATA INPUT

#PROPERTIES
#1:tempered (toughened) soda-lime glass
L1=0.0032 #length, [m]

```

```

R1=2440.0 #density, [kg/m3]
K1=1.05 #thermal conductivity, [W/mk]
Cp1=0.72 #specific heat capacity, [kJ/kgK]

#2:PET panel (polyethylene terephthalate)
L2=0.001 #length, [m]
R2=1470.0 #density, [kg/m3]
K2=0.195 #thermal conductivity, [W/mk]
Cp2=1.075 #specific heat capacity, [kJ/kgK]

#3:plastic bag (LDPE-low density polyethylene)
L3=0.001 #length [m]
R3=920.0 #density [kg/m3]
K3=0.36 #thermal conductivity [W/mk]
Cp3=2.3 #specific heat capacity [kJ/kgK]

#4:phase change material (paraffin SP25A8)
L4=0.05 #length [m]
R4=1500.0 #density [kg/m3]
K4=0.6 #thermal conductivity [W/mk]
Cps=2.6 #specific heat capacity,solid state [kj/kgK or kj/kgC]
Cpl=2.7 #specific heat capacity,liquid state [kj/kgK or kj/kgC]
Ts=22.0 #T_solid [oC]
Tl=32.0 #T_liquid [oC]

#5:plastic bag (LDPE-low density polyethylene)
L5=0.001 #length [m]
R5=920.0 #density [kg/m3]
K5=0.36 #thermal conductivity [W/mk]
Cp5=2.3 #specific heat capacity [kJ/kgK]

H1=4.0 #combined convection-radiation coefficient, upper surface, [W/m2K]
H5=4.0 #combined convection-radiation coefficient, lowesurface, [W/m2K]

aglass=0.96 #glass absorptivity, [-]

NOCT=47.0 #Nominal operating temperature, [oC]
#-----#
#Importing Tair array
Tairarray=np.loadtxt('tairarray.txt',dtype=float).reshape(-1,1)
Tairarraysize=Tairarray.size

#Importing GT array
GTarray=np.loadtxt('insolation.txt',dtype=float).reshape(-1,1)
GTarraysize=GTarray.size

#Verifying that Tair and GT arrays have the same size
if Tairarraysize!=GTarraysize:
    print('Tair, GT arrays are not of equal size!')
    sys.exit()

```



```

#Importing array
Tcellarray_pv=np.loadtxt('tcellarray_pv.txt',dtype=float).reshape(-1,1)
#-----#
#Temporal array, tarray
m=Tairarraysize #number of temporal nodes
dt=3600.0 # temporal step, [s]

tarray=np.linspace(0,m-1,num=m,endpoint=True).reshape(-1,1)
#-----#
#Spatial array, xarray
n1=15 #number of nodes, material 1 (glass)
n2=5 #number of nodes, material 2 (PET)
n3=5 #number of nodes, material 3 (plastic bag)
n4=250 #number of nodes, material 4 (PCM)
n5=5 #number of nodes, material 5 (plastic bag)
n=n1+n2+n3+n4+n5 #total number of nodes
L=L1+L2+L3+L4+L5 #total length, [m]
dx1=(2*L1)/((2*n1)-1) #spatial step, material 1 (glass), [m]
dx2=(L2/n2) #spatial step, material 2 (PET), [m]
dx3=(L3/n3) #spatial step, material 3 (plastic bag), [m]
dx4=(L4/n4) #spatial step, material 4 (PCM), [m]
dx5=(2*L5)/((2*n5)-1) #spatial step, material 5 (plastic bag), [m]
i12=n1 #layer change 1-2 (glass-PET)
i23=n1+n2 #layer change 2-3 (PET-plastic bag)
i34=n1+n2+n3 #layer change 3-4 (plastic bag-PCM)
i45=n1+n2+n3+n4 #layer change 4-5 (PCM-plastic bag)

dx=np.zeros((n-1,1),dtype=float) #spatial step, [m]
for i in range(0,n-1):
    if i<(i12-1): #material 1 (glass)
        dx[i,0]=dx1
    if i==(i12-1): #layer change 1-2 (glass-PET)
        dx[i,0]=(dx1/2)+(dx2/2)
    if i>(i12-1) and i<(i23-1): #material 2 (PET)
        dx[i,0]=dx2
    if i==(i23-1): #layer change 2-3 (PET-plastic bag)
        dx[i,0]=(dx2/2)+(dx3/2)
    if i>(i23-1) and i<(i34-1): #material 3 (plastic bag)
        dx[i,0]=dx3
    if i==(i34-1): #layer change 3-4 (plastic bag-PCM)
        dx[i,0]=(dx3/2)+(dx4/2)
    if i>(i34-1) and i<(i45-1): #material 4 (PCM)
        dx[i,0]=dx4
    if i==(i45-1): #layer change 4-5 (PCM-plastic bag)
        dx[i,0]=(dx4/2)+(dx5/2)
    if i>(i45-1): #material 5 (plastic bag)
        dx[i,0]=dx5

xarray=np.zeros((n,1),dtype=float)

```

```

for i in range(1,n):
    xarray[i,0]=xarray[i-1,0]+dx[i-1,0]
#*****#
#MAIN CODE

#Create 'blank' arrays
K=np.zeros((n,1),dtype=float) #thermal conductivity per node, [W/mK]
R=np.zeros((n,1),dtype=float) #density per node, [kg/m3]
Cp=np.zeros((n,1),dtype=float) #specific heat capacity per node, [kJ/kgK]

ld=np.zeros((n-1,1),dtype=float) #lower-diagonal elements, ld
d=np.zeros((n,1),dtype=float) #diagonal elements, d
ud=np.zeros((n-1,1),dtype=float) #upper-diagonal elements, ud
b=np.zeros((n,1),dtype=float) #right side elements, b

Tarray=np.zeros((n,1),dtype=float) #Temperature distribution, [oC]

Tcellarray=np.zeros((m,1),dtype=float) #Cell temperature over time, [oC]

Tcellapproxarray=np.zeros((m,1),dtype=float) #Cell temperature approx, [oC]

Tsurfacearray=np.zeros((m,n),dtype=float) #Surface temperature, [oC]

Tdiffarray=np.zeros((m,1),dtype=float) #Cell-air temperature difference, [oC]

Tpcmstart=np.zeros((m,1),dtype=float) #PCM temperature, first node [oC]
Tpcmmid=np.zeros((m,1),dtype=float) #PCM temperature, middle node [oC]
Tpcmend=np.zeros((m,1),dtype=float) #PCM temperature, first node [oC]
#-----#
#Initial conditions
for i in range(0,n):
    Tarray[i,0]=Tairarray[0,0] #initial temperature for all nodes
#-----#
#Finding out initial temperature distribution
ci=10 #estimated number of iterations for convergence

for p in range(0,ci):

    #K,R,Cp
    for i in range(0,n):
        if i<i12: #material 1 (glass)
            K[i,0]=K1
            R[i,0]=R1
            Cp[i,0]=Cp1
        if i>=i12 and i<i23: #material 2 (PET)
            K[i,0]=K2
            R[i,0]=R2
            Cp[i,0]=Cp2
        if i>=i23 and i<i34: #material 3 (plastic bag)
            K[i,0]=K3

```

```

R[i,0]=R3
Cp[i,0]=Cp3
if i>=i34 and i<i45: #material 4 (PCM)
    Cp4=sp25a8.cpeff(Tarray[i,0])
    K[i,0]=K4
    R[i,0]=R4
    Cp[i,0]=Cp4
if i>=i45: #material 5 (plastic bag)
    K[i,0]=K5
    R[i,0]=R5
    Cp[i,0]=Cp5

#ld
for i in range(1,n-1):
    temp1=(K[i,0]*dt)
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
    temp3=(temp1/temp2)*dx[i,0]
    ld[i-1,0]=-temp3
ld[n-2,0]=-(K5/dx5) #BC at x=L

#d
for i in range(1,n-1):
    temp1=(K[i,0]*dt)
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
    temp3=1+((temp1/temp2)*(dx[i-1,0]+dx[i,0]))
    d[i,0]=temp3
d[0,0]=(K1/dx1)+H1 #BC at x=0
d[n-1,0]=(K5/dx5)+H5 #BC at x=L

#ud
for i in range(1,n-1):
    temp1=(K[i,0]*dt)
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
    temp3=(temp1/temp2)*dx[i-1,0]
    ud[i,0]=-temp3
ud[0,0]=-(K1/dx1) #BC at x=0

#b
for i in range(1,n-1):
    b[i,0]=(Tarray[i,0]+273.15)
b[0,0]=((Tairarray[p,0]+273.15)*H1)+(aglass*GTarray[p,0]*1000.0) #BC at x=0
b[n-1,0]=((Tairarray[p,0]+273.15)*H5) #BC at x=L

#Solving system using Thomas algorithm (TDMA)
Tarray=thomas(ld,d,ud,b,n)

#Reverting temperature back to Celsius
for i in range(0,n):
    Tarray[i,0]=(Tarray[i,0]-273.15)
#-----#

```

```
#Temporal loop
for p in range(0,m):
```

```
#K,R,Cp
```

```
for i in range(0,n):
```

```
    if i<i12: #material 1 (glass)
```

```
        K[i,0]=K1
```

```
        R[i,0]=R1
```

```
        Cp[i,0]=Cp1
```

```
    if i>=i12 and i<i23: #material 2 (PET)
```

```
        K[i,0]=K2
```

```
        R[i,0]=R2
```

```
        Cp[i,0]=Cp2
```

```
    if i>=i23 and i<i34: #material 3 (plastic bag)
```

```
        K[i,0]=K3
```

```
        R[i,0]=R3
```

```
        Cp[i,0]=Cp3
```

```
    if i>=i34 and i<i45: #material 4 (PCM)
```

```
        Cp4=sp25a8.cpeff(Tarray[i,0])
```

```
        K[i,0]=K4
```

```
        R[i,0]=R4
```

```
        Cp[i,0]=Cp4
```

```
    if i>=i45: #material 5 (plastic bag)
```

```
        K[i,0]=K5
```

```
        R[i,0]=R5
```

```
        Cp[i,0]=Cp5
```

```
#ld
```

```
for i in range(1,n-1):
```

```
    temp1=(K[i,0]*dt)
```

```
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
```

```
    temp3=(temp1/temp2)*dx[i,0]
```

```
    ld[i-1,0]=-temp3
```

```
ld[n-2,0]=-(K5/dx5) #BC at x=L
```

```
#d
```

```
for i in range(1,n-1):
```

```
    temp1=(K[i,0]*dt)
```

```
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
```

```
    temp3=1+((temp1/temp2)*(dx[i-1,0]+dx[i,0]))
```

```
    d[i,0]=temp3
```

```
d[0,0]=(K1/dx1)+H1 #BC at x=0
```

```
d[n-1,0]=(K5/dx5)+H5 #BC at x=L
```

```
#ud
```

```
for i in range(1,n-1):
```

```
    temp1=(K[i,0]*dt)
```

```
    temp2=(R[i,0]*Cp[i,0]*1000.0*0.5*dx[i-1,0]*dx[i,0]*(dx[i-1,0]+dx[i,0]))
```

```
    temp3=(temp1/temp2)*dx[i-1,0]
```

```
    ud[i,0]=-temp3
```

```

ud[0,0]=-(K1/dx1) #BC at x=0

#b
for i in range(1,n-1):
    b[i,0]=(Tarray[i,0]+273.15)
b[0,0]=((Tairarray[p,0]+273.15)*H1)+(aglass*GTarray[p,0]*1000.0) #BC at x=0
b[n-1,0]=((Tairarray[p,0]+273.15)*H5) #BC at x=L

#Solving system using Thomas algorithm (TDMA)
Tarray=thomas(ld,d,ud,b,n)

#Reverting temperature back to Celsius
for i in range(0,n):
    Tarray[i,0]=(Tarray[i,0]-273.15)

#Cell temperature distribution over time
Tcellarray[p,0]=Tarray[i12,0]

#Cell temperature approximaton
Tcellapproxarray[p,0]=Tairarray[p,0]+\
    ((NOCT-20.0)*((GTarray[p,0]*1000.0)/800.0))

for i in range(0,n): #surface plot, temperature distribution over x,over t
    Tsurfacearray[p,i]=Tarray[i,0]

#Cell-air temperature difference
Tdiffarray[p,0]=Tarray[i12,0]-Tairarray[p,0]

#PCM temperature
middle=math.floor((i34+i45)/2)
Tpcmstart[p,0]=Tarray[i34,0] #first PCM node
Tpcmmid[p,0]=Tarray[middle,0] #middle PCM node
Tpcmend[p,0]=Tarray[i45,0] #last PCM node

'''
#Plotting temperature distribution at each time step
plt.clf() #'clears' previous figures
plt.figure(1)
plt.title('time:=%i,Tair:=%.2f,GT:=%.2f'%(p,Tairarray[p,0],GTarray[p,0]))
plt.xlabel('x [m]')
plt.ylabel('T [oC]')
plt.axis([0,L,0,100.0])
plt.plot(xarray,Tarray,color='red')
plt.axvline(x=L1,color='blue',ls='dotted')
plt.grid(True)
plt.savefig('%i.png'%p)
'''

#'''
print('*'*80)

```

```

print('Time:= %i [hours]' %(p))
print('  Tair:= %.2f [oC]' %(Tairarray[p,0]))
print('  GT:= %.2f [kW/m^2]' %(GTarray[p,0]))
print('-'*80)
print('Tarray:')
print(Tarray,Tarray.shape)
print('Tcell:= %.2f' %(Tcellarray[p,0]))
print('*'*80)
#""

np.savetxt('tcellarray_pv_sp25a8.txt',Tcellarray) #save Tcell array into txt file
#-----#
#""
plt.figure(2)
plt.title('T-t')
plt.xlabel('t [hours]')
plt.ylabel('T [oC]')
plt.axis([0.0,m-1,0.0,100.0])
plt.plot(tarray,Tcellarray_pv,color='red',ls='dashed',label='Tcell_pv')
plt.plot(tarray,Tcellarray,color='blue',label='Tcell_sp25a8')
plt.plot(tarray,Tairarray,color='black',ls='dotted',label='Tair')
#plt.plot(tarray,Tcellapproxarray,color='magenta',ls='dashdot',label='Tclaprx')
#plt.plot(tarray,Tdiffarray,color='green',ls='dashdot',label='Tdiff')
plt.legend(loc='upper right')
plt.grid(True)
plt.savefig('tcell_pv_sp25a8.png')
plt.show()
#""

#""
plt.figure(3)
plt.title('Tsp25a8-t')
plt.xlabel('t [hours]')
plt.ylabel('T [oC]')
plt.axis([0.0,m-1,0.0,100.0])
plt.plot(tarray,Tpcmstart,color='red',label='first PCM node')
plt.plot(tarray,Tpcmmid,color='green',label='middle PCM node')
plt.plot(tarray,Tpcmend,color='blue',label='last PCM node')
plt.axhline(y=Ts,color='black',ls='dashed',label='Tsolid')
plt.axhline(y=Tl,color='black',ls='dashed',label='Tliquid')
plt.legend(loc='upper right')
plt.grid(True)
plt.savefig('tpcm_sp25a8.png')
plt.show()
#""

#""
#Surface plot, T-x over t
fig=plt.figure(4)
ax=fig.add_subplot(111,projection='3d')

```

```

sx,st=np.meshgrid(xarray,tarray)
ax.plot_surface(sx,st,Tsurfacearray,rstride=1,cstride=1,cmap='jet',linewidth=0)
ax.set_ylim(0.0,L)
ax.set_ylim(0.0,m-1)
ax.set_zlim(0.0,100.0)
ax.set_xlabel('x')
ax.set_ylabel('t')
ax.set_zlabel('T')
plt.savefig('surface_pv_sp25a8.png')
plt.show()
"""
#-----#
"""
#System matrices
systemarray=np.zeros((n,4),dtype=float)
for i in range(0,n-1):
    systemarray[i+1,0]=ld[i,0]
    systemarray[i,2]=ud[i,0]
for i in range(0,n):
    systemarray[i,1]=d[i,0]
    systemarray[i,3]=b[i,0]
for i in range(0,n):
    temp0=systemarray[i,0]
    temp1=systemarray[i,1]
    temp2=systemarray[i,2]
    temp3=systemarray[i,3]
    print("%10.2f %10.2f %10.2f %10.2f" %(temp0,temp1,temp2,temp3))
"""
#*****#

```

Γ.7-ΣΥΓΚΡΙΤΙΚΑ ΔΙΑΓΡΑΜΜΑΤΑ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΥΨΕΛΛΗΣ

comparison.py

```
#####  
#1.)COMMENTS  
#  
#Module that draws comparative plot of cell temperature over time  
#####  
#2.)MODULE IMPORTATION  
  
import numpy as np  
import matplotlib.pyplot as plt  
#####  
#3.)MAIN  
  
#Importing cell temperature from respective files  
Tair=np.loadtxt('tairarray.txt',dtype=float).reshape(-1,1)  
Tcellpv=np.loadtxt('tcellarray_pv.txt',dtype=float).reshape(-1,1)  
Tcellpvrt20=np.loadtxt('tcellarray_pv_rt20.txt',dtype=float).reshape(-1,1)  
Tcellpvrt27=np.loadtxt('tcellarray_pv_rt27.txt',dtype=float).reshape(-1,1)  
Tcellvsp25a8=np.loadtxt('tcellarray_pv_sp25a8.txt',dtype=float).reshape(-1,1)  
  
m=Tcellpv.size #temporal nodes  
  
tarray=np.linspace(0,m-1,num=m,endpoint=True).reshape(-1,1) #temporal array  
  
"""  
print('t:',tarray,tarray.shape)  
print('Tcell_pv:',Tcellpv,Tcellpv.shape)  
print('Tcell_pv:',Tcellpvrt20,Tcellpvrt20.shape)  
print('Tcell_pv:',Tcellpvrt27,Tcellpvrt27.shape)  
print('Tcell_pv_sp25a8:',Tcellvsp25a8,Tcellvsp25a8.shape)  
"""  
  
#Plotting cell temperature over time  
plt.clf() #clears' previous figures  
plt.figure(1)  
plt.axis([0.0,m-1,0.0,100.0])  
plt.xlabel('t [hours]')  
plt.ylabel("Tcell [oC]")  
plt.title("Tcell-t")  
plt.plot(tarray,Tair,color='black',ls='dotted',label='Tair')  
plt.plot(tarray,Tcellpv,color='red',ls='dashed',label='Tcell_pv')  
plt.plot(tarray,Tcellpvrt20,color='royalblue',label='Tcell_pv_rt20')  
plt.plot(tarray,Tcellpvrt27,color='skyblue',label='Tcell_pv_rt27')  
plt.plot(tarray,Tcellvsp25a8,color='cyan',label='Tcell_pv_sp25a8')  
"""  
  
plt.axhline(y=15,color='royalblue',ls='dashdot')  
plt.axhline(y=26,color='royalblue',ls='dashdot')
```



```
plt.axhline(y=22,color='skyblue',ls='dashdot')
plt.axhline(y=31,color='skyblue',ls='dashdot')
plt.axhline(y=22,color='cyan',ls='dashdot')
plt.axhline(y=32,color='cyan',ls='dashdot')
'''
plt.legend(loc='upper left')
plt.grid(True)
plt.savefig('comparison_plot.png')
plt.show()
#*****#
```

Γ.8-ΡΟΥΤΙΝΕΣ

Γ.8.1-ΡΟΥΤΙΝΑ ΕΠΙΛΥΣΗΣ ΤΡΙΔΙΑΓΩΝΙΩΝ ΣΥΣΤΗΜΑΤΩΝ

thomas_algorithm.py

```
#-----  
#Routine section  
#-----  
#Thomas algorithm  
#Simplified Gaussian elimination for linear triangular systems  
def thomas(ld,d,ud,b,n):  
    #Gauss elimination  
    #ld:lower diagonal/d:diagonal/ud:upper diagonal,n:number of nodes  
    for k in range(1,n):  
        lamda=ld[k-1,0]/d[k-1,0] #lamda for each line after the first  
        ld[k-1,0]=ld[k-1,0]-(lamda*d[k-1,0]) #new lower diagonal element (ud stays the same)  
        d[k,0]=d[k,0]-lamda*ud[k-1,0] #new diagonal element  
        b[k,0]=b[k,0]-lamda*b[k-1,0] #new constant B element  
    #Backwards substitution  
    x=np.zeros((n,1),dtype=float)  
    x[n-1,0]=b[n-1,0]/d[n-1,0]  
    for i in range(n-2,-1,-1):  
        x[i,0]=(b[i,0]-(ud[i,0]*x[i+1,0]))/d[i,0]  
  
    return x  
#-----  
#End of routine section  
#-----  
#-----  
#Module importation  
#-----  
import numpy as np  
#-----  
#End of module importation  
#-----
```

**Γ.8.2-ΡΟΥΤΙΝΑ ΥΠΟΛΟΓΙΣΜΟΥ ΕΝΕΡΓΟΥ ΘΕΡΜΟΧΩΡΗΤΙΚΟΤΗΤΑΣ ΓΙΑ ΤΟ ΥΛΙΚΟ
ΑΛΛΑΓΗΣ ΦΑΣΗΣ RT20**

effective_heat_capacity_rt20.py

```
#####  
#1.)COMMENTS  
#Effective heat capacity  
#Module that calculates Cpeff(T) [kJ/kgK or kJ/kgC] for the RT20.  
#Cpeff is a function of temperature with 6 different analytical  
#expressions for each of the temperature subregions.  
#####  
#2.)MAIN CODE  
def cpeff(Tc):  
  
    #RT20 properties  
    Ts=15.0 #Tsolid [oC]  
    Tl=26.0 #Tliquid [oC]  
    Cps=1.9 #[kJ/kgK or kJ/kgC]  
    Cpl=2.5 #[kJ/kgK or kJ/kgC]  
  
    #6 discrete temperature subregions define Cpeff  
    if Tc<Ts: #1.)T<15 [oC]  
        Cptemp=Cps  
    if Tc>=Ts and Tc<=19.5: #2.)15<=T<=19.5  
        Cptemp=(444.4711352)+\  
            (-170.5210626*Tc)+\  
            (25.80658991*(Tc**2))+\  
            (-1.912462169*(Tc**3))+\  
            (0.06910480624*(Tc**4))+\  
            (-0.0009667426392*(Tc**5))  
    if Tc>19.5 and Tc<=20.5: #3.)19.5<T<=20.5  
        Cptemp=(5529.086311)+\  
            (-565.4394794*Tc)+\  
            (14.50378915*(Tc**2))  
    if Tc>20.5 and Tc<=22.0: #4.)20.5<T<=22  
        Cptemp=(3780.001949)+\  
            (-342.8478233*Tc)+\  
            (7.797040983*(Tc**2))  
    if Tc>22.0 and Tc<=Tl: #5.)22<T<=26  
        Cptemp=(3735.906512)+\  
            (-536.9879458*Tc)+\  
            (28.98735267*(Tc**2))+\  
            (-0.6959772441*(Tc**3))+\  
            (0.006270432248*(Tc**4))  
    if Tc>Tl: #6.)T>26 [oC]  
        Cptemp=Cpl  
  
    return Cptemp  
#####
```

**Γ.8.3-ΡΟΥΤΙΝΑ ΥΠΟΛΟΓΙΣΜΟΥ ΕΝΕΡΓΟΥ ΘΕΡΜΟΧΩΡΗΤΙΚΟΤΗΤΑΣ ΓΙΑ ΤΟ ΥΛΙΚΟ
ΑΛΛΑΓΗΣ ΦΑΣΗΣ RT27**

effective_heat_capacity_rt27.py

```
#####  
#1.)COMMENTS  
#Effective heat capacity  
#Module that calculates Cpeff(T) [kJ/kgK or kJ/kgC] for the RT27.  
#Cpeff is a function of temperature with 6 different analytical  
#expressions for each of the temperature subregions.  
#####  
#2.)MAIN CODE  
def cpeff(Tc):  
  
    #RT20 properties  
    Ts=22.0 #Tsolid [oC]  
    Tl=31.0 #Tliquid [oC]  
    Cps=2.0 #[kJ/kgK or kJ/kgC]  
    Cpl=2.3 #[kJ/kgK or kJ/kgC]  
  
    #6 discrete temperature subregions define Cpeff  
    if Tc<Ts: #1.)T<22 [oC]  
        Cptemp=Cps  
    if Tc>=Ts and Tc<=25.0: #2.)22<=T<=25  
        Cptemp=(-329.500062)+\  
            (106.3162953*Tc)+\  
            (-13.35423683*(Tc**2))+\  
            (0.8223480772*(Tc**3))+\  
            (-0.02488671396*(Tc**4))+\  
            (0.0002972388922*(Tc**5))  
    if Tc>25.0 and Tc<=26.8: #3.)25<T<=26.8  
        Cptemp=(9542.790195)+\  
            (-774.7503271*Tc)+\  
            (15.74031481*(Tc**2))  
    if Tc>26.8 and Tc<=27.5: #4.)26.8<T<=27.5  
        Cptemp=(41611.7679)+\  
            (-3023.692869*Tc)+\  
            (54.96396748*(Tc**2))  
    if Tc>27.5 and Tc<=Tl: #5.)27.5<T<=31  
        Cptemp=(31155.44075)+\  
            (-3820.483549*Tc)+\  
            (175.7678837*(Tc**2))+\  
            (-3.594806096*(Tc**3))+\  
            (0.02757236338*(Tc**4))  
    if Tc>Tl: #6.)T>31 [oC]  
        Cptemp=Cpl  
  
    return Cptemp  
#####
```

**Γ.8.4-ΡΟΥΤΙΝΑ ΥΠΟΛΟΓΙΣΜΟΥ ΕΝΕΡΓΟΥ ΘΕΡΜΟΧΩΡΗΤΙΚΟΤΗΤΑΣ ΓΙΑ ΤΟ ΥΛΙΚΟ
ΑΛΛΑΓΗΣ ΦΑΣΗΣ SP25A8**

effective_heat_capacity_sp25a8.py

```
#####  
#1.)COMMENTS  
#Effective heat capacity  
#Module that calculates Cpeff(T) [kJ/kgK or kJ/kgC] for the SP25A8.  
#Cpeff is a function of temperature with 6 different analytical  
#expressions for each of the temperature subregions.  
#####  
#2.)MAIN CODE  
def cpeff(Tc):  
  
    #RT20 properties  
    Ts=22.0 #Tsolid [oC]  
    Tl=32.0 #Tliquid [oC]  
    Cps=2.6 #[kJ/kgK or kJ/kgC]  
    Cpl=2.7 #[kJ/kgK or kJ/kgC]  
  
    #6 discrete temperature subregions define Cpeff  
    if Tc<Ts: #1.)T<22 [oC]  
        Cptemp=Cps  
    if Tc>=Ts and Tc<=23.8: #2.)22<=T<=23.8  
        Cptemp=(-5822.7083)+\  
            (1559.960873*Tc)+\  
            (-166.178906*(Tc**2))+\  
            (8.806172538*(Tc**3))+\  
            (-0.2322802438*(Tc**4))+\  
            (0.002441877714*(Tc**5))  
    if Tc>23.8 and Tc<=24.5: #3.)23.8<T<=24.5  
        Cptemp=(30643.84674)+\  
            (-2584.495777*Tc)+\  
            (54.51400097*(Tc**2))  
    if Tc>24.5 and Tc<=27.2: #4.)24.5<T<=27.2  
        Cptemp=(2456.776607)+\  
            (-177.5933256*Tc)+\  
            (3.228251256*(Tc**2))  
    if Tc>27.2 and Tc<=Tl: #5.)27.2<T<=32  
        Cptemp=(-3900.417315)+\  
            (553.0906948*Tc)+\  
            (-28.64445051*(Tc**2))+\  
            (0.6468310842*(Tc**3))+\  
            (-0.00539681735*(Tc**4))  
    if Tc>Tl: #6.)T>32 [oC]  
        Cptemp=Cpl  
  
    return Cptemp  
#####
```

Δ-ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1]. Plante, R. H. (2014). "Solar energy, photovoltaics and domestic hot water". Oxford: Elsevier science. 1st edition.
- [2]. Luque, A., Hegedus, S. (2011). "Handbook of photovoltaic science and engineering." New Jersey: John Wiley & Sons. 2nd edition.
- [3]. Αντωνόπουλος, Κ. Α. (2004), "Θερμικά-ηλιακά συστήματα, μέρος πρώτο", Αθήνα: Εκδόσεις Ε.Μ.Π., Τμήμα Μηχανολόγων Μηχανικών. 1^η έκδοση.
- [4]. Hale, D. V., Hoover, M. J., O'Neill, M. J. (1971). "Phase change materials handbook", Alabama: NASA report.
- [5]. Wikipedia article (2012). "Phase change materials". Link: https://en.wikipedia.org/wiki/Phase-change_material
- [6]. Παπαμανώλης, Ν. (2015). "Άρθρο περιοδικού Econ³: Εφαρμογές των υλικών αλλαγής φάσης στα κτίρια". Σύνδεσμος: http://www.econ3.gr/readmore.php?article_id=51771295788153
- [7]. Wikipedia article (2009). "Hygroscopy". Link: <https://en.wikipedia.org/wiki/Hygroscopy>
- [8]. Incropera, F. P., DeWitt, D. P., Bergman, T. L., Lavine, A. S. (2006). "Fundamentals of heat and mass transfer". New Jersey: John Wiley & Sons. 6th edition.
- [9]. Κακάτσιος, Ξ. Κ. (2002). "Μετάδοση θερμότητας". Αθήνα: Εκδόσεις Κλειδάριθμος. 1^η έκδοση.
- [10]. Στέγγου, Α., Σαγιά Ζ. (2015). "Μετάδοση Θερμότητας", Αθήνα: Εκδόσεις Ε.Μ.Π., Τμήμα Μηχανολόγων Μηχανικών. 1^η έκδοση.
- [11]. Kyocera KC175GHT-2 specifications. Link: <http://www.gsg-rietschen.de/produkte/Module/kyocera/KC175GHT-2.pdf>
- [12]. Material properties database. Link: <http://www.makeitfrom.com/>
- [13]. Μπεργελές, Γ. (2006). "Υπολογιστική ρευστομηχανική". Αθήνα: Εκδόσεις Συμεών. Έκδοση 4^η.
- [14]. Kiusalaas, J. (2013). "Numerical methods in engineering with Python 3". New York: Cambridge university press. 1st edition.
- [15]. Γιαννάκογλου, Κ. Χ., Αναγνωστόπουλος, Ι., Μπεργελές, Γ. (2003). "Αριθμητική ανάλυση για μηχανικούς". Αθήνα: Εκδόσεις Ε.Μ.Π., Τμήμα Μηχανολόγων Μηχανικών. 3^η έκδοση.
- [16]. Μετεωρολογικά δεδομένα σταθμού Ε.Μ.Π. Σύνδεσμος: <http://openmeteo.org/stations/d/1334/>
- [17]. Kravvaritis, E. D., Antonopoulos, K. A., Tzivanidis, C. (2011). "Experimental determination of the effective heat capacity function and other thermal properties for various phase change materials using the thermal delay model", Athens: Elsevier science.

[18]. Rubitherm, phase change material properties. Link: <http://www.rubitherm.de/english/>

[19]. Mertens, K. (2014). "Photovoltaics. Fundamentals, technology and practice". New Jersey: John Wiley & Sons. 1st edition.