



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

**ΣΧΟΛΗ**

**ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**

**ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
ΜΑΘΗΜΑΤΙΚΗ ΠΡΟΤΥΠΟΠΟΙΗΣΗ ΣΕ ΣΥΓΧΡΟΝΕΣ  
ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΤΗΝ ΟΙΚΟΝΟΜΙΑ**

**ΜΑΘΗΜΑΤΙΚΗ ΠΡΟΤΥΠΟΠΟΙΗΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ  
ΑΛΛΟΙΩΣΗΣ ΥΛΙΚΩΝ ΛΟΓΩ ΔΙΑΧΥΣΗΣ ΣΕ ΣΥΝΘΗΚΕΣ ΠΤΗΣΗΣ  
ΚΑΙ  
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΕ MATLAB**

Ευαγγελία Αλεξάνδρου Φτάκα

Επιβλέπων Καθηγητής: **Δημήτριος Τζανετής, Ομ.Καθηγητής ΕΜΠ**

Εξεταστική Επιτροπή: **Δημήτριος Τζανετής, Ομ.Καθηγητής ΕΜΠ**  
**Κυριακή Κυριάκη, Καθηγήτρια ΕΜΠ**  
**Βασίλειος Παπανικολάου, Καθηγητής ΕΜΠ**

ΑΘΗΝΑ, Δεκέμβριος 2014



Ευχαριστώ τους κ.κ Riccardo Rampini και Marc Van Eesbeek για την εμπιστοσύνη στο πρόσωπό μου και την αδιάλειπτη στήριξή τους κατά τη διάρκεια αυτής της προσπάθειας.  
Ευχαριστώ τους καθηγητές μου, κ.κ. Δημήτριο Τζανετή, Νικόλαο Σταυρακάκη και Κυριακή Κυριάκη, για την ευκαιρία-παρότρυνση ολοκλήρωσης της παρούσας εργασίας.

Αθήνα, Δεκέμβριος 2014



## Π Ε Ρ Ι Λ Η Ψ Η

Τα στερεά υλικά περιέχουν προσμείξεις, εκ κατασκευής ή εκ της διαδικασίας καθαρισμού τους, που διαχέονται στο περιβάλλον όταν εκτίθενται σε συνθήκες κενού, ιδιαίτερα σε υψηλές θερμοκρασίες. Μερικά υλικά όπως τα πλαστικά και τα ακρυλικά τείνουν να αποσυντίθενται υπό συνθήκες κενού. Επίσης, ουσίες που απορροφήθηκαν από τα διάφορα υλικά κατά τον χρόνο κατασκευής των σκαφών, και κατά τη διάρκεια καθαρισμού, δοκιμής και εκτόξευσης αυτών είναι δυνατόν να απελευθερωθούν από την επιφάνειά τους.

Αυτά τα αποδεσμευμένα από τα υλικά προϊόντα αποτελούν μια μόνιμη πηγή μόλυνσης για τα σκάφη. Ειδικά τα πολυμοριακά είδη όπως διάφορα τμήματα αλυσίδας υδρογοναναθράκων και οι σιλκόνες είναι τα πιο επικίνδυνα εξαιτίας της τάσης τους να εναποτίθενται σε ευαίσθητες λειτουργικές επιφάνειες. Αυτού του είδους η μόλυνση γίνεται μείζον ζήτημα σε μια μεγάλης διάρκειας πτήση αλλοιώνοντας τις οπτικές και ηλεκτρικές ιδιότητες διαφόρων στοιχείων που απαρτίζουν τα σκάφη και την ευστάθειά τους υπό ακτινοβολία.

Για τον σκοπό αυτό, ο ρυθμός διάχυσης κρίσιμων υλικών πρέπει να προσδιοριστεί. Δυναμικά τεστ διεξάγονται για τη μελέτη του μεγέθους διάχυσης και της χρονικής εξέλιξης αυτής και επιχειρούνται μακροπρόθεσμες προβλέψεις.

Η μέθοδος προσδιορισμού της κινητικής του φαινομένου διάχυσης ενός δείγματος και της εναπόθεσης των μολυντών σε συλλέκτες υπό συνθήκες κενού που ορίζει το ESTEC καλείται VBQC (Vacuum Balance Quartz Crystal – Ζυγός Κρυστάλλου Χαλαζία υπό συνθήκες κενού). Το σύστημα βασίζεται στη χρήση Μικροζυγών Κρυστάλλου Χαλαζία (QCMs - Quartz Crystal Microbalances) για τη μέτρηση της μάζας του υλικού που προέρχεται από το δείγμα και εναποτίθεται, άρα μολύνει, τις λειτουργικές επιφάνειες των σκαφών, σε συνδυασμό με τον ζυγό για την μέτρηση της συνολικής απώλειας μάζας του δείγματος λόγω διάχυσης.

Το πρότυπο τεστ που διενεργείται με το πιο πάνω σύστημα είναι ένα δυναμικό τεστ με την θερμοκρασία του δείγματος να αυξάνεται από τους 25 °C στους 125 °C με βήμα 25 °C κάθε 24 ώρες. Η δυναμική προσέγγιση χρησιμοποιείται για να επιταχυνθεί το φαινόμενο διάχυσης. Έπειτα από μαθηματική επεξεργασία των δεδομένων, οι παράμετροι που είναι απαραίτητες για την μοντελοποίηση του φαινομένου διάχυσης (acceleration factor – παράγων επιτάχυνσης, residence time – χρόνος παραμονής, activation energy – ενέργεια ενεργοποίησης) και για τις μακροπρόθεσμες προβλέψεις προσδιορίζονται.

Στο παρελθόν, η μαθηματική επεξεργασία γινόταν με τη βοήθεια διαφόρων αρχείων-φύλλων εργασίας Excel®. Η εκτέλεση όλης της ακολουθίας απαιτούσε μεγάλη ανθρώπινη παρέμβαση και υποκειμενικούς χειρισμούς. Στην παρούσα εργασία παρουσιάζεται η μαθηματική επεξεργασία των δεδομένων διάχυσης με τρόπο ευθύ και τυποποιημένο. Για τον σκοπό αυτό, αναπτύχθηκε ένα πρόγραμμα MatLab®. Αυτή η δραστηριότητα ελαχιστοποίησε την ανθρώπινη παρέμβαση προς αποφυγή αυθαίρετων επιλογών που καθιστούν την μέθοδο αναξιόπιστη ή/και συγκρούονται με τη φυσική του προβλήματος.



## ABSTRACT

Solid materials contain impurities from manufacturing / cleaning processes which diffuse (outgas) to the exterior during exposure to vacuum, especially at elevated temperature levels. Some materials like plastics and resins tend to disintegrate under space conditions. Apart from the above mentioned processes out of the material, adsorbed and absorbed contaminants from contaminant environmental conditions encountered previously, e.g. during manufacturing, cleaning, integration, test and launch phases, might be desorbed from the surfaces.

The released products represent a permanent source of contamination. Especially the high molecular species like polymers, hydrocarbon chain fragments and silicones are most dangerous due to their tendency of forming deposits at sensitive and operational surfaces. This contamination can become an important issue after a long period in flight and alter the optical, electrical and physical properties of different onboard elements and their stability under energetic irradiation.

For this purpose, the outgassing rate of critical materials has to be characterized. Dynamic routine tests are then rather compulsory to analyze the magnitude of outgassing and the temporal evolution of this outgassing rate and then carry out long term predictions.

The test method defined at ESTEC to study the kinetics of the outgassing of a sample and of the deposition of the contaminants under vacuum utilizes systems called VBQC (Vacuum Balance Quartz Crystal). The systems are based on the use of QCMs (Quartz Crystal Microbalances) to measure the CVCM (Collected Volatile and Condensable Materials) coming from the sample, coupled with vacuum balance, to measure the TML (Total Mass Loss) from the sample due to outgassing.

The standard test performed with this system is a multi-step temperature test where the sample temperature is increased from 25 °C to 125 °C by step of 25 °C every 24 hours. The multi-step approach is utilized to accelerate the outgassing phenomenon. The evolution of the TML and CVCM during the test is recorded and, after mathematical treatment of the collected data, the parameters that are necessary for the modeling of the outgassing phenomenon (acceleration factor, residence time, activation energy) and for the long-term prediction are then determined.

In the past years, the mathematical treatment was done through different Microsoft® Excel worksheets. The execution of the complete sequence required rather lot of manual intervention guided by a work instruction procedure but also requiring subjective operator decisions. To have a more straightforward and standardized data treatment a MatLab® application has been developed and it is here presented. The scope of the related activity was to minimize the manual intervention and to avoid any arbitrary choice that could be in contradiction with the implied physics.





# Π Ε Ρ Ι Ε Χ Ο Μ Ε Ν Α

<b>1</b>	<b>Ο ΕΥΡΩΠΑΪΚΟΣ ΟΡΓΑΝΙΣΜΟΣ ΔΙΑΣΤΗΜΑΤΟΣ.....</b>	<b>11</b>
1.1	Μέλη του ΕΟΔ	11
1.2	Σκοπός του ΕΟΔ	11
1.3	Προγράμματα του ΕΟΔ	11
1.4	Διοίκηση και Εγκαταστάσεις του ΕΟΔ	12
1.5	Το Ευρωπαϊκό Κέντρο Διαστημικής Έρευνας και Τεχνολογίας	13
1.6	Τα Εργαστήρια Επεξεργασίας Υλικών	13
1.7	Το Τμήμα Υλικών, Φυσικής και Χημείας	14
<b>2</b>	<b>ΕΙΣΑΓΩΓΗ.....</b>	<b>15</b>
2.1	Χρήσιμοι Ορισμοί	16
2.2	Μόλυνση λόγω Διάχυσης	17
<b>3</b>	<b>ΑΠΟΣΠΑΣΜΑ ΤΗΣ ΘΕΩΡΙΑΣ ΤΩΝ M.VAN EESBEEK &amp; A. ZWAAL .....</b>	<b>19</b>
3.1	Εισαγωγή	19
3.2	Προσέγγιση Καμπυλών Διάχυσης και Συμπύκνωσης	20
3.2.1	Καμπύλη Διάχυσης	20
3.2.1.1	Μαθηματική Προσαρμογή Δεδομένων	21
3.2.1.2	Επίδραση της Θερμοκρασίας στη Διάχυση	21
3.2.1.3	Προσομοίωση Καμπύλης Διάχυσης	21
3.2.2	Καμπύλη Συμπύκνωσης στον Συλλέκτη	22
3.3	Αποτελέσματα των Τεστ	22
<b>4</b>	<b>Η ΜΕΘΟΔΟΣ ΤΟΥ ESTEC.....</b>	<b>23</b>
4.1	Οι Βασικές Εξισώσεις	23
4.2	Περιγραφή της Μεθόδου	24
4.3	Προβλέψεις	26
4.4	Η Διάταξη VBQC	27
<b>5</b>	<b>ΜΑΘΗΜΑΤΙΚΟΣ ΧΕΙΡΙΣΜΟΣ.....</b>	<b>29</b>
5.1	Τα Δεδομένα	29
5.2	Γενική Προσαρμογή Δεδομένων με Ελάχιστα Τετράγωνα	31
5.3	Η Πρώτη Προσαρμογή	33
5.3.1	Πρώτη Μέθοδος - Χρήση 1 <sup>ης</sup> Παραγώγου	34
5.3.2	Δεύτερη Μέθοδος - Χρήση 2 <sup>ης</sup> Παραγώγου	36
5.3.3	Εναλλακτικές	39
5.4	Διαδικασία Επιτάχυνσης	43
5.5	Μακροπρόθεσμες Προβλέψεις	44
<b>6</b>	<b>ΕΚΤΕΛΕΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ.....</b>	<b>47</b>
<b>7</b>	<b>ΣΥΜΠΕΡΑΣΜΑΤΑ .....</b>	<b>53</b>
<b>8</b>	<b>ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ.....</b>	<b>55</b>
<b>9</b>	<b>ΚΩΔΙΚΕΣ MATLAB®.....</b>	<b>57</b>
9.1	AccelFactor.m	57
9.2	AccelTau.m	58
9.3	AccelTime.m	58
9.4	ActivEnergy.m	59
9.5	definetau.m	60
9.6	definetau_alter.m	60
9.7	definew.m	61

9.8	definew_alter.m	62
9.9	definewagain.m	64
9.10	derivate2.m	65
9.11	derivate3.m	65
9.12	derivate.m	66
9.13	everystep.m	67
9.14	execute.m	69
9.15	findPointC.m	127
9.16	fitdata.m	127
9.17	forLongTermFit.m	129
9.18	forLongTermFit_alter.m	129
9.19	forPointC.m	130
9.20	forValueC.m	131
9.21	gauss.m	131
9.22	generate.m	132
9.23	KeCoefficient.m	132
9.24	LongTermPrediction.m	134
9.25	LongTermTau.m	134
9.26	Tsqerror.m	135

# 1. Ο Ευρωπαϊκός Οργανισμός Διαστήματος (ΕΟΔ-ESA)

Η ιδέα ενός ανεξάρτητου ευρωπαϊκού οργανισμού διαστήματος χρονολογείται από τις αρχές του 1960. Ο ΕΟΔ ιδρύθηκε το 1975, αντικαθιστώντας τους οργανισμούς δορυφόρων και εκτόξευσης ESRO και ELDO.

## 1.1. Μέλη του ΕΟΔ

Στον ΕΟΔ συμμετέχουν 18 κράτη-μέλη: Αυστρία, Βέλγιο, Γαλλία, Γερμανία, Δανία, Ελβετία, Ελλάδα, Ηνωμένο Βασίλειο, Ιρλανδία, Ισπανία, Ιταλία, Λουξεμβούργο, Νορβηγία, Ολλανδία, Πορτογαλία, Σουηδία, Τσεχία και Φινλανδία.

Ο Καναδάς συμμετέχει σε ορισμένα προγράμματα κατόπιν συμφωνίας.

Η Ουγγαρία, η Ρουμανία, η Πολωνία, η Εσθονία και η Σλοβενία είναι συνεργαζόμενα μέλη από την Ευρώπη.

## 1.2. Σκοπός του ΕΟΔ

Σκοπός είναι ο σχεδιασμός και η υλοποίηση του Ευρωπαϊκού διαστημικού προγράμματος. Τα προγράμματα του Οργανισμού σχεδιάζονται με σκοπό τον εμπλουτισμό της γνώσης μας σχετικά με τη Γη, το άμεσο διαστημικό της περιβάλλον, το Ηλιακό Σύστημα και το Σύμπαν, όπως και την ανάπτυξη βασισμένων στους δορυφόρους τεχνολογιών και υπηρεσιών και την προώθηση των Ευρωπαϊκών βιομηχανιών. Ο ΕΟΔ συνεργάζεται επίσης στενά με διαστημικούς οργανισμούς εκτός Ευρώπης, ώστε να επωφελείται όλη η ανθρωπότητα από το διάστημα.

## 1.3. Προγράμματα του ΕΟΔ

Όλα τα κράτη μέλη συμμετέχουν σε δραστηριότητες και προγράμματα διαστημικής επιστήμης υποχρεωτικά. Τα μέλη όμως επιλέγουν τον βαθμό συμμετοχής τους σε προγράμματα που σχετίζονται με:

επανδρωμένες διαστημικές αποστολές και εξερεύνηση

μικροβαρυτικές έρευνες

την παρατήρηση της Γης

τηλεπικοινωνίες

δορυφορική πλοήγηση

και ανάπτυξη πυραύλων

• Human spaceflight and Exploration

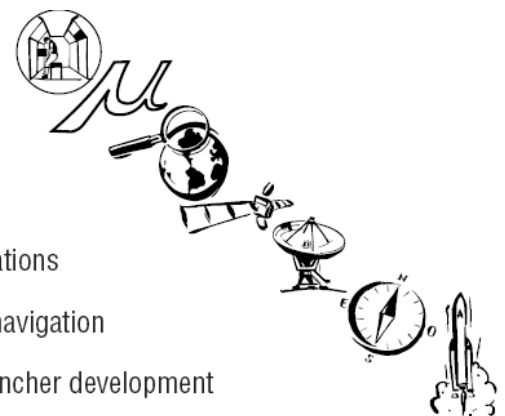
• Microgravity research

• Earth observation

• Telecommunications

• Satellite navigation

• Launcher development



Ειδικότερα, κάποια από τα προγράμματα που χειρίζεται ο ΕΟΔ είναι:

το Διαστημικό τηλεσκόπιο Χαμπλ, σε συνεργασία με την NASA

τον Envisat, ο οποίος είναι ο πιο μεγάλος δορυφόρος για την παρατήρηση της Γης

τον δορυφόρο Artemis που είναι ο πιο σύγχρονος δορυφόρος τηλεπικοινωνιών

τον δορυφόρο Giove-A για το πρόγραμμα Galileo (ευρωπαϊκή απάντηση για το GPS)

το μη επανδρωμένο διαστημόπλοιο Mars Express για την παρατήρηση του Άρη

το μη επανδρωμένο διαστημόπλοιο Venus Express για την παρατήρηση της Αφροδίτης

ο τεχνητός δορυφόρος της Σελήνης Smart 1

το όχημα εξερεύνησης του δορυφόρου του Κρόνου Τιτάνα Huygens

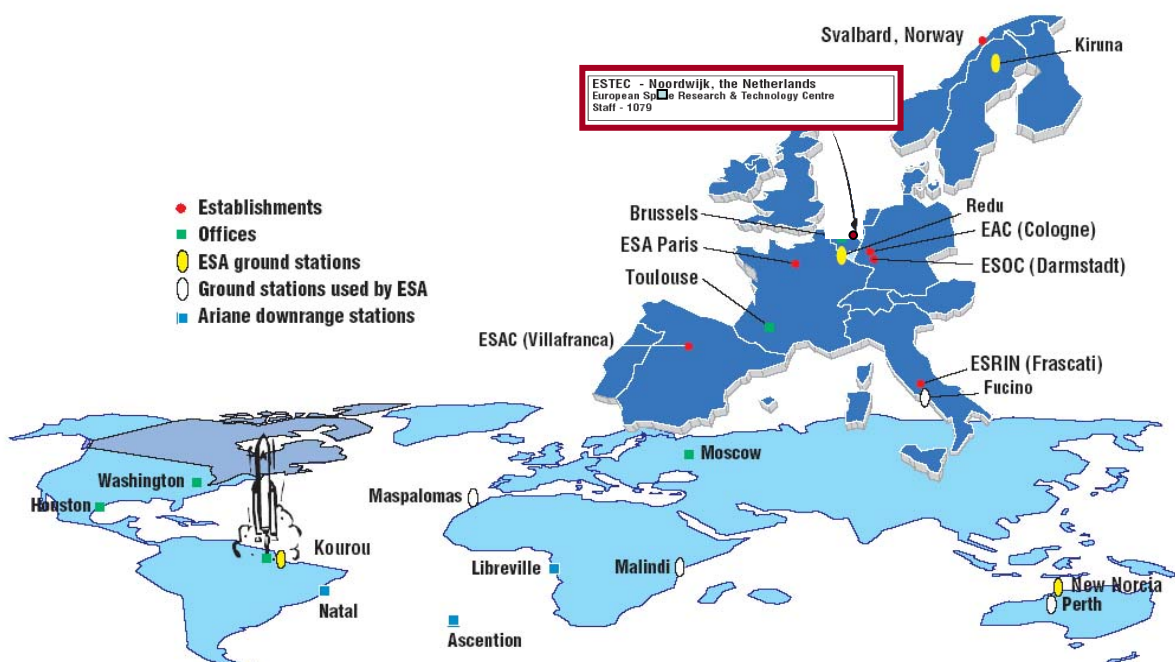
το διαστημόπλοιο Πλανκ

το ΒεριColombo για την παρατήρηση του Ερμή

το ExoMars για την εξερεύνηση του Άρη

το τηλεσκόπιο Gaia probe το οποίο θα κάνει τρισδιάστατη χαρτογράφηση του Γαλαξία μας.

#### 1.4. Διοίκηση και Εγκαταστάσεις του ΕΟΔ



Τα κεντρικά γραφεία του ΕΟΔ βρίσκονται στο Παρίσι, όπου λαμβάνονται οι αποφάσεις σχετικά με τις πολιτικές και τα προγράμματα. Ωστόσο, ο ΕΟΔ έχει επίσης έναν αριθμό από επιχειρησιακά και ερευνητικά κέντρα στην Ευρώπη, κάθε ένα από τα οποία έχει διαφορετικά καθήκοντα.

- Το EAC (European Astronauts Centre ή Ευρωπαϊκό Κέντρο Αστροναυτών), το οποίο βρίσκεται στην Κολωνία της Γερμανίας
- Το ESAC (European Space Astronomy Centre ή Ευρωπαϊκό Κέντρο Διαστημικής Αστρονομίας), το οποίο βρίσκεται στην Villanueva de la Canada, κοντά στη Μαδρίτη της Ισπανίας
- Το ESOC (European Space Operations Centre ή Ευρωπαϊκό Κέντρο Διαστημικών Επιχειρήσεων), το οποίο βρίσκεται στο Darmstadt της Γερμανίας
- Το ESRIN (European Space Research Institute ή Ευρωπαϊκό Ινστιτούτο Διαστημικών Ερευνών), το οποίο βρίσκεται στο Frascati, κοντά στη Ρώμη της Ιταλίας.
- Το ESTEC (European Space Research and Technology Centre ή Ευρωπαϊκό Κέντρο Διαστημικής Έρευνας και Τεχνολογίας), το οποίο βρίσκεται στο Noordwijk της Ολλανδίας.

Ένα νέο κέντρο εγκαταστάθηκε στο Harwell, κοντά στο Oxfordshire του Ηνωμένου Βασιλείου. Επιπλέον, ο ΕΟΔ διαθέτει γραφεία διασύνδεσης στο Βέλγιο, τις Ηνωμένες Πολιτείες και τη Ρωσία, μια βάση εκτόξευσης στη Γαλλική Γουιάνα και επίγειους σταθμούς παρακολούθησης σε διάφορες περιοχές του κόσμου.

### **1.5. Το Ευρωπαϊκό Κέντρο Διαστημικής Έρευνας και Τεχνολογίας (ESTEC)**

Όπως αναφέρθηκε, ο ΕΟΔ έχει εγκαταστάσεις σε διάφορες ευρωπαϊκές χώρες. Το ευρωπαϊκό κέντρο διαστημικής έρευνας και τεχνολογίας, η τεχνολογική καρδιά του ΕΟΔ, βρίσκεται στο Noordwijk της Ολλανδίας. Τα περισσότερα προγράμματα του ΕΟΔ γεννιούνται και αναπτύσσονται σ' αυτό. Περισσότεροι από 2000 επιστήμονες και μηχανικοί εργάζονται σε δεκάδες διαστημικά προγράμματα.

Εκτός από τις εκτοξεύσεις, το ESTEC διαχειρίζεται σχεδόν όλα τα προγράμματα του ΕΟΔ. Στο Noordwijk, οι ειδικοί εργάζονται σε επιστημονικές αποστολές, επανδρωμένες ή μη πτήσεις, τηλεπικοινωνίες, στη δορυφορική πλοήγηση και την παρατήρηση της Γης.

### **1.6. Τα Εργαστήρια Επεξεργασίας Υλικών**

Τα εργαστήρια επεξεργασίας υλικών είναι απαραίτητα στην ευρωπαϊκή διαστημική βιομηχανία. Οι μηχανικοί του οργανισμού έχουν αναπτύξει το μεγαλύτερο μέρος του εξοπλισμού που στεγάζεται σε αυτά τα εργαστήρια αποκλειστικά για τις αξιολογήσεις υλικών που χρησιμοποιούνται και πρέπει να παραμένουν λειτουργικά υπό τους περιορισμούς του διαστημικού περιβάλλοντος.

Οι εγκαταστάσεις του τμήματος παρέχουν υπηρεσίες στον ΕΟΔ, σε διεθνείς διαστημικούς οργανισμούς και σε εταιρείες συμβαλλόμενες με τον ΕΟΔ για την κατασκευή διαστημικού υλικού. Οι μηχανικοί λαμβάνουν πολυδάπανες και μακροπρόθεσμες αποφάσεις για τις μελλοντικές γενιές δορυφόρων και σκαφών, βασιζόμενοι στα αποτελέσματα των τεστ που διενεργούνται σε αυτά τα εργαστήρια.

Μεταξύ άλλων, στα εργαστήρια επεξεργασίας υλικών γίνεται ανάλυση υλικών για τον προσδιορισμό αποτυχίας αυτών που πιθανώς θα οδηγήσουν σε ατυχήματα κατά τη διάρκεια της κατασκευής, μεταφοράς και δοκιμής των σκαφών. Τα τεστ των εργαστηρίων προσδιορίζουν και επικυρώνουν τα κατάλληλα υλικά για το διαστημικό περιβάλλον.

Το εργαστήριο ενεργεί για τις ακόλουθες περιοχές:

- την περιβαλλοντική επίδραση στα υλικά, όπως απαέρωση, μόλυνση, UV/μοριακή ακτινοβολία, ηλεκτροστατική φόρτιση, ατομική διάβρωση και συμπεριφορά υπό την επίδραση θραυσμάτων/μικρομετεωριτών
- την περιβαλλοντική επίδραση στα υλικά κατά την αποθήκευση, όπως καταπόνηση και βιομεταλλική διάβρωση
- τεχνικές αναλύσεις και διερεύνηση αποτυχίας όλων των διαδικασιών παραγωγής/κατασκευής, συμπεριλαμβανομένων των πολυμερών και κεραμικών υλικών

## **1.7. Το Τμήμα Υλικών, Φυσικής και Χημείας**

Ο κύριος ρόλος του τμήματος είναι η διασφάλιση της τεχνικής υποστήριξης των προγραμμάτων του ΕΟΔ, με έμφαση στη χρήση μη-μεταλλικών υλικών, καλύπτοντας επίσης τον θερμικό έλεγχο αυτών, οργανικά και ανόργανα συγκολλητικά υλικά, συνθετικά υλικά κλπ, σε διαστημικές εφαρμογές. Αυτό περιλαμβάνει την αξιολόγηση των γενικών χαρακτηριστικών των υλικών σε συνδυασμό με την επίδραση των διαφόρων περιβαλλοντικών παραγόντων του διαστήματος για επανδρωμένες και μη αποστολές.

Κεντρικός στόχος του τμήματος είναι ο προσδιορισμός και η αξιολόγηση των διαστημικών περιβαλλοντικών επιδράσεων στα υλικά, την επίδραση της ακτινοβολίας (μοριακή, UV κλπ), την αντίσταση στην οξείδωση. Το τμήμα ευθύνεται για την προτυποποίηση των περιβαλλοντικών επιδράσεων με στόχο μακροπρόθεσμες προβλέψεις αντοχής/ζωής των υλικών όπως προκύπτουν από διεξαγόμενα τεστ θερμοκρασιακής επιτάχυνσης.

Το τμήμα διαθέτει εργαστηριακές εγκαταστάσεις που παρέχουν υπηρεσίες σε όλες τις αποστολές του ΕΟΔ, υποστήριξη στα τεχνολογικά προγράμματα του ESTEC, αντιμετώπιση όλων των αξιολογικών απαιτήσεων βιομηχανικών συνεργατών και ευρωπαϊκών εθνικών προγραμμάτων.

## 2. ΕΙΣΑΓΩΓΗ

Η μόλυνση μπορεί να οριστεί ως οποιαδήποτε ξένη ουσία. Γενικά, κατηγοριοποιείται σε δυο ευρείες ομάδες, την μοριακή και τη σωματιδιακή.

Η μοριακή μόλυνση αναφέρεται στη συσσώρευση ξεχωριστών ατόμων ξένης ουσίας. Για παράδειγμα, η γνωστή οσμή των πλαστικών ή «μυρωδιά νέου αυτοκινήτου». Αυτή είναι ένδειξη πτητικών μορίων που προέρχονται από οργανικά υλικά. Η μοριακή μόλυνση είναι πιθανότερη κατά την επεξεργασία εδάφους, αλλά είναι περισσότερο ανησυχητική σε τροχιά.

Η σωματιδιακή μόλυνση αναφέρεται στην εναπόθεση ορατών μικρού μεγέθους συνονθυλευμάτων ύλης. Οι επιφάνειες που σκονίζονται και τα γυαλιά που απαιτούν περιοδικό καθάρισμα είναι ενδείξεις της παρουσίας σωματιδίων στην ατμόσφαιρα. Αυτά τα σωματίδια, που κυρίως συγκεντρώνονται σε επιχειρήσεις εδάφους, θα επηρεάσουν εκτεθειμένες επιφάνειες.

Αποτελεσματικός έλεγχος μολύνσεων είναι απαραίτητος για την επιτυχία διαστημικών προγραμμάτων γιατί η παρουσία μολυντών, ακόμα και σε μικροσκοπικές ποσότητες, επιδεινώνει την απόδοση του υλικού των σκαφών. Η παρουσία μολυσματικών ουσιών σε επιφάνειες θερμικού ελέγχου θα αλλοιώσει την αναλογία απορρόφησης/εκπομπής και θα μεταβάλει τη θερμική ισορροπία, ενώ οι μολυντές σε φωτοηλεκτρικές διατάξεις θα μειώσουν την απόδοση ενέργειας. Σε θερμικά όργανα οι μολυντές θα μειώσουν τη ρυθμαπόδοση σημάτων και θα τα διασκορπίσουν πέρα από τη ρυθμισμένη διάθλαση, συνεπώς θα μειώσουν περαιτέρω την απόδοση [2].

Ο έλεγχος μολύνσεων για διαστημικά προγράμματα είναι μια επαναληπτική διαδικασία που εξαρτάται από τους στόχους της αποστολής.

Τα περισσότερα σκάφη διαθέτουν όργανα που λειτουργούν με τηλεχειριστήριο και είναι εξαιρετικά ευαίσθητα. Αν ο στόχος της αποστολής περιλαμβάνει συλλογή δεδομένων με απομακρυσμένους αισθητήρες, αυτό καθορίζει τον τύπο των απαιτούμενων οπτικών οργάνων σε όρους συχνοτήτων, ισχύος σήματος, ευκρίνειας και λοιπών σχετικών παραμέτρων. Οι περιορισμοί που αυτές οι τιμές θέτουν στον σχεδιασμό του σκάφους θα επηρεάσουν άλλες παραμέτρους της αποστολής όπως το υψόμετρο της τροχιάς, την κλίση, την εκκεντρότητα, τη σχέση με άλλους δορυφόρους στον αστερισμό (αν είναι εφαρμόσιμη) κλπ. Εφόσον επιτευχθεί το κατάλληλο βέλτιστο στο σχεδιασμό του σκάφους, καθορίζονται οι οριακές τιμές μόλυνσης του φορτίου, των επιφανειών θερμικού ελέγχου και των φωτοηλεκτρικών διατάξεων για την σωστή λειτουργία του σκάφους. Αυτοί οι περιορισμοί μόλυνσης επιβάλλουν περαιτέρω περιορισμούς στην επιλογή των υλικών, στο σύστημα εξαερισμού, τη θέση του συστήματος προώθησης, στην κατανάλωση ενέργειας, στη θερμοκρασία του σκάφους κλπ. Ίσως είναι απαραίτητη η επανάληψη του σχεδιασμού του σκάφους και των υποσυστημάτων του πολλές φορές για να επιτευχθεί ο οικονομικότερος σχεδιασμός [2].

## 2.1. ΧΡΗΣΙΜΟΙ ΟΡΙΣΜΟΙ [3]

### Καθαριότητα/Αποστείρωση

Μια εδραιωμένη μέγιστη επιτρεπτή ποσότητα μολυσματικών ουσιών κατ' όγκο ή κατά συστατικό.

### Αποστειρωμένος χώρος

Κλειστός χώρος όπου γίνεται έλεγχος της σωματιδιακής και μοριακής ύλης στον αέρα, μαζί με έλεγχο θερμοκρασίας, υγρασίας και πίεσης, ως ορίζεται. Μαζί με την καθαριότητα του αέρα, ο αποστειρωμένος χώρος ορίζει σχεδιαστικές και λειτουργικές απαιτήσεις (φιλτράρισμα αέρα, ρυθμός ροής αέρα κλπ) και τη μέγιστη επιτρεπτή μόλυνση του αέρα.

### Μολυντής

Συγκεκριμένο είδος μόλυνσης.

### Μόλυνση

Κάθε ξένο σώμα. Πιο συγκεκριμένα, ανεπιθύμητα ξένα υλικά (σωματιδιακά και μοριακά) πάνω σε επιφάνειες ή ενσωματωμένα σε υγρά και αέρια. Σε τροχιά, περιλαμβάνονται επίσης ιπτάμενα σωματίδια στο πεδίο ενός αισθητήρα.

### Έλεγχος Μόλυνσης

Κάθε οργανωμένη ενέργεια ελέγχου του επιπέδου μόλυνσης.

### Μοριακή μόλυνση

Ανεπιθύμητη ξένη ύλη αορίστων διαστάσεων. Αυτό περιλαμβάνει διαβρωτικά και μη λεπτά στρώματα λαδιού, λίπους, χημικά κατάλοιπα, αποτυπώματα, καθώς και οτιδήποτε προέρχεται από θερμικές και χημικές εφαρμογές και ασύμβατα υλικά. Αυτά προέρχονται από μια διαδικασία καλούμενη διάχυση. Οι μοριακοί μολυντές συχνά σχηματίζουν σταγόνες και χάντρες που είναι καλύτερα να αντιμετωπίζονται ως σωματιδιακοί μολυντές.

### Σωματιδιακή μόλυνση

Ανεπιθύμητη ξένη ουσία μικρού μεγέθους, ορατού όμως μήκους, πλάτους και πάχους.

### Συνολική απώλεια μάζας - Total Mass Loss (TML)

Συνολική ποσότητα υλικού που έχει διαχυθεί από κάποιο είδος υλικού το οποίο διατηρείται σε συγκεκριμένη σταθερή θερμοκρασία και πίεση για συγκεκριμένο χρόνο. Η TML υπολογίζεται από τη διαφορά μάζας του υλικού πριν και μετά το τεστ και εκφράζεται ως ποσοστό της αρχικής μάζας.

### Πτητικό συμπυκνώσιμο υλικό - Volatile Condensable Material (VCM)

Το μέρος του υλικού που έχει διαχυθεί και ίσως συγκεκρωθεί από κάποιον συλλέκτη, συνήθως κάποιοι με χαμηλότερη θερμοκρασία.

### Συγκεντρωμένο πτητικό συμπυκνώσιμο υλικό - Collected Volatile Condensable Material (CVCM)

Η ποσότητα του υλικού που έχει διαχυθεί και συγκεντρωθεί σε κάποιον συλλέκτη με συγκεκριμένη σταθερή θερμοκρασία. Το CVCM εκφράζεται ως ποσοστό της αρχικής



μάζας του υπό εξέταση υλικού και υπολογίζεται από τη διαφορά μάζας στον δίσκο του συλλέκτη πριν και μετά το τεστ.

## 2.2. ΜΟΛΥΝΣΗ ΛΟΓΩ ΔΙΑΧΥΣΗΣ

Τα πειραματικά δεδομένα υποδεικνύουν ότι η διάχυση ποικίλει ως:

- (i) συνάρτηση του χρόνου
  - (ii) αντίστροφη συνάρτηση κάποιας δύναμης του χρόνου, ή
  - (iii) ανεξάρτητα από τον χρόνο, ανάλογα με τον μηχανισμό της διαδικασίας διάχυσης.
- Αυτές οι τρεις διαδικασίες διάχυσης είναι γνωστές ως εκρόφιση/*desorption*, διάχυση/*diffusion* και αποσύνθεση/*decomposition*, αντίστοιχα.

Εκρόφιση είναι η απελευθέρωση μορίων που βρίσκονται στην επιφάνεια του υλικού και συγκρατούνται λόγω ηλεκτρικών χημικών δυνάμεων.

Διάχυση/*diffusion* είναι η ομογενοποίηση που συμβαίνει λόγω τυχαίων θερμικών κινήσεων. Οι μολυντές που διαχέονται από την επιφάνεια ενός υλικού ίσως έχουν αρκετή θερμική ενέργεια για να διαφύγουν της επιφάνειας και εξατμίζονται στο περιβάλλον.

Αποσύνθεση είναι ένας τύπος χημικής αντίδρασης, κατά την οποία μια ένωση διαιρείται σε δυο ή περισσότερα απλούστερα στοιχεία, που εν συνεχεία διαχέονται μέσω εκρόφισης ή διάχυσης/*diffusion*.

Εκτός από την χρονική εξάρτηση, κάθε διαδικασία φαίνεται να εξαρτάται εκθετικά από ένα μοναδικό εύρος τιμών ενέργειας ενεργοποίησης,  $E_a$  (η απαιτούμενη ενέργεια για να ξεκινήσει η διαδικασία), και θερμοκρασίας,  $T$  (το μέτρο της διαθέσιμης θερμικής ενέργειας), σύμφωνα με τη σχέση  $-E_a/RT$  ( $R$  η σταθερά των αερίων). Οι ενέργειες ενεργοποίησης ορίζουν το εύρος θερμοκρασίας πέραν της οποίας οι διάφορες αντιδράσεις θεωρούνται πιθανές (αν είναι χημικά δυνατές εξ αρχής):

<u>ΜΗΧΑΝΙΣΜΟΣ</u>	<u>ΕΞΑΡΤΗΣΗ ΧΡΟΝΟΥ</u>	<u>ΕΝΕΡΓΕΙΑ ΕΝΕΡΓΟΠΟΙΗΣΗΣ (Kcal/mole)</u>
<i>Desorption</i>	$t^{-1}$ or $t^{-2}$	1 - 10
<i>Diffusion</i>	$t^{-1/2}$	5 - 15
<i>Decomposition</i>	n/a	20 - 80

Επειδή η εκρόφιση αφορά μόνο τα μόρια της επιφάνειας συνήθως συνεισφέρει συγκριτικά ελάχιστα στη συνολική απώλεια μάζας σε τροχιά, αν και έχει χαμηλής θερμοκρασίας εξάρτηση και μικρή χρονική σταθερά. Σημειώνουμε, παρ' όλ' αυτά, ότι η εκρόφιση είναι ο υπεύθυνος μηχανισμός μόλυνσης προερχόμενης από μέταλλα. Ομοίως, η αποσύνθεση συνήθως συνεισφέρει συγκριτικά ελάχιστα στη συνολική απώλεια μάζας λόγω υψηλής θερμοκρασίας εξάρτηση και ανεξαρτησία χρόνου. Η διάχυση/*diffusion*, όμως, έχει μεσαίας θερμοκρασίας εξάρτηση και μεσαίας τιμής χρονική σταθερά. Επειδή η διάχυση/*diffusion* είναι ο υπεύθυνος μηχανισμός της διάχυσης οργανικών υλικών, και περιλαμβάνει τη συνολική μάζα της παρούσας οργανική ύλης, είναι η κύρια πηγή διάχυσης σε τροχιά [1].



### 3. ΑΠΟΣΠΑΣΜΑ ΤΗΣ ΘΕΩΡΙΑΣ ΤΩΝ M.VanEesbeek & A.Zwaal

#### 3.1. ΕΙΣΑΓΩΓΗ

Η διάχυση πολυμερών υπό συνθήκες κενού είναι μια διαδικασία που περιλαμβάνει πολλά διαφορετικά είδη, π.χ. πρόσθετες ουσίες, είδη χαμηλού μοριακού βάρους, προϊόντα αποδόμησης κλπ. Καθένα από αυτά έχει την δική του κινητική αποδόμησης, διάχυσης και εξάτμισης. Οι συναρτήσεις διάχυσης μπορούν συνεπώς να εκφραστούν ως γραμμικός συνδυασμός συγκεκριμένου αριθμού εκθετικών εξισώσεων [4].

Από τα δεδομένα διάχυσης, που συλλέχθηκαν κατά τη διάρκεια μετρήσεων απώλειας μάζας με χρήση ζυγού υπό κενό, δεν δύναται να περιγραφεί η κινητική του φαινομένου διάχυσης για κάθε είδος χωριστά. Ωστόσο, ανεξάρτητα από τον αριθμό των ειδών που εμπλέκονται και του τύπου διάχυσης, ο ρυθμός διάχυσης μπορεί να εκφραστεί ως:

$$\frac{dW_s}{dt} = \sum_i W_{s,i} / \tau_{s,i} \quad eq 1$$

όπου  $W_{s,i}$  είναι η συνολική ποσότητα-μάζα αγνώστων ειδών με μέσο χρόνο παραμονής  $\tau_{s,i}$ .

Η ολοκλήρωση του ρυθμού αυτού οδηγεί στη συνάρτηση της συνολικής απώλειας μάζας- Total Mass Loss (TML):

$$W_{tml} = \sum_i W_{s_0,i} \cdot (1 - \exp[-t / \tau_{s,i}]) \quad eq 2$$

Πρακτικά, αυτό σημαίνει ότι οι TML-συναρτήσεις περιγράφονται ως το άθροισμα εκθετικών εξισώσεων [4].

Όσο περισσότερες είναι οι εξισώσεις, τόσο καλύτερη είναι η μαθηματική προσαρμογή των πειραματικών καμπυλών.

Η προσαρμογή είναι έγκυρη μόνο για ισόθερμες TML-καμπύλες και η λογική εξαγωγή σε μεγαλύτερους χρόνους περιορίζεται σε επαναλήψεις της χρονικής διάρκειας του τεστ.

Παρ' όλα αυτά, από μετρήσεις σε υψηλότερες θερμοκρασίες, μπορούν να εξαχθούν οι παράγοντες επιτάχυνσης των  $\tau$ 's ως συνάρτηση της θερμοκρασίας. Αυτοί οι παράγοντες χρησιμοποιούνται για να συναχθεί η καμπύλη σε χαμηλότερη θερμοκρασία.

Το μοντέλο χρόνου παραμονής μπορεί να χρησιμοποιηθεί για φαινόμενα προσρόφησης ή συμπύκνωσης στις επιφάνειες.

Οι επιφάνειες μπορούν να μολυνθούν. Ο βαθμός καθορίζεται από την ταχύτητα του συστήματος και τον χρόνο παραμονής σε αυτές [1].

Αυτός ο βαθμός προκύπτει από τις εξισώσεις eq 3a ή 3b:

$$\frac{dW_w}{dt} = \sum_i \left[ (1 - \alpha_p) \cdot \frac{dW_{s,i}}{dt} - \alpha_p \cdot \frac{dW_{e,i}}{dt} \right] \quad \text{eq 3a}$$

όπου  $W_w$  : η ποσότητα των προϊόντων διάχυσης στους τοίχους του θαλάμου

$W_{s,i}$  : η ποσότητα δυνητικών προϊόντων διάχυσης στην πηγή

$\frac{dW_{e,i}}{dt}$  : ο ρυθμός απορρόφησης από την επιφάνεια =  $W_w / \tau_w$

$\alpha_p$  : κριτικός παράγων του ανοίγματος της αντλίας

Η εξίσωση eq 3a μπορεί επίσης να γραφεί ως:

$$\frac{dW_w}{dt} = \sum_i \left[ (1 - \alpha_p) \cdot \frac{W_{s,i}}{\tau_{s,i}} - \alpha_p \cdot \frac{W_{w,i}}{\tau_{w,i}} \right] \quad \text{eq 3b}$$

όπου  $\tau_{s,i}$  : ο χρόνος παραμονής του είδους στην πηγή

$\tau_{w,i}$  : ο χρόνος παραμονής στον τοίχο

Από επιταχυνόμενα τεστ προσδιορίζεται η σχέση μεταξύ χρόνων παραμονής και θερμοκρασίας.

Αυτό το μοντέλο μόλυνσης επιτρέπει προβλέψεις διάχυσης σε διαφορετικές θερμοκρασίες για μεγάλους χρόνους (πολλά χρόνια).

Η επίδραση της διάχυσης των υλικών μπορεί επίσης να προβλεφθεί.

## 3.2. ΠΡΟΣΕΓΓΙΣΗ ΚΑΜΠΥΛΩΝ ΔΙΑΧΥΣΗΣ ΚΑΙ ΣΥΜΠΥΚΝΩΣΗΣ

### 3.2.1. ΚΑΜΠΥΛΗ ΔΙΑΧΥΣΗΣ

Η θερμοκρασία του υλικού αυξάνεται από τους 25 °C στους 125 °C με βήμα 25 °C για περιόδους 24 ωρών.

Η θερμοκρασία του συλλέκτη διατηρείται σταθερή στους -25 °C.

Τα «βήματα θερμοκρασίας» επιλέγονται ώστε να επιταχυνθεί το τεστ διάχυσης σε τέτοιο βαθμό ώστε να δύναται προσομοίωση πολλών ετών στους 25 °C (διαστημικές αποστολές).

Για να αποκτήσουμε (από τις μετρήσεις) τις μαθηματικές εξισώσεις διάχυσης, απαιτείται μαθηματική προσαρμογή των καμπυλών και προσδιορισμός των παραγόντων επιτάχυνσης.

### 3.2.1.1. ΜΑΘΗΜΑΤΙΚΗ ΠΡΟΣΑΡΜΟΓΗ ΔΕΔΟΜΕΝΩΝ

Τα δεδομένα της καμπύλης διάχυσης προσεγγίζονται με χρήση της μεθόδου ελαχίστων τετραγώνων από ένα άθροισμα τυπικά 6 εκθετικών συναρτήσεων:

$$W_{iml} = \sum_i A_i \cdot (1 - \exp[-t / B_i]) \quad eq 4$$

όπου  $B_i$  : εκθετικοί παράγοντες, μεταξύ των τιμών 0.1 και 50 ώστε να καλυφθεί όλο το χρονικό μεσοδιάστημα (π.χ. 0.1, 0.5, 1, 5, 20, 50)

$A_i$  : γραμμικοί/προεκθετικοί παράγοντες. Αυτοί υπολογίζονται ώστε να ελαχιστοποιείται η διαφορά ανάμεσα στις πραγματικές και τις εκτιμηθείσες τιμές σε όλο το διάστημα του χρόνου

Αυτός ο τύπος είναι ίδιος με αυτόν της eq 2, αλλά οι παράγοντες  $A_i$  και  $B_i$  δεν έχουν φυσικό νόημα.

Θα ήταν δυνατή η προσέγγιση των καμπυλών με χρήση ν-βαθμού πολωνύμου ή λογαριθμικής συνάρτησης, αλλά αυτό οδηγεί σε ανυπέβλητα προβλήματα κατά τον υπολογισμό των παραγόντων επιτάχυνσης [4].

### 3.2.1.2. ΕΠΙΔΡΑΣΗ ΤΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΣΤΗ ΔΙΑΧΥΣΗ

Οι "μαθηματικά προσαρμοσμένες" καμπύλες διαφορίζονται για να εξαχθούν οι ρυθμοί σε διαφορετικές θερμοκρασίες. Από αυτούς καθορίζονται οι παράγοντες επιτάχυνσης, που δίνονται ως:

$$K_a = \left[ \frac{dW}{dt} \right]_{T_j} / \left[ \frac{dW}{dt} \right]_{T_i} \quad T_i \rightarrow T_j \quad eq 5$$

Συνεπώς, τα τεστ πρέπει να διενεργούνται σε διαφορετικές θερμοκρασίες. Σ' αυτή την περίπτωση τα τεστ γίνονται σε 5 διαφορετικές θερμοκρασίες, και προκύπτουν έτσι 4 παράγοντες επιτάχυνσης από τους οποίους μπορεί να προσδιοριστεί η σχέση μεταξύ χρόνου παραμονής και θερμοκρασίας, πάλι με χρήση ελαχίστων τετραγώνων.

Για κάθε υλικό, εμπειρικές εξισώσεις της μορφής:

$$\tau_{T_2} = \tau_{T_1} \cdot a \cdot \exp(-b \cdot T_2) \quad eq 6$$

δίνουν τον χρόνο παραμονής σε κάθε θερμοκρασία.

### 3.2.1.3. ΠΡΟΣΟΜΟΙΩΣΗ ΚΑΜΠΥΛΗΣ ΔΙΑΧΥΣΗΣ

Η συνάρτηση χρόνου παραμονής-θερμοκρασίας χρησιμοποιείται για την μετατροπή όλων των δεδομένων του πραγματικού δυναμικού τεστ σε μια θερμοκρασία (π.χ. 25 °C). Από αυτά τα δεδομένα, υπολογίζουμε τη συνάρτηση διάχυσης του τύπου:

$$W_{tml} = \sum_i W_{0,i} \cdot (1 - \exp[-t / \tau_{s,i}]) \quad eq 7$$

Εδώ  $W_{0,i}$  είναι η συνολική ποσότητα κάθε είδους διάχυσης με χρόνο παραμονής  $\tau_{s,i}$ , που βρισκόταν αρχικά στο δείγμα.

Στην πραγματικότητα, η  $\tau_{s,i}$  μπορεί να είναι η μέση τιμή των χρόνων παραμονής διαφορετικών ειδών [2].

Υπό συγκεκριμένους περιορισμούς (μη αρνητικές τιμές για τα  $\tau$  και  $W$ ), μπορούμε να υπολογίσουμε τις τιμές  $W_{0,i}$  και  $\tau_{s,i}$  για ένα δεδομένο σύνολο εξισώσεων.

Στην περίπτωσή μας, έχουμε επιλέξει τις  $\tau_{s,i}$  τιμές για να απλοποιήσουμε τον υπολογισμό, γιατί καθώς φαίνεται η σχέση μεταξύ  $\tau_{s,i}$  και  $W_{0,i}$  είναι συνεχής συνάρτηση.

Οι τελικές εξισώσεις εκφράζουν μερικώς την μετρημένη διάχυση και μερικώς προβλέψεις βασισμένες σε μετρημένη διάχυση σε άλλες θερμοκρασίες, που μετατράπηκαν σύμφωνα με τις συναρτήσεις χρόνου παραμονής-θερμοκρασίας.

### 3.2.2. ΚΑΜΠΥΛΗ ΣΥΜΠΥΚΝΩΣΗΣ ΣΤΟΝ ΣΥΛΛΕΚΤΗ (QCM)

Μια παρόμοια αντιμετώπιση όπως στο 3.2.1. ακολουθείται για τις καμπύλες συμπύκνωσης στον συλλέκτη και οι προκύπτουσες εξισώσεις υποδεικνύουν την ποσότητα των προϊόντων διάχυσης που συλλέχθηκαν στους  $-25^\circ\text{C}$  στη διάρκεια του πραγματικού τεστ:

$$W_{cvcm} = \sum_i W_{c,i} \cdot (1 - \exp[-t / \tau_{s,i}]) \quad eq 8$$

όπου  $W_{c,i}$  είναι η ποσότητα των "συλλεχθέντων" ειδών του υλικού.

## 3.3. ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΩΝ ΤΕΣΤ

### ΔΙΑΧΥΣΗ ΚΑΙ ΣΥΜΠΥΚΝΩΣΗ ΣΤΟΝ ΣΥΛΛΕΚΤΗ [4]

Εξίσωση Διάχυσης:

$$\%TML = \sum_i W_i \cdot (1 - \exp(-t / \tau_i))$$

Εξίσωση Συμπύκνωσης στον Συλλέκτη:

$$\%CVCM = \sum_i W_{c,i} \cdot (1 - \exp(-t / \tau_i))$$

## 4. Η ΜΕΘΟΔΟΣ ΤΟΥ ESTEC

Όλα τα υλικά περιέχουν κλασματικές ποσότητες πτητικών χημικών στοιχείων, είτε στην επιφάνεια ή στο εσωτερικό τους. Κατά την έκθεση σε κενό, ιδιαίτερα σε υψηλές θερμοκρασίες, αυτά τα στοιχεία με την πάροδο του χρόνου ρέουν προς τις επιφάνειες μέσω διάχυσης/diffusion, και αποδεσμεύονται από αυτές μέσω του μηχανισμού της εκρόφησης. Σε ακραίες θερμοκρασίες τείνουν ακόμη και να αποσυντίθενται σε απλούστερα στοιχεία. Ο συνδυασμός αυτών των διαδικασιών καλείται διάχυση [1].

Δυναμικά τεστ διάχυσης δίνουν τη δυνατότητα πλήρους χαρακτηρισμού του φαινομένου ως συνάρτηση του χρόνου. Ωστόσο, είναι αδύνατη η διεξαγωγή δοκιμών για την πλήρη χρονική διάρκεια μιας αποστολής, γι αυτό χρησιμοποιούνται διάφορες μέθοδοι για την προέκταση των αποτελεσμάτων από σχετικά μικρής διάρκειας τεστ σε όλη τη διάρκεια της αποστολής.

Μια απλή μέθοδος είναι η διεξαγωγή ενός δυναμικού τεστ με διάρκεια της τάξης μιας εβδομάδας και η προέκταση του ρυθμού διάχυσης, γραμμικά προσαρμοσμένου στο τέλος του τεστ, σε μεγαλύτερες περιόδους. Μια άλλη μέθοδος περιλαμβάνει τη χρήση κάποιας συνάρτησης προσαρμογής, συνήθως λογαριθμικής, και τη χρήση της ίδιας συνάρτησης για μακροπρόθεσμες προβλέψεις. Αυτές οι μέθοδοι σχετίζονται με τεστ στα οποία διατηρείται η θερμοκρασία σταθερή και στην πραγματικότητα εστιάζουν στην κατανόηση των διαφορών μηχανισμών διάχυσης.

Άλλες τεχνικές, όπως αυτή του ESTEC, επικεντρώνονται στην κατανόηση της κινητικής του φαινομένου, στοχεύοντας στην εκτίμηση της επίδρασης της θερμοκρασίας και την πρόβλεψη με χρήση των παραγόντων επιτάχυνσης.

### 4.1. ΟΙ ΒΑΣΙΚΕΣ ΕΞΙΣΩΣΕΙΣ

Η διάχυση συχνά περιγράφεται ως μια αντίδραση πρώτης τάξης, δηλαδή το υλικό παράγει αέρια σε ρυθμό ανάλογο της μάζας του, κάνοντας χρήση του νόμου του Arrhenius [2]. Ο ρυθμός διάχυσης μπορεί να περιγραφεί από τη σχέση:

$$\frac{dW}{dt} = \frac{W}{\tau} \quad (1)$$

όπου  $W$  είναι η ποσότητα μάζας που συνεισφέρει στη διάχυση  
 $\tau$  η σταθερά χρόνου παραμονής του φαινομένου διάχυσης, για την οποία ισχύει ο νόμος Arrhenius:

$$\tau = \tau_0 \cdot e^{\frac{E_a}{RT}} \quad (2)$$

όπου  $\tau_0$  η περίοδος ταλάντωσης του μορίου που βρίσκεται στην επιφάνεια του υλικού στην θερμοκρασία αναφοράς  $T_0$

$E_a$  ενέργεια ενεργοποίησης

$R$  η σταθερά των αερίων

$T$  θερμοκρασία σε K

Η ολοκλήρωση της (1) δίνει:

$$W = W_0 \cdot \exp^{-t/\tau} \quad (3)$$

όπου  $W_0$  η αρχική μάζα που συνεισφέρει στη διάχυση

Η απώλεια μάζας μπορεί συνεπώς να εκφραστεί ως:

$$W_{loss} = W - W_0 = W_0 \cdot (1 - \exp^{-t/\tau}) \quad (4)$$

Στην πραγματικότητα, το φαινόμενο διάχυσης είναι μια διαδικασία που περιλαμβάνει πολλά διαφορετικά είδη μορίων. Καθένα από αυτά έχει τη δική του συγκεκριμένη κινητική και, αν υποθέσουμε ότι όλα εξελίσσονται σύμφωνα με μια πρώτης τάξης αντίδραση, τότε η (1) μπορεί να αναγραφεί για το είδος  $i$  ως:

$$\left( \frac{dW}{dt} \right)_i = \frac{W_i}{\tau_i} \quad (5)$$

δίνοντας συνολική απώλεια μάζας ίση με:

$$W_{loss} = \sum_i W_{0,i} \cdot (1 - \exp^{-t/\tau_i}) \quad (6)$$

Πρακτικά αυτό σημαίνει ότι οι συναρτήσεις συνολικής απώλειας μάζας ( $W_{loss} = TML$ ), όπως προκύπτουν από συνεχείς μετρήσεις, μπορούν να περιγραφούν ως άθροισμα εκθετικών συναρτήσεων.

Σε αυτή τη βάση, πρέπει να σημειωθεί ότι, εφόσον είναι αδύνατο να περιγραφούν οι παράμετροι κινητικής κάθε είδους χωριστά, η συνολική διαδικασία απλοποιείται θεωρώντας οικογένειες μορίων με μέσο χρόνο παραμονής  $\tau_i$  [4].

## 4.2. ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΜΕΘΟΔΟΥ

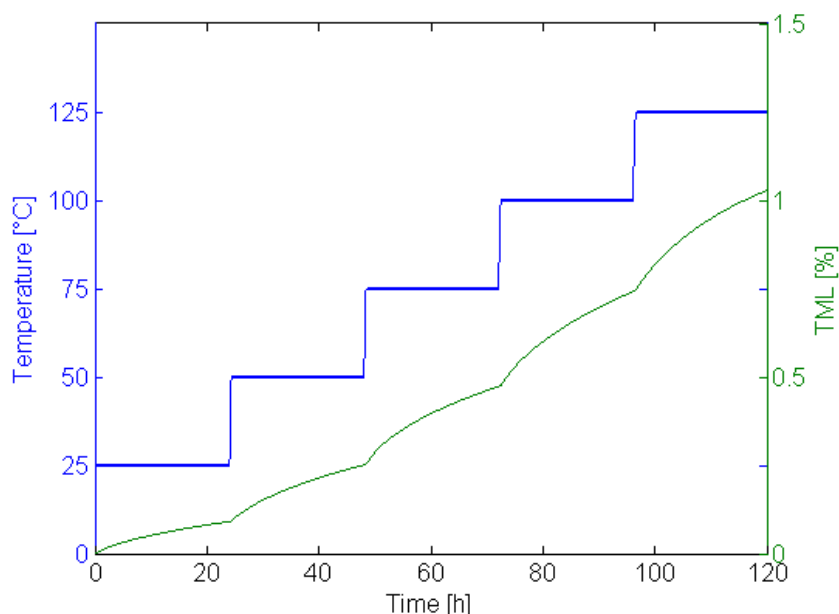
Η εξίσωση (6) ισχύει μόνο για ισόθερμες TML καμπύλες και, ανεξάρτητα από την ποιότητα της προσαρμογής, σε ένα τεστ διεξαχθέν σε σταθερή θερμοκρασία μια λογική προέκταση στον χρόνο των αποτελεσμάτων θα περιοριζόταν σε διάρκεια κάποιες φορές πολλαπλάσια της πραγματικής διάρκειας του τεστ. Αντίθετα, στην δυναμική προσέγγιση με ισόθερμες μετρήσεις σε διαφορετικές θερμοκρασίες, μπορούν να εξαχθούν οι παράγοντες επιτάχυνσης ως συνάρτηση των αυξήσεων της θερμοκρασίας και να χρησιμοποιηθούν για να προεκτείνουν τη διάρκεια του τεστ σε χαμηλότερες θερμοκρασίες [4].

Η δυναμική μέθοδος λοιπόν χρησιμοποιείται στο ESTEC για να επιταχυνθεί το φαινόμενο διάχυσης και να υπολογιστούν οι απαραίτητες παράμετροι για την προτυποποίησή του και την διεξαγωγή μακροπρόθεσμων προβλέψεων. Ακριβέστερα, η μέθοδος είναι ένα «πολυθερμοκρασιακό» τεστ στη διάρκεια του οποίου η θερμοκρασία του δείγματος αυξάνεται από τους 25 °C στους 125 °C με βήμα 25 °C κάθε 24 ώρες. Έτσι καταγράφεται συνεχώς η TML του δείγματος όπως υπολογίζεται από τον ζυγό της διάταξης. Επίσης καταγράφονται από μικροζυγούς κρυστάλλου



χαλαζία (quartz crystal microbalances - QCMs), που ψύχονται και διατηρούνται σε θερμοκρασίες -25 °C, -50 °C και -75 °C, οι συχνότητες ταλάντωσής τους καθώς συλλέγουν κάποια από τα μόρια που έχουν διαχυθεί.

Τυπικές καμπύλες θερμοκρασίας – TML φαίνονται στην [εικόνα 4-1], με την επιτάχυνση της διάχυσης που αντιστοιχεί στην αύξηση θερμοκρασίας να είναι προφανής.



εικόνα 4- 1

Στην δυναμική προσέγγιση η επίδραση της θερμοκρασίας στη διάχυση μπορεί να περιγραφεί συνδυάζοντας τις εξισώσεις (1) και (2):

$$\frac{dW}{dt} = \frac{W}{\tau_0} e^{-\frac{E_a}{RT}} \quad (7)$$

Γράφοντας την εξίσωση (7) για δυο θερμοκρασίες και έπειτα θεωρώντας τον λόγο τους, μπορούμε να ορίσουμε έναν παράγοντα επιτάχυνσης ως:

$$k_\alpha = \frac{\left(\frac{dW}{dt}\right)_{T_2}}{\left(\frac{dW}{dt}\right)_{T_1}} = e^{\frac{E_a}{R} \left(\frac{T_2 - T_1}{T_1 \cdot T_2}\right)} \quad (8)$$

και να εκφράσουμε την ενέργεια ενεργοποίησης ως:

$$E_\alpha = R \cdot \left(\frac{T_1 \cdot T_2}{T_2 - T_1}\right) \cdot \ln k_\alpha \quad (9)$$

Ωστόσο, η εξίσωση (8) είναι έγκυρη για ένα είδος μορίου. Στην πιο ρεαλιστική περίπτωση που υπάρχουν περισσότερα του ενός είδη, οι παράγοντες επιτάχυνσης υπολογίζονται ως:

$$k_{\alpha} = \frac{\left(\frac{dW}{dt}\right)_{T_2}}{\left(\frac{dW}{dt}\right)_{T_1}} = \frac{\sum_i \left(\frac{W_i}{\tau_i}\right)_{T_2}}{\sum_i \left(\frac{W_i}{\tau_i}\right)_{T_1}} = \frac{\sum_i \frac{W_i}{\tau_{0,i}} \cdot e^{\frac{E_{\alpha,i}}{R \cdot T_2}}}{\sum_i \frac{W_i}{\tau_{0,i}} \cdot e^{\frac{E_{\alpha,i}}{R \cdot T_1}}} \quad (10)$$

Η ακριβής επίλυση της (10) απαιτεί γνώση των  $W_i$ ,  $\tau_{0,i}$  και  $E_{\alpha,i}$  για κάθε είδος, αλλά τέτοιες πληροφορίες δεν προκύπτουν από το τεστ. Η μέθοδος του ESTEC θεωρεί έγκυρη την εξίσωση (8) και εκτιμά την ενέργεια ενεργοποίησης για κάθε αύξηση θερμοκρασίας ενός μοναδικού ‘μέσου’ είδους [4].

### 4.3. ΠΡΟΒΛΕΨΕΙΣ

Για κάθε ισόθερμη καμπύλη  $j$ , μπορεί να επιταχυνθεί ο χρόνος με χρήση των αντίστοιχων παραγόντων επιτάχυνσης ως;

$$t_{acc,T_j} = t_{T_j} + \sum_{k=j+1}^n \left( t_{T_k} \cdot \prod_{m=j+1}^k (k_{\alpha})_{T_m, T_{m-1}} \right) \quad (11)$$

- όπου  $t_{acc,T_j}$  ο επιταχυνθείς χρόνος σε θερμοκρασία  $T_j$   
 $t_{T_j}$  ο πραγματικός χρόνος σε θερμοκρασία  $T_j$   
 $t_{T_k}$  ο πραγματικός χρόνος σε θερμοκρασίες  $T_k < T_j$   
 $n$  ο συνολικός αριθμός των βημάτων θερμοκρασίας

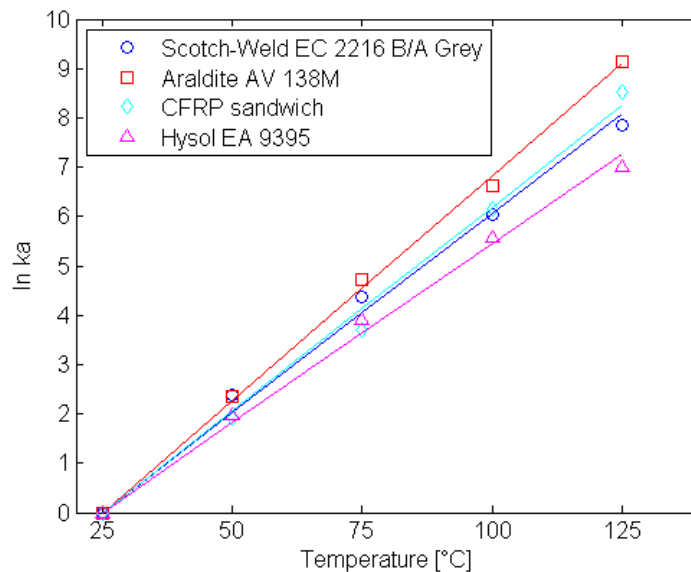
Θεωρώντας θερμοκρασία αναφοράς,  $T_{ref}$ , τη χαμηλότερη θερμοκρασία του τεστ, μπορούμε να παράγουμε την επιταχυνθείσα ισόθερμη TML-συνάρτηση συνδυάζοντας τα πραγματικά δεδομένα με τον επιταχυνθέντα χρόνο για θερμοκρασία  $T_{ref}$ .

Μπορούμε να χρησιμοποιήσουμε την εξίσωση (6) για να προσαρμόσουμε την επιταχυνθείσα TML ως το άθροισμα εκθετικών συναρτήσεων, και να υπολογίσουμε τα νέα διανύσματα  $W_{0,i}$  και  $\tau_i$  για τις προβλέψεις στην θερμοκρασία αναφοράς.

Μακροπρόθεσμες προβλέψεις για άλλες θερμοκρασίες είναι εφικτές και βασίζονται στον εμπειρικό νόμο που προκύπτει με παρατήρηση των διαγραμμάτων του νεπέριου λογαρίθμου του  $k_{\alpha}$  ως συνάρτηση της θερμοκρασίας. Μάλιστα, σε αυτά τα διαγράμματα τα πειραματικά δεδομένα μπορούν γραμμικά να προσεγγιστούν [εικόνα 4-2] σύμφωνα με τη σχέση;

$$\ln k_{\alpha} = k_e \cdot (T - T_{ref}) \quad (12)$$

όπου  $k_\alpha$  είναι ο συντελεστής εξάρτησης θερμοκρασίας-χρόνου παραμονής, που χρησιμοποιείται για την εξαγωγή των χρονικών σταθερών  $\tau_T$  σε μια γενική θερμοκρασία  $T$  από εκείνες στην θερμοκρασία αναφοράς.



εικόνα 4- 2

Η εξίσωση (8) μπορεί να αναγραφεί ως:

$$k_\alpha = \frac{\left(\frac{dW}{dt}\right)_T}{\left(\frac{dW}{dt}\right)_{T_{ref}}} = \frac{\frac{W}{\tau_T}}{\frac{W}{\tau_{T_{ref}}}} = \frac{\tau_{ref}}{\tau_T} \quad (13)$$

και σε συνδυασμό με τις εξισώσεις (11) και (12) προκύπτει ότι:

$$\tau_T = \tau_{T_{ref}} \cdot e^{-k_e \cdot (T - T_{ref})} \quad (14)$$

#### 4.4. Η ΔΙΑΤΑΞΗ VBQC

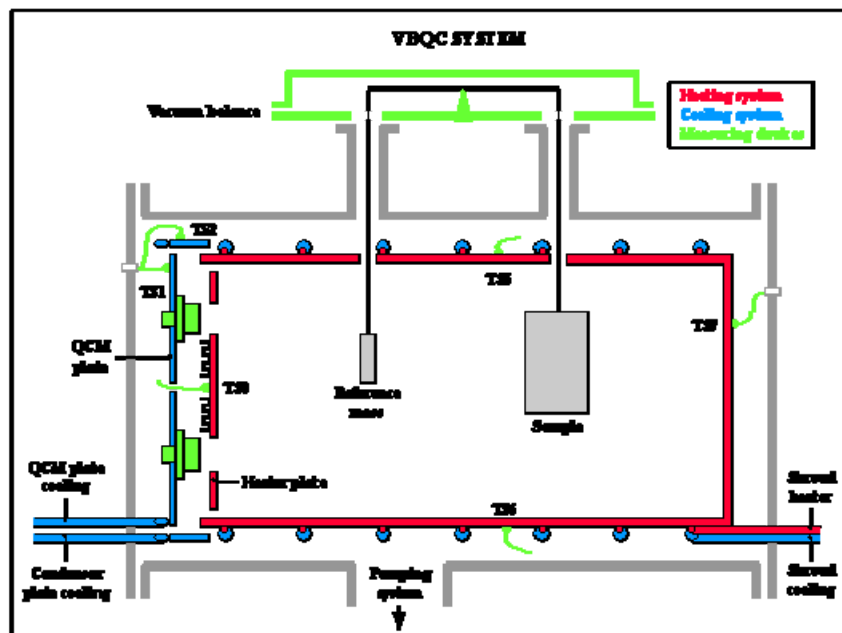
Η διάταξη που έχουν αναπτύξει οι μηχανικοί του ΕΟΔ για την διεξαγωγή δυναμικών πειραμάτων διάχυσης ονομάζεται VBQC (Vacuum Balance Quartz Crystal – Ζυγός Κρυστάλλου Χαλαζία υπό συνθήκες κενού) [εικόνα 4-4]. Η διάταξη αυτή αποτελείται από συστήματα θέρμανσης και ψύξης και από συσκευές μετρήσεων [εικόνα 4-3].

Το σύστημα θέρμανσης χρησιμοποιείται για την αύξηση της θερμοκρασίας του δειγματικού υλικού μέσω ακτινοβολίας. Αποτελείται από δυο μέρη, τον κύριο θάλαμο και την πλάκα θέρμανσης, με την θερμοκρασία και των δυο να ελέγχεται και να καταγράφεται. Το σύστημα ψύξης αποτελείται από την πλάκα των μικροκρυστάλλων χαλαζία, την πλάκα του συμπυκνωτή και το ψυκτικό περίβλημα του θαλάμου που επιτρέπει την έναρξη του πειράματος σε θερμοκρασία χαμηλότερη της θερμοκρασίας δωματίου, όποτε χρειάζεται. Η ψύξη πραγματοποιείται με χρήση

κρυστάτη που περιέχει μεθανόλη και σταθεροποιεί την θερμοκρασία των τριών QCMs στους  $-25^{\circ}\text{C}$ ,  $-50^{\circ}\text{C}$  και  $-75^{\circ}\text{C}$  [5].

Σημείωση: Η μέθοδος του ESTEC βασίζεται στην θεωρία των M.VanEesbeek και A.Zwaal, στην οποία προτείνεται και η χρήση ενός μικροζυγού κρυστάλλου χαλαζία. Οι μηχανικοί όμως ενσωμάτωσαν στη διάταξη τρεις μικροζυγούς ως αναφέρεται άνωθεν, και καταγράφονται μετρήσεις για καθέναν εξ αυτών.

Η πλάκα του συμπυκνωτή χρησιμοποιείται κατά τη λειτουργία καθαρισμού του συστήματος, και συγκεντρώνει το εναπομείναν υλικό καθώς ο θάλαμος θερμαίνεται στους  $150^{\circ}\text{C}$  για περίπου 44 ώρες. Τέλος, σημειώνεται ότι με την διάταξη VBQC επιτυγχάνονται πιέσεις της τάξης των  $10^{-6}\text{ mbar}$  [5].



εικόνα 4- 3



εικόνα 4- 4

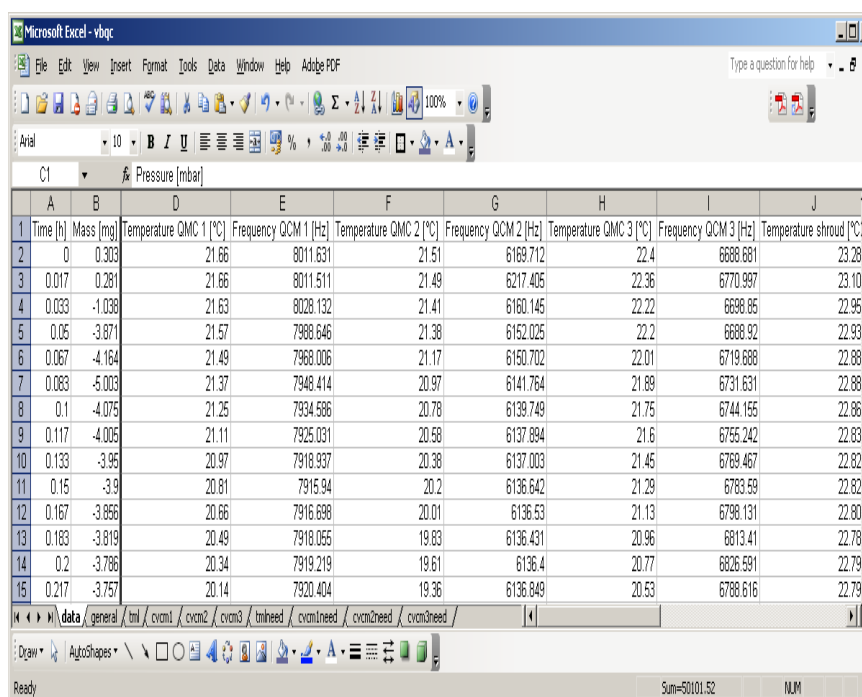
## 5. ΜΑΘΗΜΑΤΙΚΟΣ ΧΕΙΡΙΣΜΟΣ

Στο παρελθόν η μαθηματική επεξεργασία των δεδομένων των δυναμικών τεστ διάχυσης γινόταν μέσω διαφόρων αρχείων Excel®. Η εκτέλεση όλης της ακολουθίας απαιτούσε χρόνο, ανθρώπινη παρουσία και υποκειμενικές αποφάσεις. Με στόχο τον περιορισμό του κινδύνου της υποκειμενικότητας διατηρώντας όμως τον έλεγχο της συνολικής διαδικασίας, αναπτύχθηκε μια εφαρμογή στο MatLab® τα βασικά στάδια επεξεργασίας της οποίας παρουσιάζονται στην ενότητα αυτή.

### 5.1. ΤΑ ΔΕΔΟΜΕΝΑ

Η φόρτωση των δεδομένων στον υπολογιστή προς επεξεργασία γίνεται αυτόματα από τον υπολογιστή της διάταξης νβqc με τη λήξη του πειράματος. Συγκεκριμένα τα δεδομένα που χρησιμοποιούνται στην ανάλυση είναι [εικόνα 5-1]:

- ο χρόνος σε h
- η μάζα του δείγματος σε mg
- η θερμοκρασία των QCMs σε °C για τα 3 QCMs
- η συχνότητα των QCMs σε Hz για τα 3 QCMs
- η θερμοκρασία του καλύμματος/θαλάμου σε °C



Time [h]	Mass [mg]	Temperature QCM 1 [°C]	Frequency QCM 1 [Hz]	Temperature QCM 2 [°C]	Frequency QCM 2 [Hz]	Temperature QCM 3 [°C]	Frequency QCM 3 [Hz]	Temperature shroud [°C]
0	0.303	21.66	8011.631	21.51	6169.712	22.4	6688.681	23.28
0.017	0.281	21.66	8011.511	21.49	6217.405	22.36	6770.997	23.10
0.033	-1.038	21.63	8028.132	21.41	6160.145	22.22	6698.85	22.95
0.05	-3.871	21.57	7988.646	21.38	6152.025	22.2	6688.92	22.93
0.067	-4.164	21.49	7968.006	21.17	6150.702	22.01	6719.688	22.88
0.083	-5.003	21.37	7948.414	20.97	6141.764	21.89	6731.631	22.88
0.1	-4.075	21.25	7934.586	20.78	6139.749	21.75	6744.155	22.86
0.117	-4.005	21.11	7925.031	20.58	6137.894	21.6	6755.242	22.83
0.133	-3.95	20.97	7918.937	20.38	6137.003	21.45	6769.467	22.82
0.15	-3.9	20.81	7915.94	20.2	6136.642	21.29	6783.59	22.82
0.167	-3.856	20.66	7916.698	20.01	6136.53	21.13	6798.131	22.80
0.183	-3.819	20.49	7918.055	19.83	6136.431	20.96	6813.41	22.78
0.2	-3.786	20.34	7919.219	19.61	6136.4	20.77	6826.591	22.79
0.217	-3.757	20.14	7920.404	19.36	6136.849	20.53	6788.616	22.79

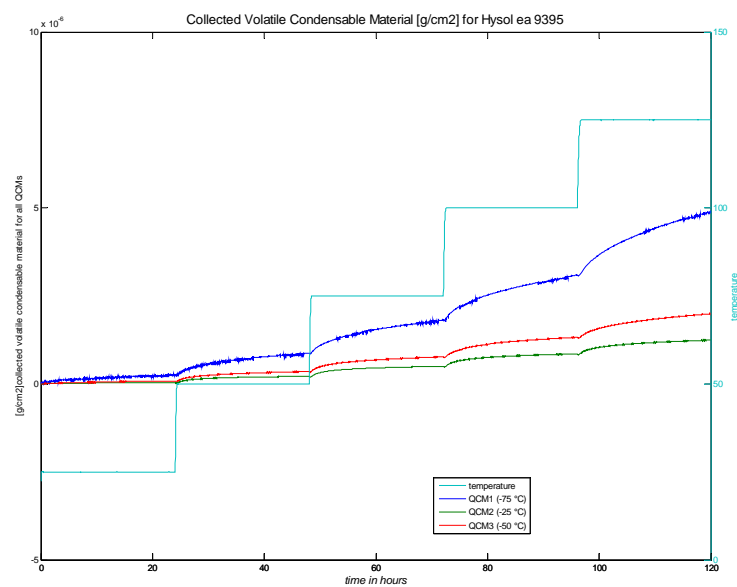
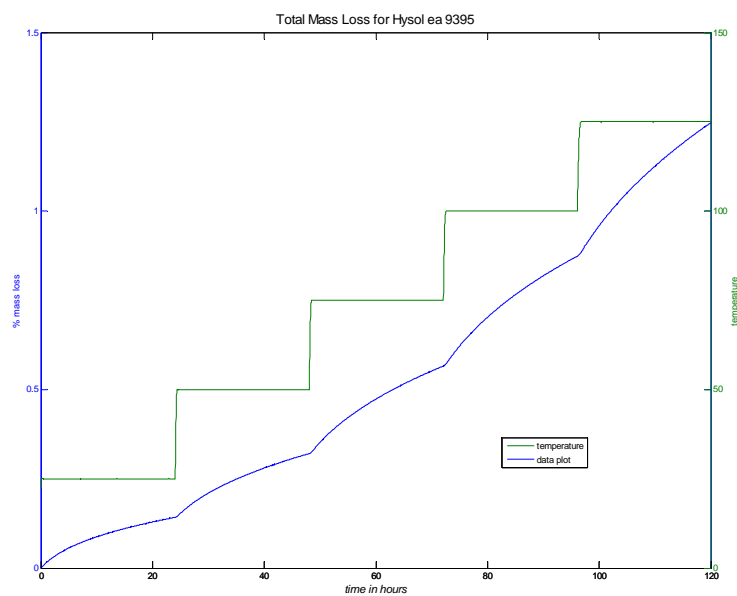
εικόνα 5- 1

Τα δεδομένα αυτά χωρίζονται με βάση τον χρόνο σε ισόθερμες φάσεις από το MatLab® αμέσως μετά τον υπολογισμό της συνολικής απώλειας μάζας (TML) ως ποσοστό της αρχικής μάζας του δείγματος και των ποσοτήτων μάζας που συγκεντρώθηκαν στους τρεις συλλέκτες (CVCMS) ως [6]:

$$\frac{Mass - Mass(1)}{SampleMass} \cdot 100\% \quad \text{και} \quad (frequencyQCM_i - frequencyQCM_i(1)) \cdot sens \cdot 100\%$$

όπου Mass	είναι η μάζα του δείγματος όπως μετρήθηκε από τον ζυγό σε συνθήκες κενού, είναι δηλαδή η διαφορά μάζας ανάμεσα στη μάζα του δείγματος και του αντίβαρου στον άλλο βραχίονα
SampleMass	η αρχική μάζα του εξεταζόμενου δείγματος
frequencyQCM <sub>i</sub>	η συχνότητα του <i>i</i> , <i>i</i> = 1, 2, 3, Μικροζυγού Κρυστάλλου Χαλαζία (Quartz Crystal Microbalance)
και sens	η ευαισθησία του QCM <sub>i</sub> ( $= 1.96 \cdot 10^{-9} \text{ g/Hz}$ ) [6]

Θεωρώντας τα γραφήματα των άνωθεν υπολογισθέντων ποσοτήτων σε συνάρτηση με τον χρόνο τα αποτελέσματα θα είναι τα ακόλουθα [εικόνες 5-2, 5-3]:



εικόνες 5- 2, 5-3

Σύμφωνα με τη θεωρία που εφαρμόζουμε, αυτά τα δεδομένα μπορούν να προσαρμοστούν από τα παρακάτω μαθηματικά μοντέλα – για κάθε ισόθερμη περίοδο:

$$TML = \sum_{i=1}^m w_i \cdot (1 - e^{-t/\tau_i})$$

$$CVCM_k = \sum_{i=1}^m w_i \cdot (1 - e^{-t/\tau_i}) \quad , k=1, 2, 3$$

όπου  $w_i \geq 0, i = 1, \dots, m$   
 $\tau_i > 0, i = 1, \dots, m, \text{ δοσμένα}$

## 5.2. ΓΕΝΙΚΗ ΠΡΟΣΑΡΜΟΓΗ ΔΕΔΟΜΕΝΩΝ ΜΕ ΕΛΑΧΙΣΤΑ ΤΕΤΡΑΓΩΝΑ

Δίνονται ζεύγη δεδομένων  $(t_j, y_j)$ ,  $j = 1, 2, \dots, n$   
 όπου  $y_j = y(t_j)$ ,  $y$ : άγνωστη συνάρτηση

Έστω ότι η  $y$  θα προσαρμοστεί από έναν γραμμικό συνδυασμό γνωστών συναρτήσεων  $\Phi_i$ ,  $i = 1, 2, \dots, m$   
 Τότε,

$$y(t) = \sum_{i=1}^m w_i * \Phi_i(t) \quad (\text{το μαθηματικό μοντέλο})$$

Το πρόβλημα είναι η επιλογή κατάλληλων τιμών για τις μεταβλητές  $w_i$ ,  $i=1, 2, \dots, m$ .

ΣΗΜΕΙΩΣΕΙΣ: -Το μοντέλο είναι γραμμικό, παρόλο που οι  $\Phi_i$ ,  $i = 1, 2, \dots, m$ ,  
 μπορεί να μην είναι γραμμικές  
 -  $n \geq m$  (συνθήκη)

Το σφάλμα του  $j$  – σημείου είναι

$$e_j = \sum_{i=1}^m w_i * \Phi_i(t_j) - y_j \quad , j = 1, 2, \dots, n$$

Σύμφωνα με το κριτήριο των ελαχίστων τετραγώνων, οι τιμές  $w_i$ ,  $i = 1, 2, \dots, m$ , επιλέγονται έτσι ώστε το σφάλμα

$$e = \sqrt{\sum_{j=1}^n e_j^2} = \sqrt{\sum_{j=1}^n \left( \sum_{i=1}^m w_i * \Phi_i(t_j) - y_j \right)^2} \quad \text{ελαχιστοποιείται.}$$

Συνεπώς,  $e$  ελάχιστο ( $e \geq 0$ )  $\Leftrightarrow e^2$  ελάχιστο. Έτσι λοιπόν, για  $i = 1, 2, \dots, m$ , η επόμενη συνθήκη πρέπει να ικανοποιείται ή μπορούμε να πούμε ότι ο στόχος είναι να ευρεθούν τα  $w_i$ ,  $i = 1, 2, \dots, m$ , που ικανοποιούν την ακόλουθη συνθήκη:

$$\frac{\partial e^2}{\partial w_i} = 0$$

Για  $i=k$ ,

$$\frac{\partial e^2}{\partial w_k} = 0$$

$$\Leftrightarrow \left[ \sum_{j=1}^n \left( \sum_{i=1}^m w_i * \Phi_i(t_j) - y_j \right)^2 \right]' (w_k) = 0$$

$$\Leftrightarrow 2 * \sum_{j=1}^n \left( \sum_{i=1}^m w_i * \Phi_i(t_j) - y_j \right) * \Phi_k(t_j) = 0$$

$$\Leftrightarrow \sum_{i=1}^m \sum_{j=1}^n \Phi_i(t_j) * \Phi_k(t_j) * w_i = \sum_{j=1}^n y_j * \Phi_k(t_j)$$

$$\Leftrightarrow \mathbf{A} \circ \mathbf{w} = \mathbf{b}$$

όπου

$$\mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}_{m \times 1}, \quad \mathbf{b} = \begin{bmatrix} \sum_{j=1}^n \Phi_1(t_j) * y_j \\ \vdots \\ \sum_{j=1}^n \Phi_m(t_j) * y_j \end{bmatrix}_{m \times 1}$$

και

$$\mathbf{A} = \begin{bmatrix} \sum_{j=1}^n \Phi_1(t_j) * \Phi_1(t_j) & \dots & \sum_{j=1}^n \Phi_1(t_j) * \Phi_m(t_j) \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^n \Phi_m(t_j) * \Phi_1(t_j) & \dots & \sum_{j=1}^n \Phi_m(t_j) * \Phi_m(t_j) \end{bmatrix}_{m \times m}$$

ΣΗΜΕΙΩΣΕΙΣ: -Στην περίπτωση αυτή,  $\Phi_i(t_j) = 1 - e^{-t_j / \tau_i}$

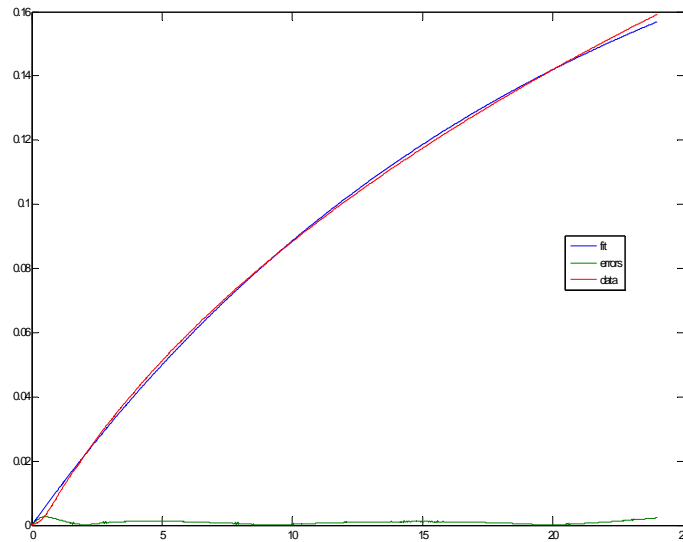
-Για την εφαρμογή των περιορισμών μη αρνητικότητας, το πρόγραμμα επαναλαμβάνει τη διαδικασία ορίζοντας μηδενικές τις αρνητικές τιμές έως ότου ελαχιστοποιηθεί το συνολικό σφάλμα.



### 5.3. Η ΠΡΩΤΗ ΠΡΟΣΑΡΜΟΓΗ

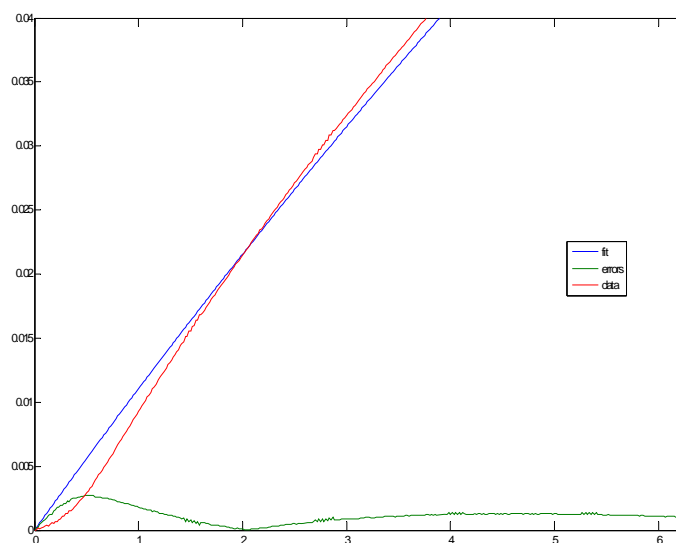
Επιλύοντας το σύστημα  $A \circ w = b$   
υπό τη συνθήκη  $w \geq 0$

για μια ισόθερμη περίοδο, μπορούμε να κάνουμε το γράφημα των δεδομένων και της προσαρμοσμένης γραμμής σε συνάρτηση με τον χρόνο [εικόνα 5-4]:



εικόνα 5- 4

Εάν μεγεθύνουμε το πρώτο μέρος του γραφήματος [εικόνα 5-5], μπορούμε να διακρίνουμε ένα πρόβλημα που προκύπτει από το γεγονός ότι το  $w$  είναι ένα διάνυσμα που αποτελείται από μη-αρνητικές τιμές (φυσικός περιορισμός: τα  $w_i$  αντιστοιχούν σε μάζα). Αυτό σημαίνει ότι δεν μπορούμε να επιτύχουμε την αλλαγή καμπυλότητας που εμφανίζουν τα δεδομένα – η οποία συμβαίνει διότι η αύξηση της θερμοκρασίας πρακτικά, για παράδειγμα από 25 °C σε 50 °C, δεν μπορεί να πραγματοποιηθεί αυτομάτως.



εικόνα 5- 5

Σύμφωνα με την εξίσωση (8) της ενότητας 4, οι παράγοντες επιτάχυνσης για κάθε αύξηση θερμοκρασίας δίνονται από τον λόγο του ρυθμού διάχυσης στη νέα θερμοκρασία προς αυτόν στην αμέσως προηγούμενη θερμοκρασία.

Στην ιδανική περίπτωση της στιγμιαίας αύξησης της θερμοκρασίας του δείγματος, κάθε παράγοντας επιτάχυνσης θα υπολογιζόταν εύκολα με προσαρμογή των δεδομένων δυο περιόδων από ένα άθροισμα εκθετικών συναρτήσεων όπως προτείνεται από την εξίσωση (6), ενότητα 4, λαμβάνοντας έπειτα τον λόγο μεταξύ των παραγώγων αυτών ακριβώς πριν και ακριβώς μετά την αλλαγή της θερμοκρασίας.

Στην πραγματικότητα, ενώ η προσαρμογή των δεδομένων στο τέλος κάθε περιόδου είναι σωστή, καθώς το δείγμα βρίσκεται υπό σταθερή θερμοκρασία, δεν ισχύει το ίδιο για την αρχή κάθε περιόδου. Στην αρχή, καθώς η θερμοκρασία αυξάνεται απαιτούνται μερικά λεπτά έως ότου λάβει την επιθυμητή τιμή και το δείγμα βρεθεί σε ισορροπία. Για τον λόγο αυτό, οι τιμές των δεδομένων δεν αντιστοιχούν στις ιδανικές, δηλαδή σε εκείνες μιας στιγμιαίας αύξησης της θερμοκρασίας. Συνεπώς, το αρχικό μέρος των δεδομένων πρέπει να αντικατασταθεί **με κάποιον τρόπο** με ένα εικονικό σύνολο δεδομένων, κατά το δυνατόν πλησιέστερο στις ιδανικές τιμές [7].

**Στόχος είναι λοιπόν η εύρεση μιας μεθόδου παραγωγής των δεδομένων που θα προέκυπταν αν η μετάβαση μεταξύ των επιθυμητών θερμοκρασιών κάθε περιόδου μπορούσε να συμβεί αυτομάτως, με βήμα.**

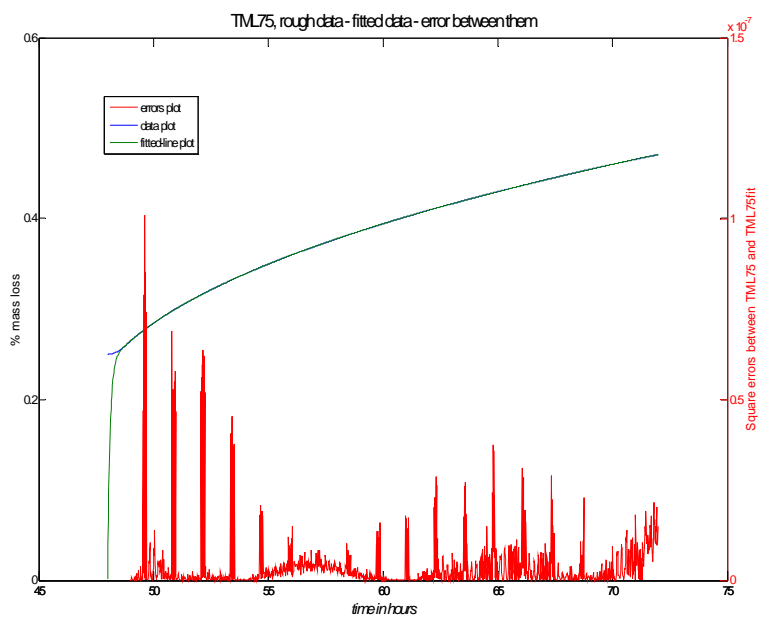
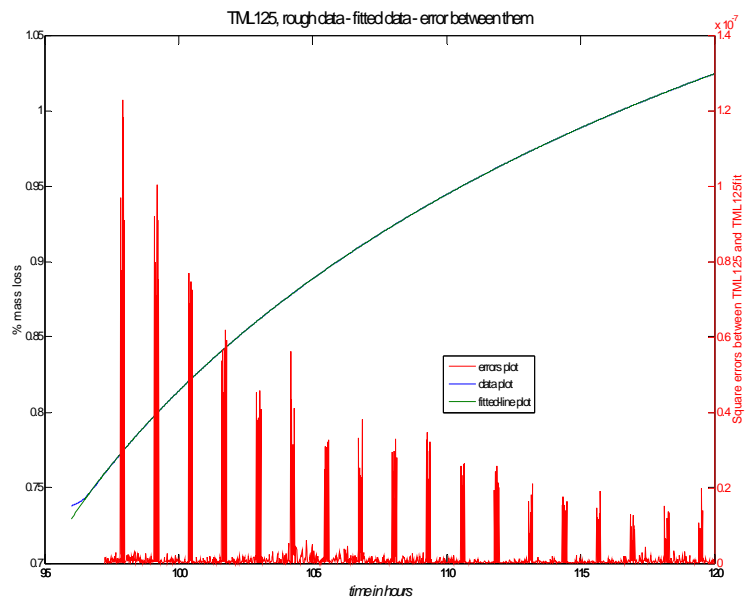
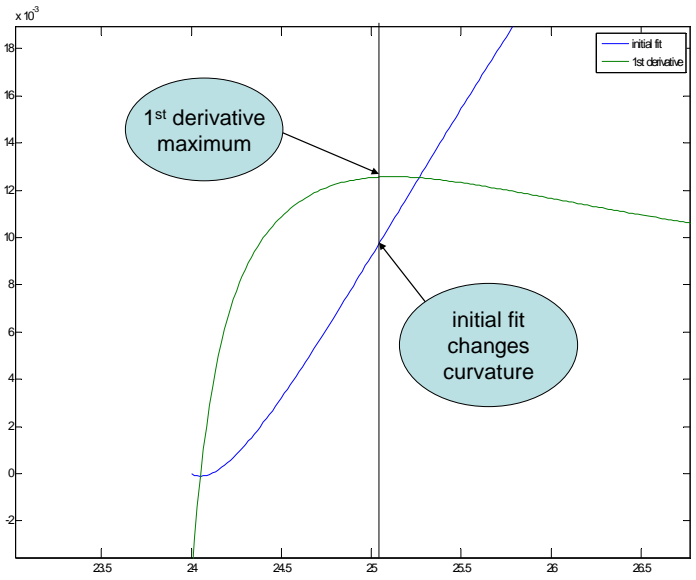
Η διαδικασία αποκοπής και αντικατάστασης μέρους των δεδομένων είναι λεπτή και πολύ σημαντική υπόθεση. Πρέπει αφενός το μέρος αυτό να είναι το ελάχιστο δυνατό, αφετέρου να υπάρχει λογική, συνέπεια και αποτελεσματικότητα στη διαδικασία που θα ακολουθείται. Λαμβάνοντας υπόψη τα παραπάνω, δυο μέθοδοι προτάθηκαν, και οι δυο βασιζόμενες σε μια αρχική προσαρμογή χωρίς περιορισμούς στις τιμές των  $w_i$  και τη μελέτη της παραγωγού αυτής.

### **5.3.1. ΠΡΩΤΗ ΜΕΘΟΔΟΣ – ΧΡΗΣΗ 1<sup>ης</sup> ΠΑΡΑΓΩΓΟΥ**

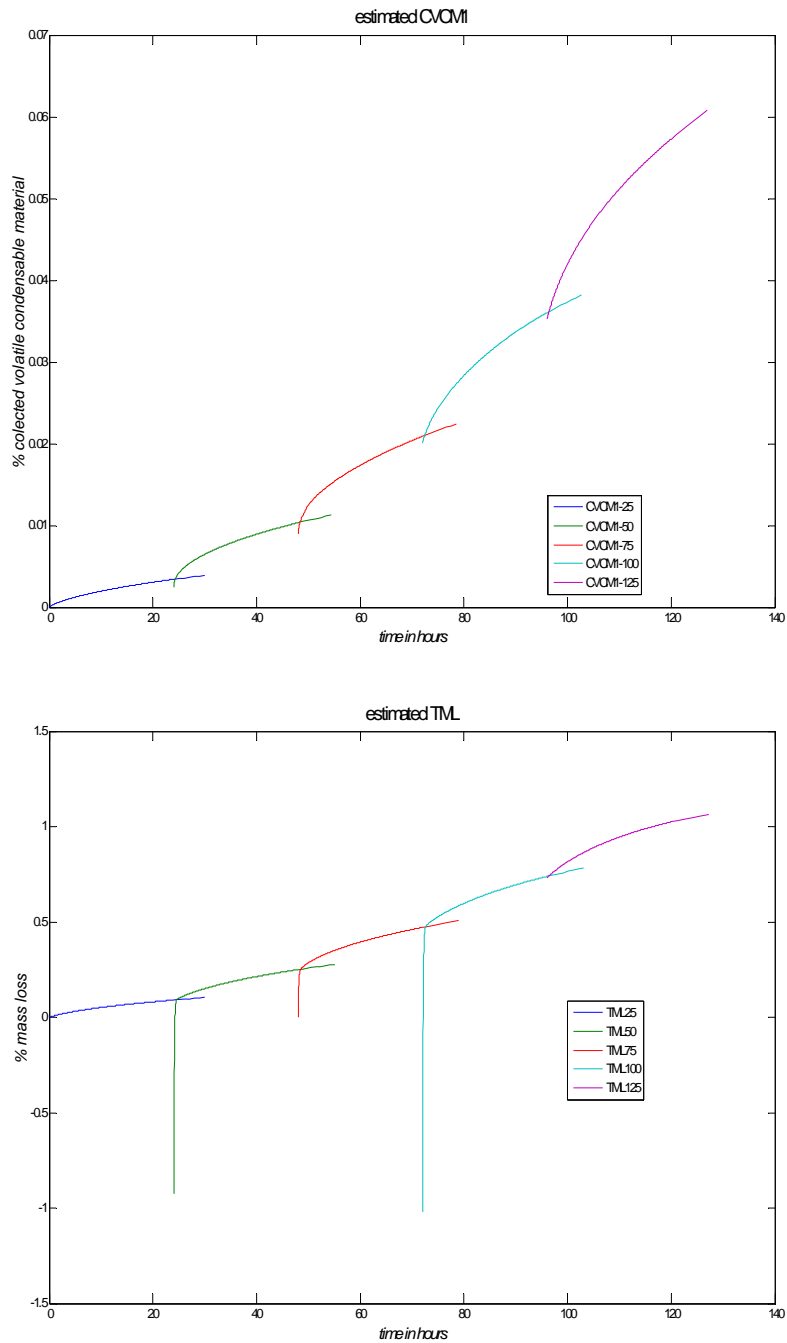
Αναλυτικά, στην πρώτη μέθοδο, προσεγγίζουμε τα δεδομένα επιτρέποντας αρνητικές τιμές για τα  $w_i$ ,  $i=1, \dots, m$ , αν χρειάζεται. Θεωρητικά, αυτή είναι η καλύτερη δυνατή προσαρμογή. Έπειτα, παραγωγίζουμε την αρχική προσαρμογή, βρίσκουμε το σημείο στο οποίο η παράγωγος μεγιστοποιείται, δηλαδή το σημείο εκείνο από το οποίο τα δεδομένα αρχίζουν να αυξάνονται με επιβραδυνόμενο ρυθμό [εικόνα 5-6] και διαγράφουμε όλα τα προηγούμενα δεδομένα. Προσεγγίζουμε ξανά τα δεδομένα με την ίδια συνάρτηση υπό περιορισμούς μη-αρνητικότητας για το διάνυσμα  $w$  και τελικά χρησιμοποιώντας το  $w$  παράγουμε τα δεδομένα που θα υποκαταστήσουν εκείνα που διαγράψαμε κινούμενοι πίσω στο χρόνο [εικόνα 5-7].

Ακολουθώντας τη διαδικασία αυτή, εξετάστηκαν δεδομένα από διάφορα δείγματα. Όμως, επειδή τα ιδανικά δεδομένα δεν αναμένεται να έχουν σταθερό ρυθμό αύξησης, τα αποτελέσματα δεν υπήρξαν πάντα ικανοποιητικά. Συγκεκριμένα, αποδείχθηκε ότι η προς τα πίσω κίνησή μας στο χρόνο είναι μη ελεγχόμενη [εικόνα 5-8].

Τελικά, η συνολική εικόνα των προσαρμοσμένων δεδομένων αποτελούσε μια έκπληξη, κάποτε όντας ‘τέλεια’, κάποτε αγγίζοντας το αδύνατο [εικόνες 5-9, 5-10].



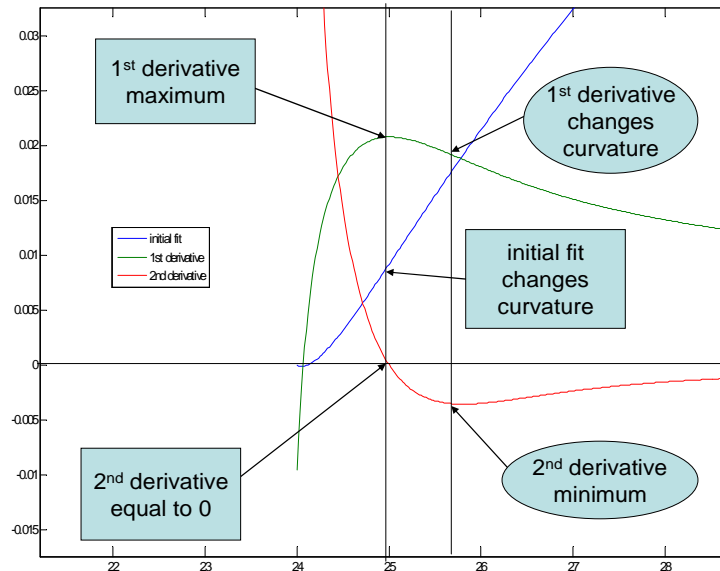
εικόνες 5- 6, 5-7, 5-8



εικόνες 5- 9, 5-10

### 5.3.2. ΔΕΥΤΕΡΗ ΜΕΘΟΔΟΣ – ΧΡΗΣΗ 2<sup>ης</sup> ΠΑΡΑΓΩΓΟΥ

Μια δεύτερη μέθοδος προτάθηκε ώστε να πλησιάσουμε την ‘ιδανική’ συμπεριφορά των δεδομένων. Αυτή η μέθοδος κάνει χρήση και της δεύτερης παραγώγου της πρώτης άνευ περιορισμών αρνητικότητας προσαρμογής. Συγκεκριμένα, διαγράφουμε περισσότερα (σε σχέση με την προηγούμενη μέθοδο) από τα πραγματικά δεδομένα βρίσκοντας το σημείο που ελαχιστοποιείται η δεύτερη παράγωγος [εικόνα 5-11], και έπειτα παράγουμε από εκείνο το σημείο και πίσω εικονικά δεδομένα υποθέτωντας γραμμικό ρυθμό αύξησης με κλίση την ελάχιστη τιμή που ευρέθηκε – αυτό διότι δεν θέλουμε να αλλάζει η καμπυλότητα της πρώτης παραγώγου [7].



εικόνα 5-11

Η διαδικασία συνοψίζεται ως εξής [εικόνα 5-12]:

- βρίσκουμε τον χρόνο  $\bar{t}$  :

$$\left[ \frac{d^2 y}{dt^2} \right]_{\bar{t}} = \min \left\{ \left[ \frac{d^2 y}{dt^2} \right] \right\}$$

- αντικαθιστούμε την 1<sup>η</sup> παράγωγο από  $t=0$  ως  $t=\bar{t}$  με τη συνάρτηση:

$$\frac{dy}{dt} = \left[ \frac{dy}{dt} \right]_{\bar{t}} - \left[ \frac{d^2 y}{dt^2} \right]_{\bar{t}} \cdot (\bar{t} - t)$$

- αντικαθιστούμε τις y-τιμές από  $t=0$  ως  $t=\bar{t}$  με τη συνάρτηση:

$$y = \bar{y} - \left[ \frac{dy}{dt} \right]_{\bar{t}} \cdot (\bar{t} - t)$$

- προσεγγίζουμε τα νέα δεδομένα

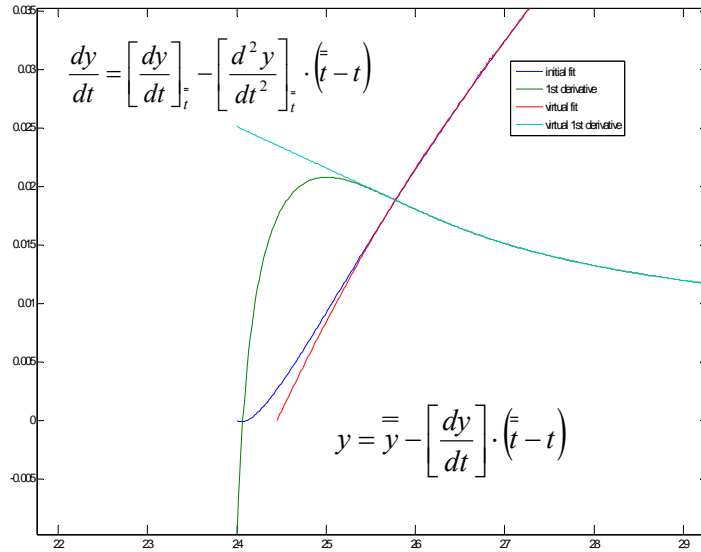
Σημείωση: από  $t=\bar{t}$  ως  $t=t\text{-last}$ , τα δεδομένα παραμένουν ίδια.

Αυτή η προσέγγιση κρίθηκε αρκούντως ικανοποιητική και είναι αυτή που προτιμήθηκε [εικόνα 5-13].

Σημείωση: Τα σχετικά σφάλματα εμφανίζονται πολύ μεγάλα στην αρχή. Αυτό δεν οφείλεται σε κακή προσαρμογή. Εξηγείται εύκολα από τον ορισμό του σχετικού σφάλματος:

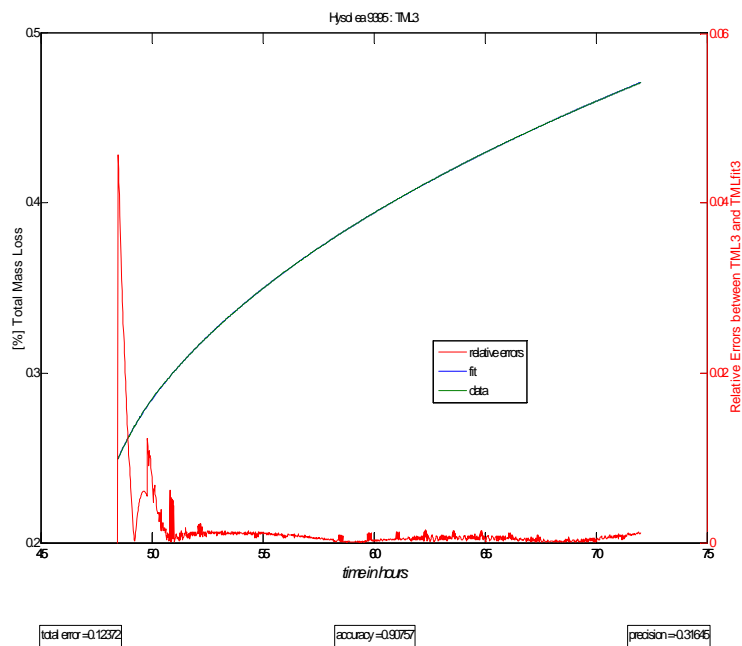
$$RelErr = \frac{|y - y_{fit}|}{y} \text{ και αρχικά } y \rightarrow 0$$

(τα δεδομένα πάντα κανονικοποιούνται/μεταφέρονται στο (0,0) για να προσαρμοστούν)



εικόνα 5-12

Φυσικά, όσο καλύτερη είναι η προσαρμογή τόσο ακριβέστερη είναι η εύρεση του σημείου αποκοπής και τόσο πλησιέστερη στη φυσική του φαινομένου θα είναι η εξαγωγή των εικονικών δεδομένων. Μια κακή προσαρμογή θα δώσει ανακριβή θέση του σημείου αποκοπής και, σημαντικότερο όλων, εσφαλμένη κλίση του ρυθμού διάχυσης.



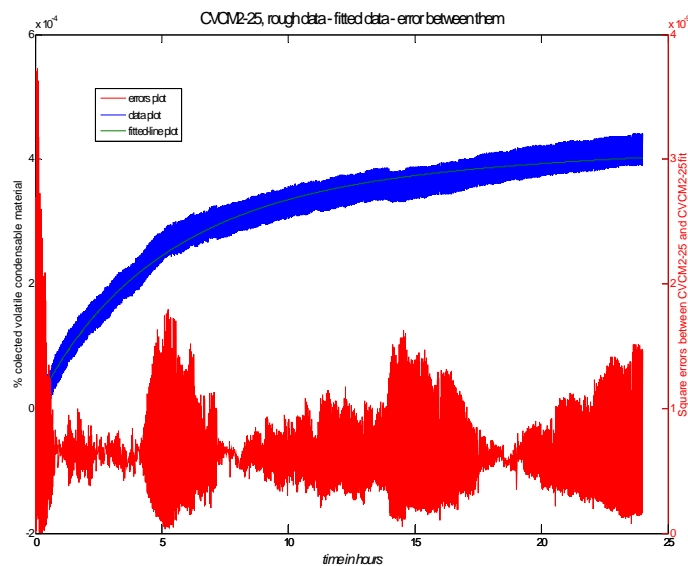
εικόνα 5-13

### 5.3.3. ΕΝΑΛΛΑΚΤΙΚΕΣ

Όπως ειπώθηκε, για κάθε περίοδο, μετά το σημείο αποκοπής, το δείγμα θεωρείται ότι βρίσκεται υπό σταθερή θερμοκρασία και τα πραγματικά δεδομένα που καταγράφηκαν χρησιμοποιούνται στην ανάλυση. Έτσι οι καμπύλες TML και CVCM αποτελούνται συνολικά από ένα εικονικό αρχικό μέρος παραβολικού σχήματος ως προσδιορίστηκε στην προηγούμενη παράγραφο και, μετά το σημείο αποκοπής, από τα πραγματικά δεδομένα.

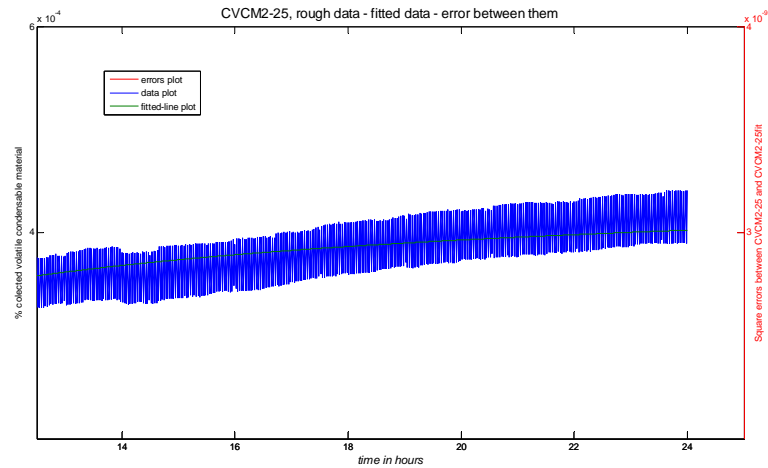
Οι νέες καμπύλες προσαρμόζονται ξανά από ένα άθροισμα εκθετικών συναρτήσεων (εξίσωση 6, ενότητα 4), αλλά αυτή τη φορά έχοντας απαλείψει την επίδραση της σταδιακής αύξησης της θερμοκρασίας στα δεδομένα στο προηγούμενο βήμα, η προσαρμογή γίνεται υπό περιορισμούς μη αρνητικότητας. Μάλιστα, σε αυτή την προσαρμογή, αν θέσουμε αρκετά χαμηλά την ανοχή στις τιμές, κάποιες από τις μάζες τείνουν στο μηδέν προσφέροντάς μας πληροφορίες σχετικά με τους χρόνους παραμονής (ανάμεσα από τις χρονικές σταθερές που χρησιμοποιούνται) που συνεισφέρουν στην προσαρμογή.

Ενίστε, ιδίως για τα δεδομένα που αφορούν στη συμπύκνωση στους συλλέκτες, η διαδικασία που περιγράφηκε προηγουμένως δεν είναι ικανοποιητική [εικόνα 5-14]. Σε αντίθεση με τις καμπύλες TML, οι καμπύλες CVCM προκύπτουν όχι από άμεση μέτρηση μάζας αλλά από μέτρηση της συχνότητας ταλάντωσης που μεταφράζεται σε μάζα μέσω της ευαισθησίας των συλλεκτών. Αυτό έχει ως αποτέλεσμα οι καμπύλες αυτές να είναι 'πυκνές', να υπάρχει δηλαδή έντονη διακύμανση στα δεδομένα.



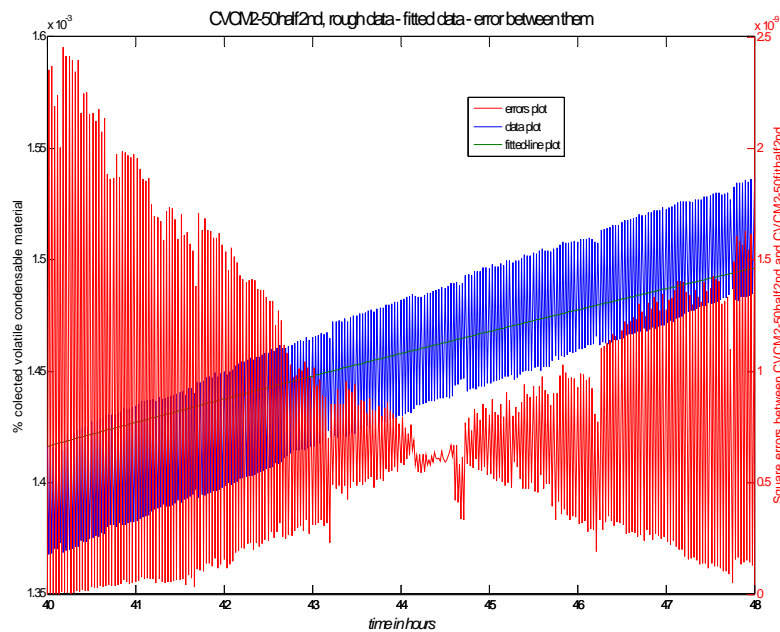
εικόνα 5-14

Τελικά, η προσαρμοσμένη καμπύλη περνά μέσα από τα δεδομένα ελαχιστοποιώντας το συνολικό σφάλμα, όμως επειδή μεγαλύτερο ενδιαφέρον για την ανάλυσή μας έχουν οι ρυθμοί αύξησης στην αρχή και στο τέλος κάθε περιόδου, είναι μεγάλης σημασίας η προσαρμογή να 'ακολουθεί' την πορεία των δεδομένων. Εστιάζοντας στο τελευταίο μέρος της [εικόνας 5-14], παρατηρούμε ότι η προσέγγιση δεν αντιπροσωπεύει τα πραγματικά δεδομένα, που φαίνεται να παρουσιάζουν μεγαλύτερη αυξητική τάση από την προσαρμοσμένη γραμμή [εικόνα 5-15].



εικόνα 5-15

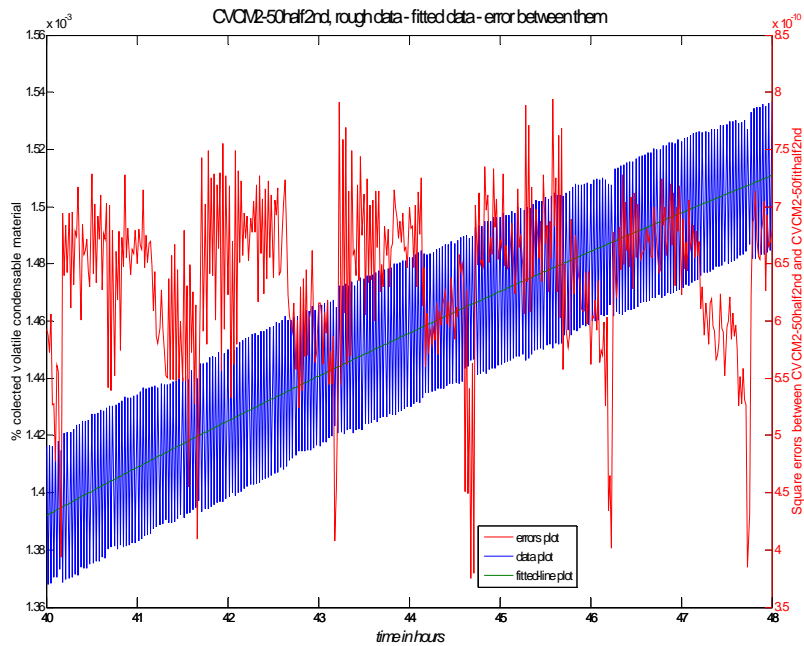
Τέσσερις ιδέες συζητήθηκαν για την επίλυση αυτού του προβλήματος, τρεις εξ αυτών απαιτούν τον διαχωρισμό των δεδομένων.



εικόνα 5-16

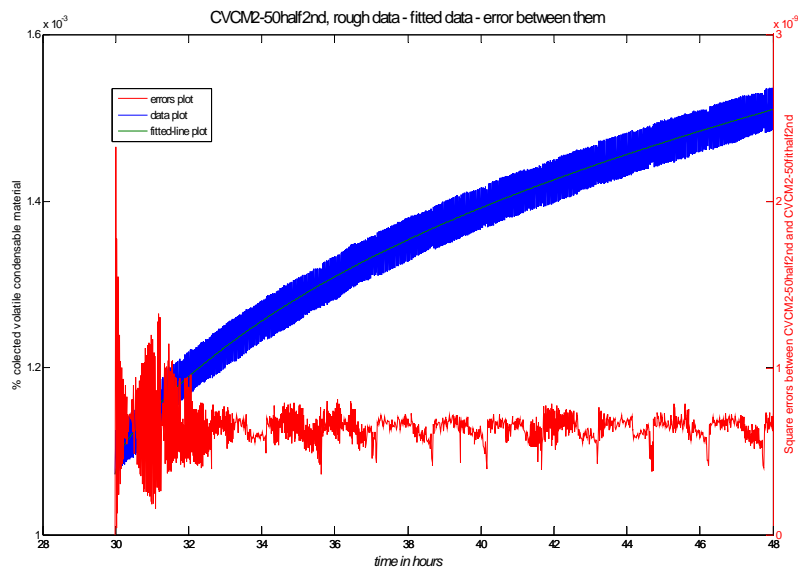
- (1) Στην [εικόνα 5-16] , απεικονίζεται η προσαρμογή των τελευταίων 8 ωρών μιας συγκεκριμένης περιόδου. Η ιδέα είναι να προσεγγίσουμε τα πρώτο και τελευταίο μέρη μιας περιόδου και να πάρουμε διαφορετικές τιμές του  $w$  για την αρχή και το τέλος. Για το πρώτο μέρος η ιδέα λειτούργησε επιτυχώς, αλλά για το τελευταίο δεν λειτούργησε διόλου.
- (2) Η δεύτερη ιδέα είναι παρόμοια με την πρώτη με τη μόνη διαφορά ότι τα δεδομένα του τελευταίου μέρους προσεγγίζονται χωρίς μεταφορά στο  $(0,0)$  [εικόνα 5-17]. Αυτή η ιδέα αποδείχθηκε λειτουργική.





εικόνα 5-17

(3) Η τρίτη ιδέα είναι και αυτή παρόμοια με την πρώτη με τη διαφορά ότι το τελευταίο μέρος των δεδομένων αποτελείται από το τελευταίο 75% όλων των δεδομένων [εικόνα 5-18]. Και αυτή η ιδέα λειτούργησε επιτυχώς, αλλά είναι περίπλοκη καθώς σ' αυτή την περίπτωση το πρώτο και το τελευταίο μέρος των δεδομένων επικαλύπτονται μερικώς.

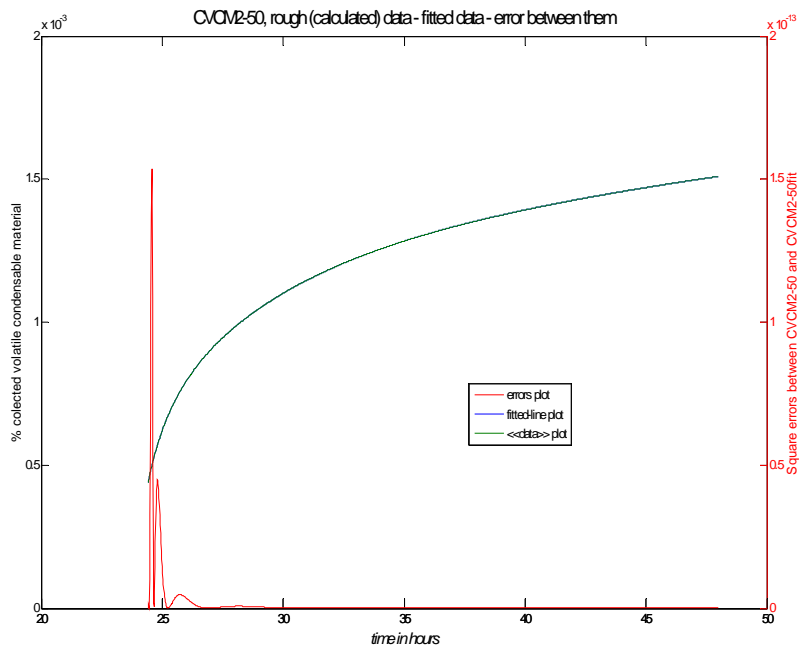


εικόνα 5-18

Οι τρεις ιδέες που μόλις αναφέρθηκαν περιλαμβάνουν διαχωρισμό των δεδομένων (η τρίτη και μερική επικάλυψη) με αποτέλεσμα κάθε περίοδος να περιγράφεται από δυο διανύσματα  $\mathbf{w}$ . Επειδή αυτός ο διαχωρισμός αφενός περιπλέκει τη διαδικασία, αφετέρου δεν αντιστοιχεί στη φυσική του φαινομένου, προτιμήθηκε σε τέτοιες περιπτώσεις να χρησιμοποιείται η ιδέα που ακολουθεί.

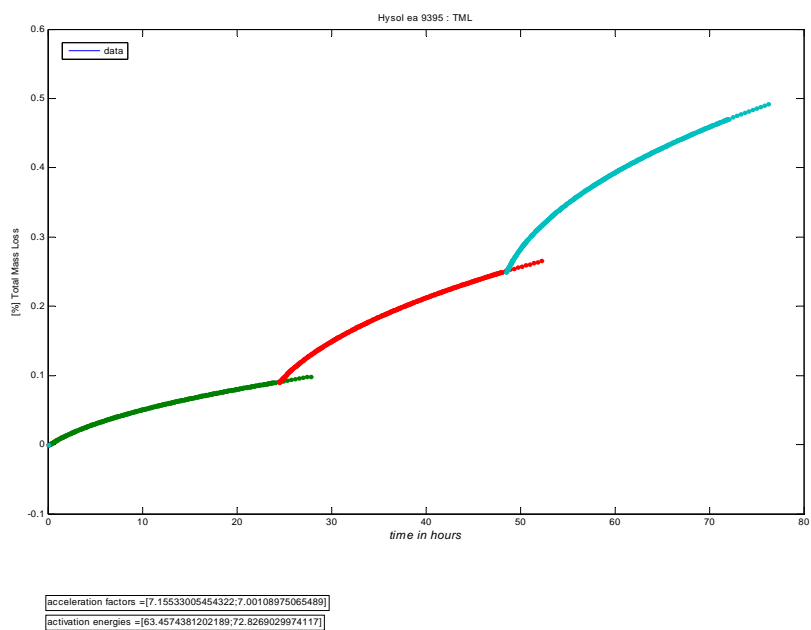
(4) Η τέταρτη ιδέα βασίζεται στην αρχική προσαρμογή [εικόνα 5-19]. Εφόσον η αρχική προσαρμογή είναι θεωρητικά η καλύτερη δυνατή, μπορεί να χρησιμοποιείται σε τέτοιες προβληματικές περιπτώσεις. Δύο είναι τα πλεονεκτήματα αυτής και είναι προφανή:

- Η αρχική προσαρμογή αντιπροσωπεύει τα δεδομένα με τον βέλτιστο τρόπο.
- Η αρχική προσαρμογή είναι απλώς μια γραμμή, δεν υπάρχουν διακυμάνσεις σ' αυτή.



εικόνα 5-19

Τελικά, σε κάθε περίπτωση, ευρέθη ο τρόπος οι προσαρμοσμένες γραμμές να σέβονται τη φυσική του φαινομένου διάχυσης [εικόνα 5-20] ώστε να προχωρήσει η διαδικασία επιτάχυνσης αυτού.



εικόνα 5-20

#### 5.4. ΔΙΑΔΙΚΑΣΙΑ ΕΠΙΤΑΧΥΝΣΗΣ

Μετά την προσαρμογή, για κάθε αύξηση θερμοκρασίας, προσδιορίζεται το σημείο τομής των διαδοχικών καμπυλών και για εκείνο το σημείο οι παράγοντες επιτάχυνσης και οι ενέργειες ενεργοποίησης μπορούν να υπολογιστούν [εικόνα 5-21] σύμφωνα με τις εξισώσεις 8 και 9 (ενότητα 4).

## Παράγοντες Επιτάχυνσης & Ενέργειες Ενεργοποίησης

$$K_{ij} = \frac{\left[ \frac{dW}{dt} \right]_j}{\left[ \frac{dW}{dt} \right]_i} \quad (\text{όπως υπολογίζονται στα σημεία τομής})$$

$$E_{ij} = \frac{R \cdot T_i \cdot T_j}{T_j - T_i} \cdot \ln(K_{ij})$$

$i=1, \dots, m-1$  and  $j=2, \dots, m$  (m είναι ο αριθμός των χρονικών περιόδων)

όπου  $R=8.314472 \text{ J/moleK}$   
και  $T$  η θερμοκρασία σε K

εικόνα 5- 21

Επίσης, μπορεί να υπολογιστεί ο Ke-συντελεστής (συντελεστής εξάρτησης θερμοκρασίας – χρόνου παραμονής) που δίνεται από την ακόλουθη εξίσωση:

$$\left( \frac{dW}{dt} \right)_T = \left( \frac{dW}{dt} \right)_{T_{ref}} \cdot e^{Ke \cdot (T - T_{ref})}$$

Στην περίπτωση 5 περιόδων, για παράδειγμα, παίρνουμε τις παρακάτω 4 σχέσεις:

$$\ln(K_{12}) = Ke_1 \cdot (T_2 - T_1)$$

$$\ln(K_{12} \cdot K_{23}) = Ke_2 \cdot (T_3 - T_1)$$

$$\ln(K_{12} \cdot K_{23} \cdot K_{34}) = Ke_3 \cdot (T_4 - T_1)$$

$$\ln(K_{12} \cdot K_{23} \cdot K_{34} \cdot K_{45}) = Ke_4 \cdot (T_5 - T_1)$$

Ο Ke μπορεί να είναι η μέση τιμή ή μπορεί να υπολογιστεί από το απλό γραμμικό μοντέλο παλινδρόμησης.

Ο σκοπός των τεστ διάχυσης είναι η πρόβλεψη. Οι παράγοντες επιτάχυνσης χρησιμοποιούνται για να επιταχύνουμε τον χρόνο εξέλιξης του πειράματος και τις χρονικές σταθερές ώστε να ανταποκρίνονται στον νέο εικονικό χρόνο, και να προβλέψουμε τι θα συμβεί σ' ένα συγκεκριμένο υλικό μετά από χιλιάδες ώρες. Παρακάτω περιγράφουμε τις σχέσεις σύμφωνα με τις οποίες πραγματοποιείται η εν λόγω επιτάχυνση.

-Από  $t=0$  ως  $t=$ διάρκεια πρώτης περιόδου

$$acceleratedTime_1 = actualTime_1$$

$$\tau_1 = DurationPeriod_1$$

-Από  $t=$ διάρκεια περιόδου  $i$  ως  $t=$ διάρκεια περιόδου  $i+1$

$$acceleratedTime_i = actualTime_i \cdot \prod_{k=1}^{i-1} K_k + acceleratedTime_{i-1}$$

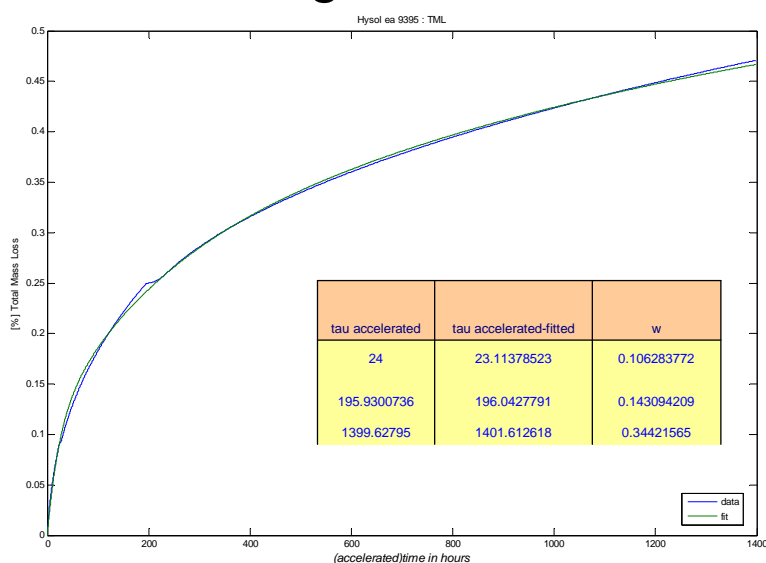
$$\tau_i = DurationPeriod_i \cdot \prod_{k=1}^{i-1} K_k + \tau_{i-1}$$

όπου  $i=2, \dots, m$ ,  $m$  είναι ο αριθμός των περιόδων

## 5.5. ΜΑΚΡΟΠΡΟΘΕΣΜΕΣ ΠΡΟΒΛΕΨΕΙΣ

Με βάση τα επιταχυνθέντα διανύσματα χρόνου και  $\tau$  και τα αρχικά δεδομένα, προχωρούμε σε μια μακροχρόνια προσαρμογή [εικόνα 5-22].

### Long – Term fit



εικόνα 5- 22

Αυτή η προσέγγιση περιλαμβάνει προσαρμογή (υπό περιορισμούς μη αρνητικότητας) των διανυσμάτων  $w$  και  $\tau$ , αλλά όχι ταυτόχρονα. Πρώτα προσεγγίζουμε το  $w$ , έπειτα βελτιστοποιούμε το  $\tau$ . Αυτή η διαδικασία είναι επαναληπτική. Τερματίζει όταν το συνολικό σφάλμα σχεδόν δεν μεταβάλλεται πλέον – και φυσικά είναι ελαχιστοποιημένο.

Η  $w$ -προσαρμογή βασίζεται στο κριτήριο των ελαχίστων τετραγώνων. Η  $\tau$ -βελτιστοποίηση επιτυγχάνεται με χρήση μιας συνάρτησης που παρέχει το MatLab®, ονομάζεται `fminsearch.m` και υλοποιεί τον αλγόριθμο Nelder-Mead.

Σημείωση: Η `fminsearch.m` πραγματοποιεί βελτιστοποίηση χωρίς περιορισμούς. Δεν είναι πιθανό να προκύψουν αρνητικές τιμές για το  $\tau$  μετά τη βελτιστοποίηση, αλλά σε περίπτωση που διαθέτουμε το MatLab® optimization toolbox στον υπολογιστή μας, χρησιμοποιούμε τη συνάρτηση `fmincon.m` για τη βελτιστοποίηση, συνάρτηση που πραγματοποιεί βελτιστοποίηση υπό περιορισμό.

Στο τελικό στάδιο, μετά τη μακροχρόνια προσαρμογή, μπορούμε να υπολογίσουμε το διάνυσμα  $\tau$  για προβλέψεις σύμφωνα με την εξίσωση 14 (ενότητα 4) για διάφορες θερμοκρασίες [εικόνα 5-23].

## Διανύσματα Χρόνου Παραμονής $\tau$ σε διάφορες θερμοκρασίες

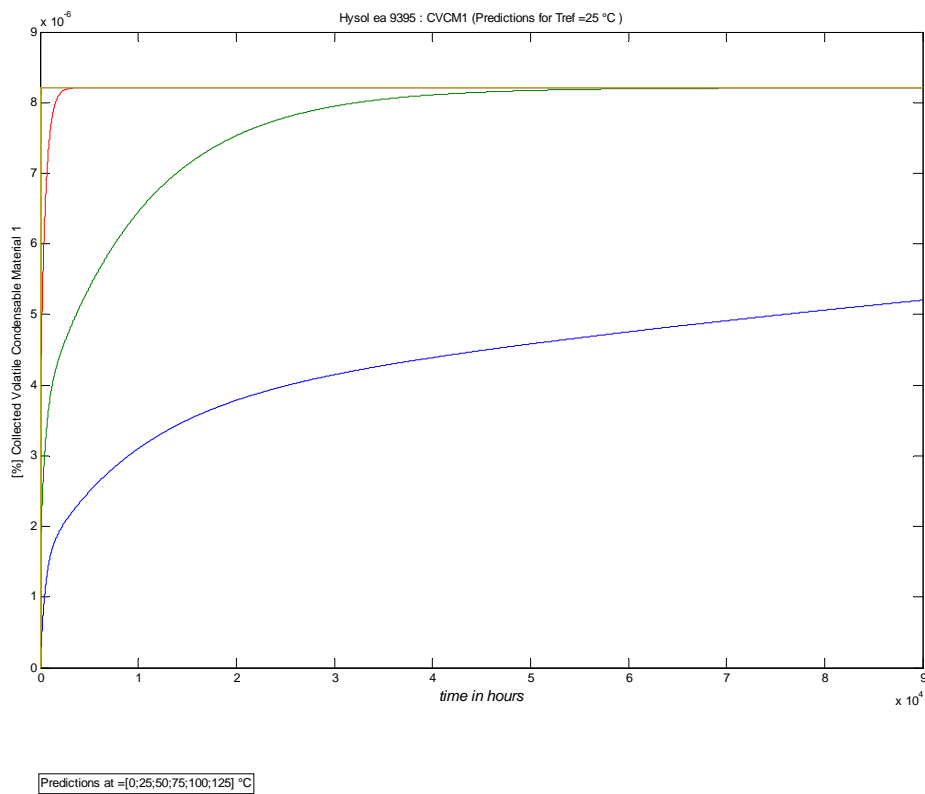
$$TAU_{atT^{\circ}C} = TAU_{acceleratedfitted} \cdot e^{(-Ke \cdot (T - T_{ref}))}$$

Θερμοκρασία Αναφοράς = 25°C

	tau L-T-P at	tau L-T-P at	tau L-T-P at	tau L-T-P at	tau L-T-P at	tau L-T-P at
Temperature (°C)	0	25	50	75	100	125
	165.0183377	23.11378523	3.237501208	0.453470255	0.063516663	0.008896651
	1399.625947	196.0427791	27.45931607	3.846170935	0.538725394	0.075458178
	10006.65975	1401.612618	196.3210481	27.49829264	3.851630305	0.539490077

εικόνα 5- 23

Χρησιμοποιώντας τα προβλεπτικά διανύσματα  $\tau$  και το διάνυσμα  $w$  που προέκυψε από τη μακροχρόνια προσαρμογή, μπορούμε να έχουμε τις προβλέψεις για μερικές χιλιάδες ώρες σε διάφορες θερμοκρασίες [εικόνα 5-24] (τυπικά αυτές που χρησιμοποιούμε στη διάρκεια του τεστ).



**εικόνα 5- 24**

Στο άνωθεν γράφημα, βλέπουμε προβλέψεις για 90.000 ώρες του συγκεντρωμένου πτητικού συμπυκνώσιμου υλικού στον μικροζυγό κρυστάλλου χαλαζία που ψύχεται στους -25 °C, δηλαδή στον πρώτο συλλέκτη (CVCM1).

Σημειώνουμε ότι η μπλε γραμμή αντιστοιχεί σε θερμοκρασία υλικού=0 °C, η πράσινη σε θερμοκρασία υλικού=25 °C, και λοιπά...

## 6. ΕΚΤΕΛΕΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Το MatLab® διαβάζει τα απαιτούμενα δεδομένα για την επεξεργασία από αρχεία excel®. Για να λειτουργήσει το πρόγραμμα, απαιτούνται συγκεκριμένες πληροφορίες από τον χρήστη που αφορούν το συγκεκριμένο υπό μελέτη υλικό. Αυτές οι πληροφορίες αφορούν το όνομα του υλικού και την αρχική του μάζα, τον πραγματικό χρόνο έναρξης του πειράματος (χρόνος 0 για την επεξεργασία), την χρονική διάρκεια κάθε περιόδου, το τ-διάλυμα, τη θερμοκρασία αναφοράς κλπ, και πρέπει να καταγραφούν στο αρχείο vbqc.xls [εικόνα 6-1].

The screenshot shows a Microsoft Excel spreadsheet with the following data and instructions:

**Sample Information:**

- Sample: hysol ea 9395
- Date of test: july 15, 2008
- mass (g): 6.142
- area (cm<sup>2</sup>): 100.000
- note: area is guessed
- Gas Constant R: 8.314472 J/mole/K

**View Factors and Sensitivity:**

	Temp. (°C)	View Factor	Sensitivity g/(Hz cm <sup>2</sup> )
TML			
CVCM 1		86	1.96E-09
CVCM 2		86	1.96E-09
CVCM 3		86	1.96E-09

**Test Parameters:**

- # Measurements: 10020
- Reference Temperature °C: 25
- actual test start (h): 2
- # periods: 3

**Instructions (from the blue box):**

**INSTRUCTIONS**

Paste all data collected from the experiment in "data" sheet A1  
 Write the name and date of the sample tested in B1 and B2 cells respectively  
 Write the mass and area of the sample in B7 and B8 cells respectively  
 The value of the gas constant is 8.314472 J/mole/K  
 Write the view factors of QCMs in cells C17:C19  
 Write the sensitivity of QCMs in cells D17:D19  
 Write the actual test start time in A25 cell  
 Write the duration of each test period in cells D25:D...  
 (NOTE: if a period is not used, leave cell empty (do not write 0))  
 Write the TAU vector in cells G25:G...  
 (NOTE: zero values are not accepted (TAU is a denominator))  
 Write the reference temperature for predictions in cell J23  
 Write the temperatures in which you want to see predictions in cells J25:J...

**Test Periods and Prediction Parameters:**

period number	duration (h)	Average Temp (°C)	TAU	length of tau	Temperature for Predictions °C	length of Pre
1	24		50	6	0	6
2	24		20		25	
3	24		5		50	
4			1		75	
5			0.5		100	
6			0.1		125	
7						
8						
9						

εικόνα 6- 1

Μετά τη συμπλήρωση όλων των απαραίτητων πεδίων, το πρόγραμμα μπορεί να λειτουργήσει.

Η εντολή στο MatLab® είναι η execute. Ο χρήστης πληκτρολογεί την εντολή και καλείται να απαντήσει σε 3 συγκεκριμένα ερωτήματα ως φαίνεται παρακάτω:

>> execute

Welcome to the MatLab data-fitting program for the outgassing tests.

The procedure is based on the theory developed by M. Van Eesbeek and A. Zwaal:  
"Outgassing and Contamination Model based on Residence Time"

-----  
How do you want the TML and CVCMS to be expressed as? [1: % - 2: mg/cm2 - 3:  
ask me later] :

-----  
Constrained optimization is part of the optimization toolbox.

The method used is "fmincon.m".

Unfortunately, it usually gets trapped in local minima because it is designed to work on problems where the objective and constraint functions are both continuous and have continuous first derivatives.

Method "fminsearch.m" performs unconstrained optimization.

It is suggested to use "fminsearch.m" - because it can often handle discontinuity - unless you notice negative values in the tau vector.

(NOTE: YES - acceptable only if the optimization toolbox is installed in your computer)

Do you want to use constrained optimization for the tau? [1: YES - 2: NO - 3: ask me later] :

-----  
Do you want to see the results in EXCEL files? [1: YES - 2: NO - 3: ask me later] :

-----  
MATLAB will proceed to a basic transmission of information to EXCEL files.

This means that you will be able to find all the information about the initial fits, the "final" fits, the intersection times, the acceleration factors, the activation energies, the Ke coefficients, the long-term fits and the long-term predictions in EXCEL files.

Any intermediate or further information might be obtained by the MATLAB via commands and possible calculations.

---

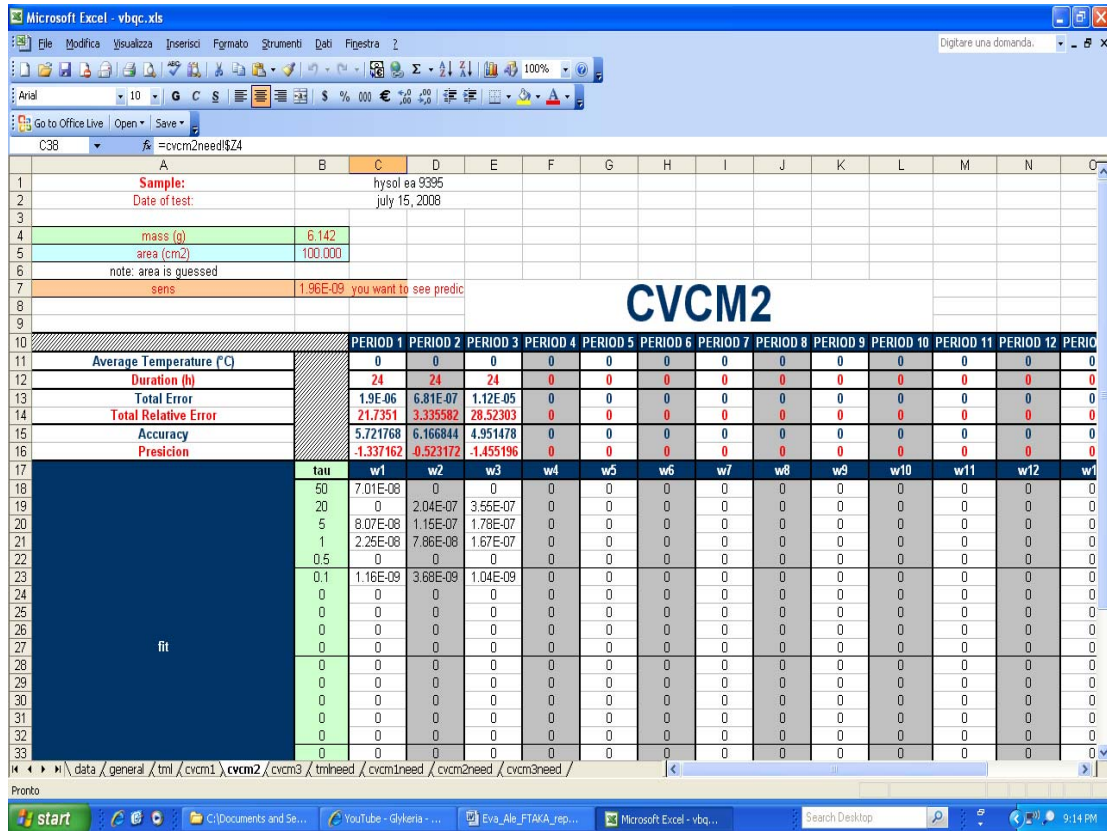
Μετά την ολοκλήρωση της επεξεργασίας από το MatLab®, ο χρήστης μπορεί να επιλέξει αν και ποια διαγράμματα θέλει να δει, μεμονωμένα ή ομαδοποιημένα με κάποιο κριτήριο π.χ. όσα αφορούν τη συνολική απώλεια μάζας ή όσα αφορούν τις μακροπρόθεσμες προβλέψεις, ή όλα ή και κανένα.

Ανάλογα με την απάντηση στην τρίτη ερώτηση, ο χρήστης μπορεί να δει κάποια από τα αποτελέσματα σε αρχεία excel®.

Το αρχείο vbqc.xls τροφοδοτείται με τα στοιχεία που προέκυψαν από την προσαρμογή των δεδομένων σε κάθε περίοδο, συμπεριλαμβανομένων και όσων παρήχθησαν στα ενδιάμεσα στάδια. Αυτά περιλαμβάνουν τις τιμές w που αντιστοιχούν σε κάθε χρόνο παραμονής τ για κάθε περίοδο, για τη συνολική απώλεια μάζας και για κάθε συλλέκτη. Διαθέσιμες είναι επίσης πληροφορίες περί του συνολικού και του σχετικού σφάλματος καθώς και περί της ακρίβειας και απακρίβειας αυτών.

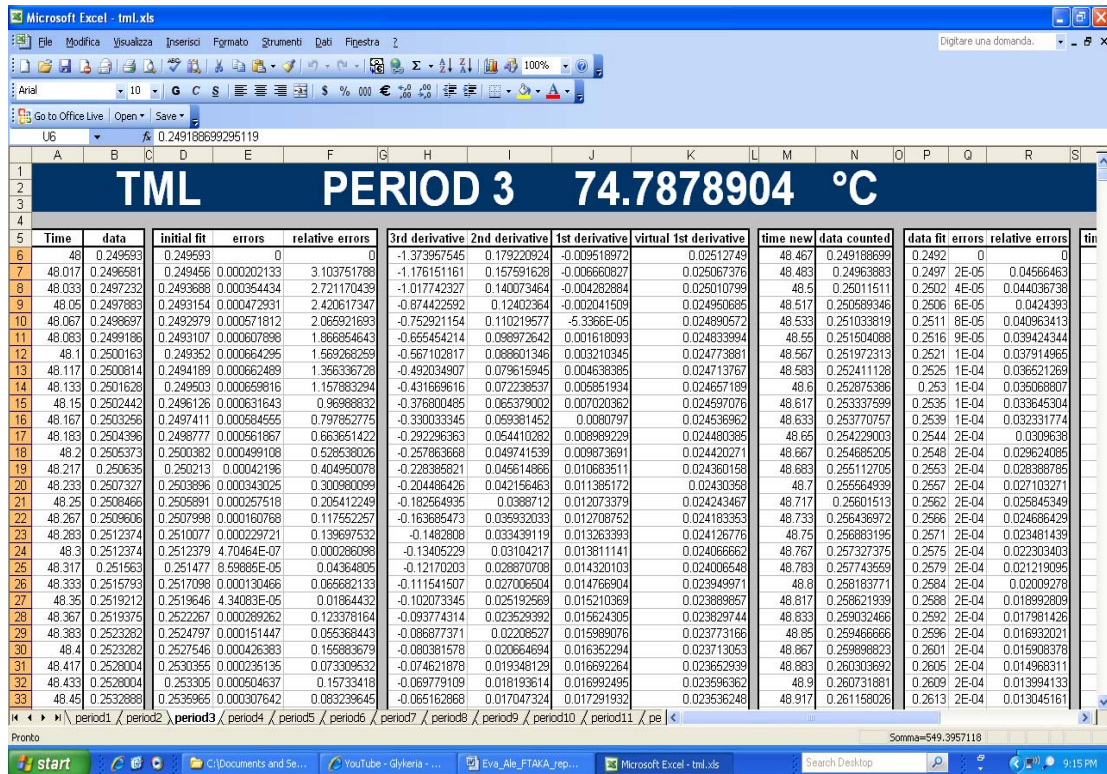
Στην [εικόνα 6-2] φαίνονται τα παραπάνω για τον συλλέκτη με θερμοκρασία -50 °C, CVCM2.





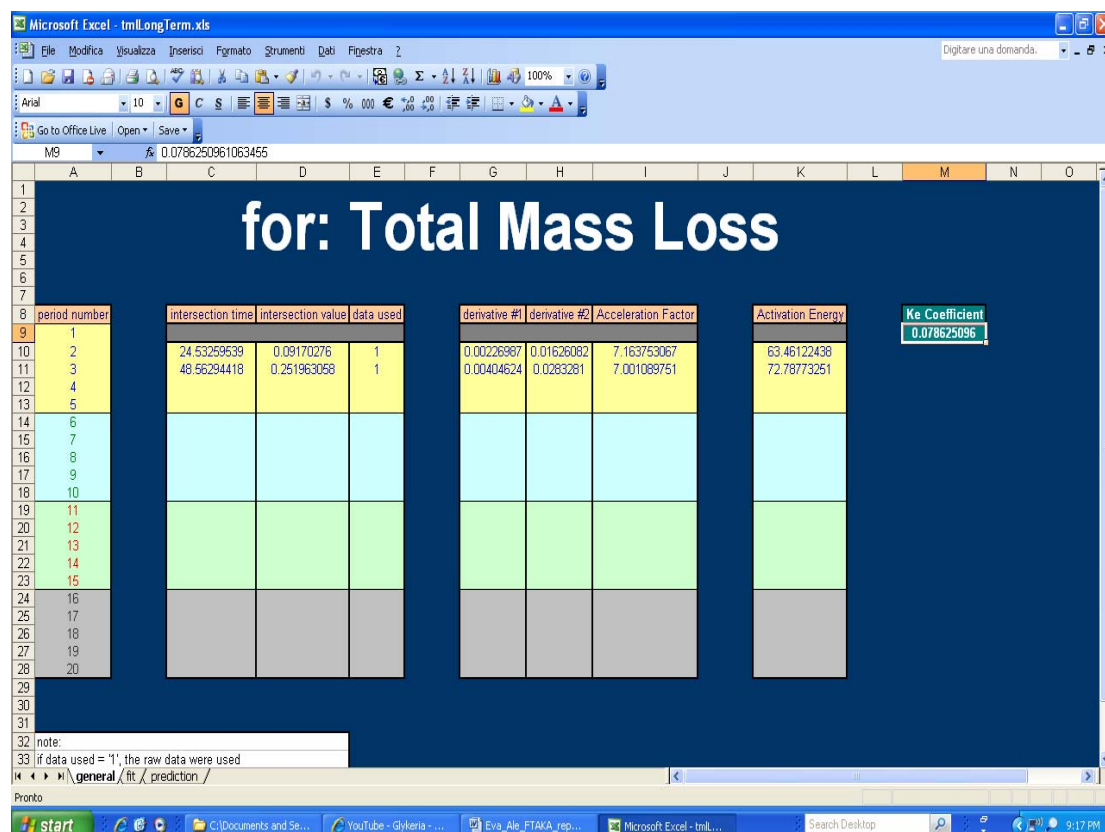
εικόνα 6-2

Το πρόγραμμα δημιουργεί επίσης δυο αρχεία excel® για καθένα από τα τέσσερα μεγέθη που μελετούμε (TML, CVCM1, CVCM2, CVCM3). Το πρώτο αφορά την προσαρμογή των δεδομένων και το δεύτερο την μακροπρόθεσμη πρόβλεψη.



εικόνα 6-3

Σε κάθε αρχείο που αφορά την προσαρμογή των δεδομένων φαίνονται για κάθε περίοδο η μέση θερμοκρασία και αναλυτικά σε κάθε χρόνο μέτρησης τα πραγματικά δεδομένα, η αρχική προσαρμογή, οι πρώτες τρεις παράγωγοι αυτής, η εικονική πρώτη παράγωγος, τα εικονικά-ιδανικά δεδομένα και οι νέοι χρόνοι, και φυσικά η οριστική προσαρμογή με όλα τα σχετικά σφάλματα [εικόνα 6-3, για την tml]. Οι πληροφορίες αυτές είναι πολύτιμες για μελλοντική επεξεργασία και για λόγους επικύρωσης υποθέσεων και εκτιμήσεων.

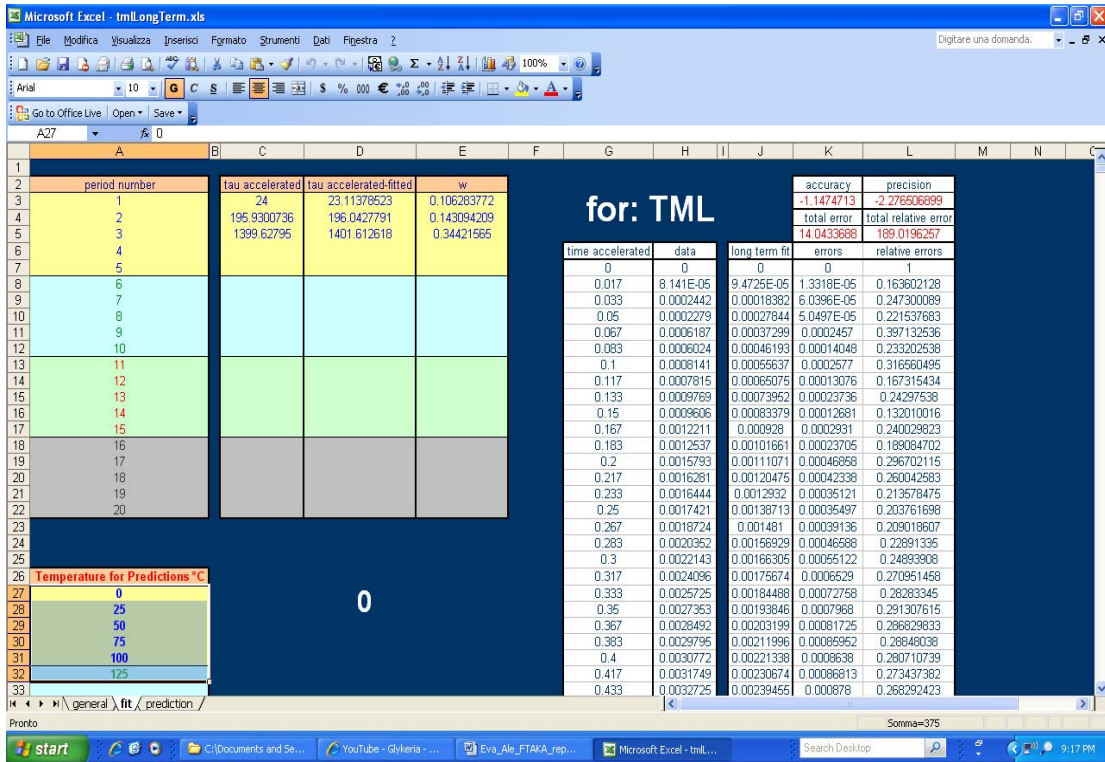


εικόνα 6- 4

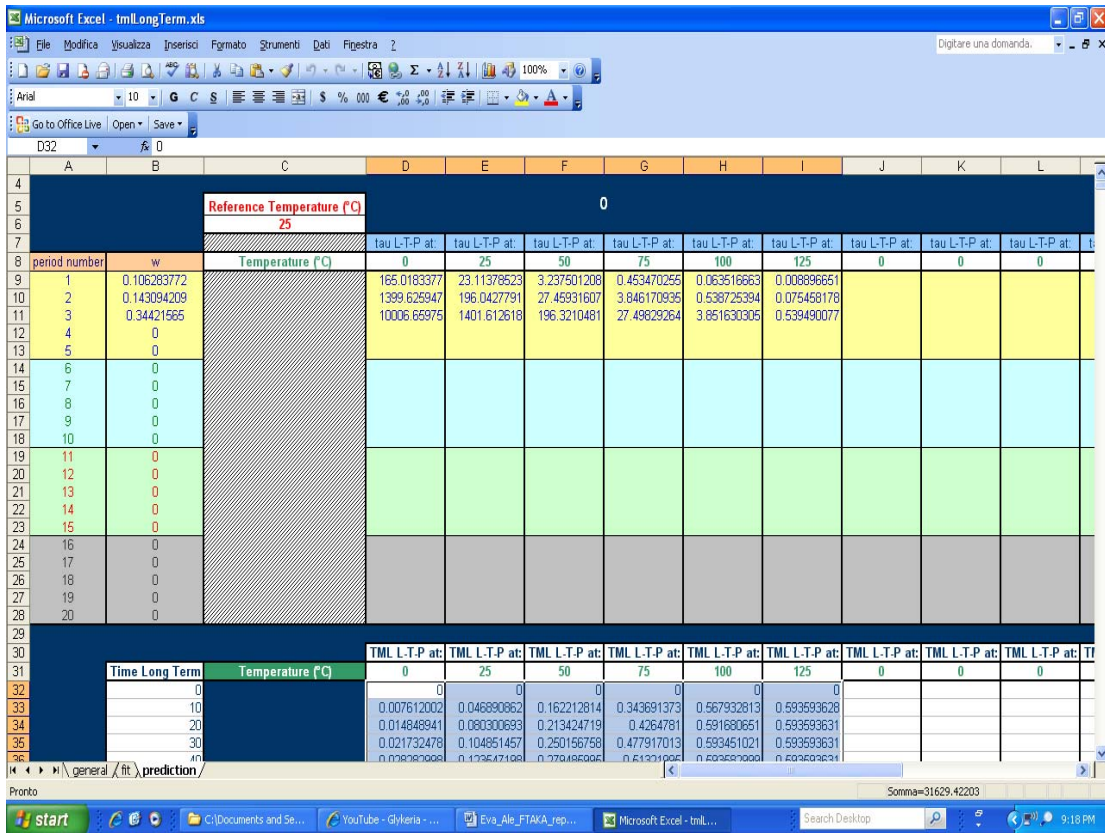
Τα αρχεία των μακροπρόθεσμων προβλέψεων περιλαμβάνουν κρίσιμες πληροφορίες της θεωρίας και αφορούν τα σημεία τομής (χρόνος, τιμή) των καμπυλών, τους παράγοντες επιτάχυνσης, τις ενέργειες ενεργοποίησης και τον συντελεστή  $K_e$  [εικόνα 6-4, για την tml].

Περιλαμβάνονται επίσης πληροφορίες που αφορούν την μακροπρόθεσμη προσαρμογή, δηλαδή οι επιταχυνθέντες χρόνοι παραμονής  $\tau$ , θεωρητικοί και προσαρμοσμένοι, και τα προσαρμοσμένα  $w$ , καθώς και όλα τα δεδομένα του πειράματος σε κάθε χρονική στιγμή με τα σχετικά σφάλματα [εικόνα 6-5, για την tml].

Τέλος, είναι διαθέσιμες οι προβλέψεις για την χρονική εξέλιξη του φαινομένου της διάχυσης και της συμπίκνωσης στους συλλέκτες σε κάθε θερμοκρασία ενδιαφέροντος, με συγκεντρωμένα τα  $\tau$ -διανύσματα για σύγκριση και περαιτέρω μελέτη [εικόνα 6-6, για την tml]. Συνεπώς, ο χρήστης έχει άμεση πρόσβαση σε όλα τα δεδομένα και όλα τα αποτελέσματα της επεξεργασίας από το MatLab®.



εικόνα 6- 5



εικόνα 6- 6



## 7. ΣΥΜΠΕΡΑΣΜΑΤΑ

Παρουσιάστηκε η βασική θεωρία διάχυσης και η μέθοδος προσδιορισμού και μελέτης της κινητικής του φαινομένου που ορίζεται στο ESTEC, συμπεριλαμβανομένης και της αναθεωρημένης σχετικής μαθηματικής επεξεργασίας.

Αναπτύχθηκε μια πρότυπη εφαρμογή στο MatLab® για την επεξεργασία των δεδομένων με τρόπο ευθύ και τυποποιημένο και τα βασικά βήματα της συνολικής διαδικασίας έχουν περιγραφεί. Κάποια από τα βήματα απαιτούν μεγαλύτερη προσοχή και η μελλοντική έρευνα πρέπει να επικεντρωθεί σε αυτά. Συγκεκριμένα, η επίδραση της θερμοκρασιακής μετάβασης σε πραγματικές συνθήκες στις τιμές των δεδομένων και η παραγωγή των 'ιδανικών' καμπυλών, δηλαδή εκείνων που θα προέκυπταν στην περίπτωση ιδανικής στιγμιαίας αύξησης της θερμοκρασίας, είναι τα σημαντικότερα σημεία της μαθηματικής επεξεργασίας που πρέπει να εστιάσουμε τις μελλοντικές προσπάθειές μας.

Στόχος είναι πάντα η εύρεση μιας μεθόδου παραγωγής των δεδομένων που θα προέκυπταν αν η μετάβαση μεταξύ των επιθυμητών θερμοκρασιών κάθε περιόδου μπορούσε να συμβεί αυτομάτως. Σε αυτή τη φάση, η αναγκαιότητα εύρεσης ικανοποιητικών και φυσικά αποδεκτών προσαρμογών οδήγησε σε ποικίλες μαθηματικές δοκιμές και, θεωρούμε, πως επιτεύχθηκε με τη χρήση της  $2^{15}$  παραγωγού της ονομαζόμενης αρχικής προσαρμογής για την προσομοίωση του φαινομένου στις αρχικές φάσεις εκάστης περιόδου.

Φυσικά, η σύγκριση μεταξύ μακροπρόθεσμων προβλέψεων και μακροχρόνιων μετρήσεων θεωρείται απαραίτητη για την τεκμηρίωση των υποθέσεων και της συνολικής προσέγγισης, καθώς και για την ανάλυση των σφαλμάτων.



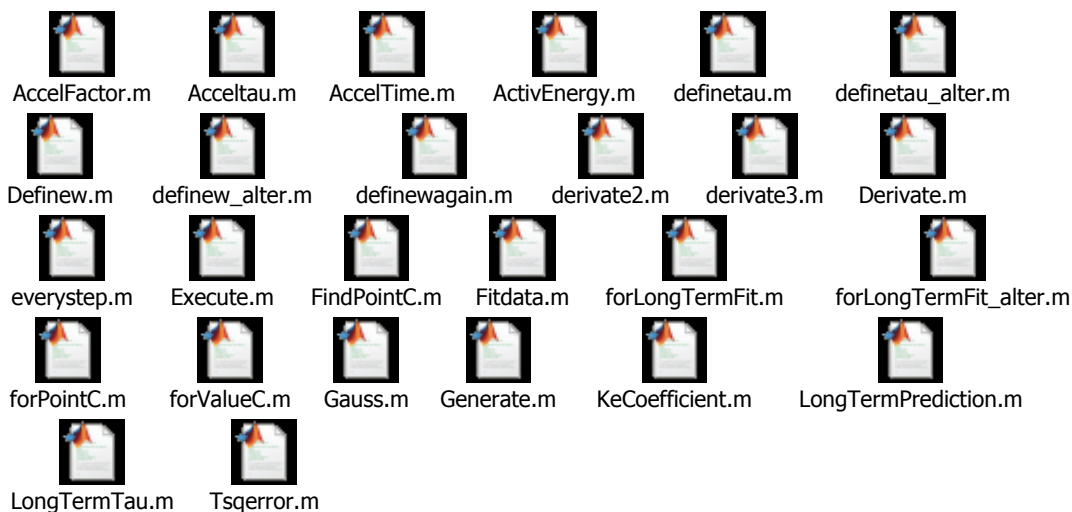
## 8. ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

- [1] Tribble A.C. , Boyadjian B., Davis J., Haffner J., McCullough E.  
Contamination Control Engineering Design Guidelines for the Aerospace Community, NASA CR 4740, May 1996
- [2] European Cooperation for Space Standardization, ECSS working group  
ECSS-E-ST-10-04C  
Space engineering – Space environment, Nov. 2008
- [3] European Cooperation for Space Standardization, ECSS working group  
ECSS-Q-70-02A  
Space Product Assurance – Thermal Vacuum Outgassing Test for the Screening of Space Materials, May 2000
- [4] Van Eesbeek M., Zwaal A.  
Outgassing and Contamination Model Based on Residence Time  
*Proceedings of the Third European Symposium on Spacecraft Materials in Space Environment*, ESA SP-232, Nov. 1985
- [5] Guyt J., Van Eesbeek M.  
QME-WI-F-03  
Standard Test Method for VBQC Outgassing Kinetics, Feb. 2000
- [6] Guyt J., Van Eesbeek M.  
QMC-WI-F-04  
Mathematical Treatment of VBQC Data, Feb. 2000
- [7] Rampini R., Ftaka E., Van Eesbeek M.  
Dynamical Outgassing Testing: A Revisited Mathematical Approach  
*International Symposium on Materials in a Space Environment*, Sept. 2009





## 9. ΚΩΔΙΚΕΣ MATLAB®



XX

### 9.1. AccelFactor.m

Υπολογίζει τον παράγοντα επιτάχυνσης μεταξύ δυο διαδοχικών χρονικών περιόδων.

```
function [k,dy1,dy2] = AccelFactor(time,t1,t2,w1,w2,taul,tau2);

if nargin==6, tau2=[50 20 5 1 0.5 0.1]'; end;
if nargin==5, tau2=[50 20 5 1 0.5 0.1]'; tau1=[50 20 5 1 0.5 0.1]';
end;

lenw1= length(w1);
lenw2= length(w2);
lenta1= length(taul);
lenta2= length(tau2);
if (lenw1 ~= lenta1) && (lenw2 ~= lenta2),
    disp ('vectors w and tau should be of equal size');
    return;
end;

a1 = zeros(lenw1,1);
a2 = zeros(lenw2,1);
b1 = zeros(lenw1,1);
b2 = zeros(lenw2,1);
dy1 = 0;
dy2 = 0;

for i=1:lenw1;
    a1(i)= exp(-(time-t1)/taul(i));
    b1(i)= (w1(i)/taul(i))*a1(i);
    dy1= sum(b1(:));
end;
```

```

for i=1:lenw2;
    a2(i)= exp(-(time-t2)/tau2(i));
    b2(i)= (w2(i)/tau2(i))*a2(i);
    dy2= sum(b2(:));
end;

```

```

k = dy2/dy1;

```

XX

## 9.2. AccelTau.m

Επιταχύνει το διάνυσμα  $\tau$  για κάθε φάση σύμφωνα με τους παράγοντες επιτάχυνσης.

```

function tau=accltau(phase,k);

lenphase=length(phase);
lenk=length(k);
if lenphase~=lenk+1,
    disp('incompatible number of phases and number of accelaration
factors: #phases=#k+1');
    return;
end;

prodk=zeros(1,lenk);
for i=1:lenk,
    prodk(i)=prod(k(1:i));
end;

tau=zeros(lenphase,1);
for i=2:lenphase,
    tau(1)=phase(1);
    tau(i)=phase(i)*prodk(i-1)+tau(i-1);
end;

```

XX

## 9.3 AccelTime.m

Επιταχύνει τον χρόνο διεξαγωγής του πειράματος σύμφωνα με τους παράγοντες επιτάχυνσης.

```

function acceleratedTime = AccelTime(TIME,j,k);

lenk=length(k);
prodk=zeros(1,lenk);
for i=1:lenk,
    prodk(i)=prod(k(1:i));
end;

```

```

Dj=diff(j);
[maxDj,postDj]=max(Dj);

time=zeros(maxDj+1,length(Dj));
Dtime=zeros(maxDj,length(Dj));
for i =1:length(Dj),
    time(1:j(i+1)-j(i)+1,i)=TIME(j(i):j(i+1));
    Dtime(1:j(i+1)-j(i),i)=diff(time(1:j(i+1)-j(i)+1,i));
end;

AccTime=zeros(maxDj+1,length(Dj));
for i=2:length(Dj),
    for k=2:j(i+1)-j(i)+1,
        AccTime(1,1)=time(1,1);
        AccTime(k,1)=time(k,1);
    end;
end;
for i=2:length(Dj),
    for k=2:j(i+1)-j(i)+1,
        AccTime(1,i)=0;
        AccTime(k,i)=sum(Dtime(1:k-1,i))*prodk(i-1)+AccTime(j(i)-j(i-1)+1,i-1);
    end;
end;

accelerTime=AccTime(find(AccTime));
acceleratedTime=[0;accelerTime];

```

XX

#### 9.4. ActivEnergy.m

Υπολογίζει την ενέργεια ενεργοποίησης μεταξύ δυο θερμοκρασιών.

```

function E = ActivEnergy(K,T1,T2,R);

if nargin==3, R=8.31; end;

if T1>T2,
    disp('The second temperature value inserted in the ActivEnergy
function should be greater than the first one');
    return;
end;

E= ((R*(T1+273.15)*(T2+273.15))/(T2-T1))*log(K)/1000;

```

XX

## 9.5. `definetau.m`

Βελτιστοποιεί το διάνυσμα  $\tau$  χωρίς περιορισμό μη αρνητικότητας.

```
function
[yout,err,terr,relerr,trelerr,acc,prec,taunew,exitflag]=definetau(t,y
,w0,tau0);

[taunew,fval,exitflag] = fminsearch(@(tau)
tsqerror(t,y,w0,tau),tau0);

lenw= length(w0);
lent= length(t);

a = ones(lenw,lent);
b = ones(lenw,lent);
yout = ones(1,lent);

err = ones(1,lent);
terr=0;
acc=0;

relerr = ones(1,lent);
trelerr=0;
prec=0;

for i=1:lenw;
    for j=1:lent;
        a(i,j)= 1-exp(-t(j)/taunew(i));
        b(i,j)= w0(i)*a(i,j);
        yout(j)= sum(b(1:lenw,j));
        err(j)= abs(y(j)-yout(j));
        terr= sum(err(:));
        acc= -log10(terr);
    end;
end;
for j=2:lent;
    relerr(j)= err(j)/abs(y(j));
    trelerr= sum(relerr(:));
    prec= -log10(trelerr);
end;
```

XX

## 9.6. `definetau_alter.m`

Βελτιστοποιεί το διάνυσμα  $\tau$  υπό περιορισμό μη αρνητικότητας.

```
function
[yout,err,terr,relerr,trelerr,acc,prec,taunew]=definetau_alter(t,y,w0
,tau0);
```

```

taunew = fmincon(@(tau) tsqerror(t,y,w0,tau),tau0,[],[],[],[],0,inf);

lenw= length(w0);
lent= length(t);

a = ones(lenw,lent);
b = ones(lenw,lent);
yout = ones(1,lent);

err = ones(1,lent);
terr=0;
acc=0;

relerr = ones(1,lent);
treterr=0;
prec=0;

for i=1:lenw;
    for j=1:lent;
        a(i,j)= 1-exp(-t(j)/taunew(i));
        b(i,j)= w0(i)*a(i,j);
        yout(j)= sum(b(1:lenw,j));
        err(j)= abs(y(j)-yout(j));
        terr= sum(err(:));
        acc= -log10(terr);
    end;
end;
for j=2:lent;
    relerr(j)= err(j)/abs(y(j));
    treterr= sum(relerr(:));
    prec= -log10(treterr);
end;

```

XX

## 9.7. definew.m

Προσαρμόζει τα δεδομένα προσδιορίζοντας το διάνυσμα  $w$  υπό περιορισμό μη αρνητικότητας.

```

function [yout,err,terr,relerr,treterr,acc,prec,w]=definew(t,y,tau);

if nargin==2, tau= [50 20 5 1 0.5 0.1]'; end;

lent= length(t);
leny = length(y);
if lent ~= leny, disp ('vectors t and y (tml or cvcm) should be of
equal size'); end;

d= length(tau);
w0= zeros(1,d)';
lenw= length(w0);

a = zeros(lenw,lent);

```

```

b = zeros(lenw,lent);
yout = zeros(1,lent);
err = zeros(1,lent);
D = zeros(lent,lenw);

for i=1:lenw;
    for j=1:lent;
        a(i,j)= 1-exp(-t(j)/tau(i));
        b(i,j)= w0(i)*a(i,j);
        yout(j)= sum(b(1:lenw,j));
        err(j)= y(j)-yout(j);
        D(j,i)= 1-exp(-t(j)/tau(i));
    end;
end;

errcol = err';
transD = D';
A = transD*D;
B = transD*errcol;

%status indicator [norm(A)*norm(inverse(A))]
%si = cond(A);

%dw = gauss(A,B);
wstart = zeros(d,1);
dw = lsqnonneg(A,B,wstart);
w = w0+dw;

terr=0;
relerr = zeros(1,lent);
treterr=0;
acc=0;
prec=0;
for i=1:lenw;
    for j=1:lent;
        a(i,j)= 1-exp(-t(j)/tau(i));
        b(i,j)= w(i)*a(i,j);
        yout(j)= sum(b(1:lenw,j));
        err(j)= abs(y(j)-yout(j));
        terr= sum(err(:));
        acc= -log10(terr);
    end;
end;
for j=2:lent;
    relerr(j)= err(j)/abs(y(j));
    treterr= sum(relerr(:));
    prec= -log10(treterr);
end;

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

### 9.8. `definew_alter.m`

Προσαρμόζει τα δεδομένα προσδιορίζοντας το διάνυσμα  $w$  υπό περιορισμό μη αρνητικότητας, με χρήση πινάκων.

```

function
[yout,err,terr,relerr,trelerr,acc,prec,w]=definew_alter(t,y,tau);

if nargin==2, tau= [50 20 5 1 0.5 0.1]'; end;

lent= length(t);
leny = length(y);
if lent ~= leny, disp ('vectors t and y (tml or cvcm) should be of
equal size'); end;

lenw= length(tau);
fee = zeros(lenw,lent);

for i=1:lenw;
    for j=1:lent;
        fee(i,j)= 1-exp(-t(j)/tau(i));
    end;
end;

transfee = fee';
A = fee*transfee;
b = fee*y;

%status indicator [norm(A)*norm(inverse(A))]
%si = cond(A);

%w = gauss(A,B);
w = lsqnonneg(A,b);

a= zeros(lenw,lent);
b= zeros(lenw,lent);
yout= zeros(1,lent);
err= zeros(1,lent);
terr=0;
relerr = zeros(1,lent);
trelerr=0;
acc=0;
prec=0;
for i=1:lenw;
    for j=1:lent;
        a(i,j)= 1-exp(-t(j)/tau(i));
        b(i,j)= w(i)*a(i,j);
        yout(j)= sum(b(1:lenw,j));
        err(j)= abs(y(j)-yout(j));
        terr= sum(err(:));
        acc= -log10(terr);
    end;
end;
for j=2:lent;
    relerr(j)= err(j)/abs(y(j));
    trelerr= sum(relerr(:));
    prec= -log10(trelerr);
end;

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

## 9.9. definewagain.m

Επαναπροσαρμόζει τα δεδομένα προσδιορίζοντας το διάνυσμα  $w$  υπό περιορισμό μη αρνητικότητας, με βάση ένα αρχικό διάνυσμα  $w_0$ .

```
function
[yout,err,terr,relerr,trelerr,acc,prec,w]=definewagain(t,y,w0,tau);

if nargin==3, tau= [50 20 5 1 0.5 0.1]'; end;
if nargin==2, tau= [50 20 5 1 0.5 0.1]'; w0= zeros(1,d)'; end;
d= length(tau);

lent= length(t);
leny = length(y);
if lent ~= leny, disp ('vectors t and y (tml or cvcm) should be of
equal size'); end;

lenw= length(w0);

a = ones(lenw,lent);
b = ones(lenw,lent);
yout = ones(1,lent);
err = ones(1,lent);
D = ones(lent,lenw);

for i=1:lenw;
    for j=1:lent;
        a(i,j)= 1-exp(-t(j)/tau(i));
        b(i,j)= (0.75*w0(i))*a(i,j);
        yout(j)= sum(b(1:lenw,j));
        err(j)= y(j)-yout(j);
        D(j,i)= 1-exp(-t(j)/tau(i));
    end;
end;

errcol = err';
transD = D';
A = transD*D;
B = transD*errcol;

%status indicator [norm(A)*norm(inverse(A))]
si = cond(A);

%dw = gauss(A,B);
dw = lsqnonneg(A,B);
w = 0.75*w0+dw;

terr=0;
relerr = zeros(1,lent);
trelerr=0;
acc=0;
prec=0;
for i=1:lenw;
    for j=1:lent;
        a(i,j)= 1-exp(-t(j)/tau(i));
        b(i,j)= w(i)*a(i,j);
        yout(j)= sum(b(1:lenw,j));
```



```

        err(j)= abs(y(j)-yout(j));
        terr= sum(err(:));
        acc= -log10(terr);
    end;
end;
for j=2:lent;
    relerr(j)= err(j)/abs(y(j));
    trelerr= sum(relerr(:));
    prec= -log10(trelerr);
end;
end;

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

### 9.10. derivate2.m

Υπολογίζει τη 2<sup>η</sup> παράγωγο μιας προσαρμογής.

```

function yout=derivate2(t,w,tau);

if nargin==2, tau= [50 20 5 1 0.5 0.1]'; end;

lent= length(t);

lenw= length(w);
lentaу= length(tau);
if lenw ~= lentaу, disp ('vectors w and tau should be of equal
size'); end;

a = ones(lenw,lent);
b = ones(lenw,lent);
yout = ones(1,lent);

for i=1:lenw;
    for j=1:lent;
        a(i,j)= exp(-t(j)/tau(i));
        b(i,j)= -(w(i)/(tau(i))^2)*a(i,j);
        yout(j)= sum(b(:,j));
    end;
end;

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

### 9.11. derivate3.m

Υπολογίζει την 3<sup>η</sup> παράγωγο μιας προσαρμογής.

```

function yout=derivate3(t,w,tau);

if nargin==2, tau= [50 20 5 1 0.5 0.1]'; end;

```

```

lent= length(t);

lenw= length(w);
lentau= length(tau);
if lenw ~= lentau, disp ('vectors w and tau should be of equal
size'); end;

a = ones(lenw,lent);
b = ones(lenw,lent);
yout = ones(1,lent);

for i=1:lenw;
    for j=1:lent;
        a(i,j)= exp(-t(j)/tau(i));
        b(i,j)= (w(i)/(tau(i))^3)*a(i,j);
        yout(j)= sum(b(:,j));
    end;
end;

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

## 9.12. derivate.m

Υπολογίζει την 1<sup>η</sup> παράγωγο μιας προσαρμογής.

```

function yout=derivate(t,w,tau);

if nargin==2, tau= [50 20 5 1 0.5 0.1]'; end;

lent= length(t);

lenw= length(w);
lentau= length(tau);
if lenw ~= lentau, disp ('vectors w and tau should be of equal
size'); end;

a = ones(lenw,lent);
b = ones(lenw,lent);
yout = ones(1,lent);

for i=1:lenw;
    for j=1:lent;
        a(i,j)= exp(-t(j)/tau(i));
        b(i,j)= (w(i)/tau(i))*a(i,j);
        yout(j)= sum(b(:,j));
    end;
end;

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

### 9.13. everystep.m

Για κάθε περίοδο, κάνει την αρχική προσαρμογή, λαμβάνει παραγώγους, εντοπίζει το σημείο αποκοπής, παράγει εικονικά δεδομένα και κάνει την τελική προσαρμογή.

```
function
[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoververtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoververtime,dyoverdtimeduo,timenew,ycou
nted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(time,y,tau);

timeadjusted=time-time(1);
yadjusted=y-y(1);
[yfitveryinitial,ERRORSforyinitial_,TotalErrorForyinitial,RELEERRORSfo
ryinitial_,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISIONfory
initial,Wforyinitial_]=fitdata(timeadjusted,yadjusted,tau);
yfitveryinitialunscaled_=yfitveryinitial+y(1);
%-----
% rows --> columns
yfitveryinitialunscaled = yfitveryinitialunscaled_';
ERRORSforyinitial = ERRORSforyinitial_';
RELEERRORSforyinitial = RELEERRORSforyinitial_';
% columns --> rows
Wforyinitial = Wforyinitial_';
%-----

yinitialduo=[yfitveryinitial'+y(1),y];

dyoververtimeinitial_=derivate(timeadjusted,Wforyinitial,tau);
d2yoverd2timeinitial_=derivate2(timeadjusted,Wforyinitial,tau);
d3yoverd3timeinitial_=derivate3(timeadjusted,Wforyinitial,tau);
%-----
% rows --> columns
dyoververtimeinitial = dyoververtimeinitial_';
d2yoverd2timeinitial = d2yoverd2timeinitial_';
d3yoverd3timeinitial = d3yoverd3timeinitial_';
%-----
[min_d2y,k_d2y]=min(d2yoverd2timeinitial);
[min_d3y,k_d3y]=min(abs(d3yoverd3timeinitial(1:(length(d3yoverd3timei
nitial)+1)/2)));
if k_d2y<=k_d3y,
    k_y=k_d2y;
else
    K_y=k_d3y;
end;

dyoververtime=zeros(length(timeadjusted),1);
ynew=zeros(length(yadjusted),1);
yinitnew=zeros(length(yfitveryinitial),1);
for i=1:k_y-1,
```

```

        dyoververtime(i)=dyoververtimeinitial(k_y)-
d2yoverd2timeinitial(k_y)*(timeadjusted(k_y)-timeadjusted(i));
        ynew(i)=yfitveryinitial(k_y)-dyoververtime(i)*(timeadjusted(k_y)-
timeadjusted(i));
        yinitnew(i)=yfitveryinitial(k_y)-
dyoververtime(i)*(timeadjusted(k_y)-timeadjusted(i));
end;
for i=k_y:length(timeadjusted),
    dyoververtime(i)=dyoververtimeinitial(i);
    ynew(i)=yadjusted(i);
    yinitnew(i)=yfitveryinitial(i);
end;

dyoverdtimeduo=[dyoververtimeinitial dyoververtime];

[value_ypositive,post_y]=max(ynew>=0);
if post_y==1,
    post_y=2;
end;

timeskipped=timeadjusted(1:post_y-2);
timeadjustedagain=timeadjusted(post_y-1:length(timeadjusted))-
timeadjusted(post_y-1);
yadjustedagain=ynew(post_y-1:length(ynew))-ynew(post_y-1);
[yfitinitial,ERRORSfory_,TotalErrorFory,RELEERRORSfory_,TotalRelErrorF
ory,ACCURACYfory,PRECISIONfory,Wfory_]=definew_alter(timeadjustedagai
n,yadjustedagain,tau);
%[yfitinitial,ERRORSfory_,TotalErrorFory,RELEERRORSfory_,TotalRelError
Fory,ACCURACYfory,PRECISIONfory,Wfory_]=definewagain(timeadjustedagai
n,yadjustedagain,Wforytest,tau);
yinitadjustedagain=yinitnew(post_y-1:length(yinitnew))-
yinitnew(post_y-1);
[yinitfitinitial,ERRORSforyinit_,TotalErrorForyinit,RELEERRORSforyini
t_,TotalRelErrorForyinit,ACCURACYforyinit,PRECISIONforyinit,Wforyinit_
]=definew_alter(timeadjustedagain,yinitadjustedagain,tau);
%[yinitfitinitial,ERRORSforyinit_,TotalErrorForyinit,RELEERRORSforyini
t_,TotalRelErrorForyinit,ACCURACYforyinit,PRECISIONforyinit,Wforyinit
_]=definewagain(timeadjustedagain,yinitadjustedagain,Wforyinittest,ta
u);
%-----
% rows --> columns
ERRORSfory = ERRORSfory_';
ERRORSforyinit = ERRORSforyinit_';
RELEERRORSfory = RELEERRORSfory_';
RELEERRORSforyinit = RELEERRORSforyinit_';
% columns --> rows
Wfory = Wfory_';
Wforyinit = Wforyinit_';
%-----

timenew=timeadjustedagain+timeadjusted(post_y-1)+time(1);
yfit_=yfitinitial+ynew(post_y-1)+y(1);
ycounted=yadjustedagain+ynew(post_y-1)+y(1);
yinitfit_=yinitfitinitial+yinitnew(post_y-1)+y(1);
yinitcounted=yinitadjustedagain+yinitnew(post_y-1)+y(1);
%-----
% rows --> columns
yfit = yfit_';
yinitfit = yinitfit_';
%-----

```

```

yfitfirst = yfit(1);
yfitlast = yfit(length(yfit));
yinitfitfirst = yinitfit(1);
yinitfitlast = yinitfit(length(yinitfit));

yduo=[yfit ycounted];
yinitduo=[yinitfit yinitcounted];

yadjustment=ynew(post_y-1)+y(1);
yinitadjustment=yinitnew(post_y-1)+y(1);
tadjustment=timeadjusted(post_y-1)+time(1);

timeextra = [timeadjustedagain'
timeadjusted(length(timeadjusted)):1.25*timeadjusted(length(timeadju
sted))];
yextra=generate(timeextra,Wfory,tau);
yinitextra=generate(timeextra,Wforyinit,tau);
timecomplete_=timeextra+timeadjusted(post_y-1)+time(1);
ycomplete_=yextra+ynew(post_y-1)+y(1);
yinitcomplete_=yinitextra+yinitnew(post_y-1)+y(1);
%-----
% rows --> columns
timecomplete = timecomplete_';
lentimecomplete = length(timecomplete_);
ycomplete = ycomplete_';
yinitcomplete = yinitcomplete_';
%-----

```

XX

#### 9.14. execute.m

Είναι η εντολή που εκτελεί όλη την ακολουθία του προγράμματος και είναι αυτή που πληκτρολογεί ο χρήστης.

```

clear all;
format long e;

disp('Welcome to the MatLab data-fitting program for the outgassing
tests');
disp('The procedure is based on the theory developed by');
disp('M. Van Eesbeek and A. Zwaal');
disp('"Outgassing and Contamination Model based on Residence Time"');
disp('-----
----');

a=0;
while (a ~= 1 && a ~= 2 && a ~= 3),
    a = input('How do you want the TML and CVCMS to be expressed as?
[1: % - 2: mg/cm2 - 3: ask me later] :');
end;
disp('-----
----');

```

```

b=0;
disp('Constrained optimization is part of the optimization toolbox');
disp('The method used is "fmincon.m". Unfortunately, it usually gets
trapped in local minima')
disp('because it is designed to work on problems where the objective
and constraint functions')
disp('are both continuous and have continuous first derivatives. ');
disp('Method "fminsearch.m" performs unconstrained optimization. ');
disp('It is suggested to use "fminsearch.m" - because it can often
handle discontinuity - ');
disp('unless you notice negative values in the tau vector. ');
while (b ~= 1 && b ~= 2 && b ~= 3),
    disp('NOTE:');
    disp('(YES - acceptable only if the optimization toolbox is
installed in your computer)');
    b = input('Do you want to use constrained optimization for the
tau? [1: YES - 2: NO - 3: ask me later] :');
end;
disp('-----
----');

c=0;
while (c ~= 1 && c ~= 2 && c ~= 3),
    c = input('Do you want to see the results in EXCEL files? [1: YES
- 2: NO - 3: ask me later] :');
    if c==1,
        disp('MATLAB will proceed to a basic transmission of
information to EXCEL files. ');
        disp('This means that you will be able to find all the
information about the initial fits, the "final" fits, ');
        disp('the intersection times, the acceleration factors, the
activation energies, the Ke coefficients, ');
        disp('the long-term fits and the long-term predictions in
EXCEL files. ');
        disp('Any intermediate or further information might be
obtained by the MATLAB via commands and possible calculations. ');
    end;
end;
disp('-----
----');

%-----

SampleName = xlsread('vbqc.xls','general','B1:B1');
mass = xlsread('vbqc.xls','general','B7:B7'); % sample mass in g
SampleMass = mass*1000; % sample mass in mg
area = xlsread('vbqc.xls','general','B8:B8');
Rgasconst = xlsread('vbqc.xls','general','B11:B11');

VF1 = xlsread('vbqc.xls','general','C17:C17');
VF2 = xlsread('vbqc.xls','general','C18:C18');
VF3 = xlsread('vbqc.xls','general','C19:C19');

sensTML = xlsread('vbqc.xls','general','D16:D16');
sens1 = xlsread('vbqc.xls','general','D17:D17');
sens2 = xlsread('vbqc.xls','general','D18:D18');
sens3 = xlsread('vbqc.xls','general','D19:D19');

measurements = xlsread('vbqc.xls','general','B21:B21');

```

```

periods = xlsread('vbqc.xls','general','B25:B25');
lentau = xlsread('vbqc.xls','general','H25:H25');

ActualTestStart = xlsread('vbqc.xls','general','A25:A25');
duration =
xlsread('vbqc.xls','general',strcat('D25:D',num2str(periods+24)));
TAU =
xlsread('vbqc.xls','general',strcat('G25:G',num2str(lentau+24)));
lenTAU = length(TAU);

Treference = xlsread('vbqc.xls','general','J22:J22');
lenTpre = xlsread('vbqc.xls','general','K25:K25');
Tpredictions =
xlsread('vbqc.xls','general',strcat('J25:J',num2str(lenTpre+24)));

%-----

time =
xlsread('vbqc.xls','data',strcat('A2:A',num2str(measurements+1)));
Balance =
xlsread('vbqc.xls','data',strcat('B2:B',num2str(measurements+1)));

pressure =
xlsread('vbqc.xls','data',strcat('C2:C',num2str(measurements+1)));

TemperatureQCM1 =
xlsread('vbqc.xls','data',strcat('D2:D',num2str(measurements+1)));
FrequencyQCM1 =
xlsread('vbqc.xls','data',strcat('E2:E',num2str(measurements+1)));
TemperatureQCM2 =
xlsread('vbqc.xls','data',strcat('F2:F',num2str(measurements+1)));
FrequencyQCM2 =
xlsread('vbqc.xls','data',strcat('G2:G',num2str(measurements+1)));
TemperatureQCM3 =
xlsread('vbqc.xls','data',strcat('H2:H',num2str(measurements+1)));
FrequencyQCM3 =
xlsread('vbqc.xls','data',strcat('I2:I',num2str(measurements+1)));

TemperatureShroud =
xlsread('vbqc.xls','data',strcat('J2:J',num2str(measurements+1)));
TemperatureHeatingPlate =
xlsread('vbqc.xls','data',strcat('K2:K',num2str(measurements+1)));
TemperatureSample =
xlsread('vbqc.xls','data',strcat('L2:L',num2str(measurements+1)));
TemperatureQCMplate =
xlsread('vbqc.xls','data',strcat('M2:M',num2str(measurements+1)));
TemperatureColdRing =
xlsread('vbqc.xls','data',strcat('N2:N',num2str(measurements+1)));
TemperatureExternalHeater =
xlsread('vbqc.xls','data',strcat('O2:O',num2str(measurements+1)));

%-----

SumDuration = zeros(1,periods);
for i=1:periods,
    SumDuration(i) = sum(duration(1:i));
end;

maxim = zeros(1,periods+1);

```

```

position = zeros(1,periods+1);
for i=2:periods+1,
    [maxim(1),position(1)] = max(time==ActualTestStart);
    [maxim(i),position(i)] = max(time==SumDuration(i-
1)+ActualTestStart);
end;

%-----

temperatureQCM1 = TemperatureQCM1(position(1):position(periods+1));
temperatureQCM2 = TemperatureQCM2(position(1):position(periods+1));
temperatureQCM3 = TemperatureQCM3(position(1):position(periods+1));

temperatureShroud =
TemperatureShroud(position(1):position(periods+1));
temperatureHeatingPlate =
TemperatureHeatingPlate(position(1):position(periods+1));
temperatureSample =
TemperatureSample(position(1):position(periods+1));
temperatureQCMplate =
TemperatureQCMplate(position(1):position(periods+1));
temperatureColdRing =
TemperatureColdRing(position(1):position(periods+1));

meantemperatureQCM1 = mean(temperatureQCM1);
meantemperatureQCM2 = mean(temperatureQCM2);
meantemperatureQCM3 = mean(temperatureQCM3);

Dposition = diff(position);
[maxDposition,postDposition] = max(Dposition);

PeriodTemperature = zeros(maxDposition+1,periods);
for i=1:periods,
    PeriodTemperature(1:position(i+1)-position(i)+1,i) =
TemperatureShroud(position(i):position(i+1));
end;

AveragePeriodTemperature = zeros(periods,1);
for i=1:periods,

AveragePeriodTemperature(i)=mean(PeriodTemperature(1:position(i+1)-
position(i)+1,i));
end;

%-----

Time = time(position(1):position(periods+1))-time(position(1));

MassBalance = Balance(position(1):position(periods+1));

frequencyQCM1 = FrequencyQCM1(position(1):position(periods+1));
frequencyQCM2 = FrequencyQCM2(position(1):position(periods+1));
frequencyQCM3 = FrequencyQCM3(position(1):position(periods+1));

if a==3,
    while (a ~= 1 && a ~= 2),
        a = input('How do you want the TML and CVCMS to be expressed
as? [1: % - 2: mg/cm2] :');
    end;

```



```

end;

if a==1,
    tml = ((MassBalance-MassBalance(1))./SampleMass)*100;
    cvcm1 = (((frequencyQCM1-
frequencyQCM1(1))*sens1*VF1)./SampleMass)*100;
    cvcm2 = (((frequencyQCM2-
frequencyQCM2(1))*sens2*VF2)./SampleMass)*100;
    cvcm3 = (((frequencyQCM3-
frequencyQCM3(1))*sens3*VF3)./SampleMass)*100;
    symbolismos = '%';
elseif a==2,
    tml = (MassBalance-MassBalance(1))./area;
    cvcm1 = (((frequencyQCM1-
frequencyQCM1(1))*sens1*VF1)./area)*1000;
    cvcm2 = (((frequencyQCM2-
frequencyQCM2(1))*sens2*VF2)./area)*1000;
    cvcm3 = (((frequencyQCM3-
frequencyQCM3(1))*sens3*VF3)./area)*1000;
    symbolismos = 'mg/cm2';
end;

%-----

blah = zeros(1,periods+1);
PostForAverage = zeros(1:periods+1);
for i=2:periods+1,
    [blah(i),PostForAverage(i)] = max(Time==SumDuration(i-1)-1);
end;

%-----

maximadjusted = zeros(1,periods+1);
positionadjusted = zeros(1,periods+1);
for i=2:periods+1,
    [maximadjusted(1),positionadjusted(1)] = max(Time==0);
    [maximadjusted(i),positionadjusted(i)] = max(Time==SumDuration(i-
1));
end;

TIME = zeros(maxDposition+1,periods);
TML = zeros(maxDposition+1,periods);
CVCM1 = zeros(maxDposition+1,periods);
CVCM2 = zeros(maxDposition+1,periods);
CVCM3 = zeros(maxDposition+1,periods);
for i=1:periods,
    TIME(1:positionadjusted(i+1)-positionadjusted(i)+1,i) =
Time(positionadjusted(i):positionadjusted(i+1));
    TML(1:positionadjusted(i+1)-positionadjusted(i)+1,i) =
tml(positionadjusted(i):positionadjusted(i+1));
    CVCM1(1:positionadjusted(i+1)-positionadjusted(i)+1,i) =
cvcm1(positionadjusted(i):positionadjusted(i+1));
    CVCM2(1:positionadjusted(i+1)-positionadjusted(i)+1,i) =
cvcm2(positionadjusted(i):positionadjusted(i+1));
    CVCM3(1:positionadjusted(i+1)-positionadjusted(i)+1,i) =
cvcm3(positionadjusted(i):positionadjusted(i+1));
end;

%-----
%-----

```

⌘-----  
⌘-----  
⌘-----

```
TMLfitveryinitialunscaled=0;  
ERRORSforTMLinitial=0;  
TotalErrorForTMLinitial=0;  
RELEERRORSforTMLinitial=0;  
TotalRelErrorForTMLinitial=0;  
ACCURACYforTMLinitial=0;  
PRECISIONforTMLinitial=0;  
WforTMLinitial=0;
```

```
TMLinitialduo=0;
```

```
dTMLoverdtimeinitial=0;  
d2TMLoverd2timeinitial=0;  
d3TMLoverd3timeinitial=0;  
dTMLoverdtime=0;
```

```
dTMLoverdtimeduo=0;
```

```
timenewTML=0;  
TMLcounted=0;  
TMLinitcounted=0;
```

```
TMLfit=0;  
ERRORSforTML=0;  
TotalErrorForTML=0;  
RELEERRORSforTML=0;  
TotalRelErrorForTML=0;  
ACCURACYforTML=0;  
PRECISIONforTML=0;  
WforTML=0;
```

```
TMLinitfit=0;  
ERRORSforTMLinit=0;  
TotalErrorForTMLinit=0;  
RELEERRORSforTMLinit=0;  
TotalRelErrorForTMLinit=0;  
ACCURACYforTMLinit=0;  
PRECISIONforTMLinit=0;  
WforTMLinit=0;
```

```
TMLduo=0;  
TMLinitduo=0;
```

```
timecompleteTML=0;  
lentimecompleteTML=0;  
TMLcomplete=0;  
TMLinitcomplete=0;
```

```
tadjustmentTML=0;  
TMLadjustment=0;  
TMLinitadjustment=0;
```

```
TMLfitfirst=0;  
TMLfitlast=0;  
TMLinitfitfirst=0;
```

```

TMLinitfitlast=0;
for i=1:periods,

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoververtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoververtime,dyoverdtimeduo,timenew,yco
unted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TML(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU);
    eval([genvarname('TMLfitveryinitialunscaled', who)
'=yfitveryinitialunscaled']);
    eval([genvarname('ERRORSforTMLinitial', who)
'=ERRORSforyinitial']);
    eval([genvarname('TotalErrorForTMLinitial', who)
'=TotalErrorForyinitial']);
    eval([genvarname('RELEERRORSforTMLinitial', who)
'=RELEERRORSforyinitial']);
    eval([genvarname('TotalRelErrorForTMLinitial', who)
'=TotalRelErrorForyinitial']);
    eval([genvarname('ACCURACYforTMLinitial', who)
'=ACCURACYforyinitial']);
    eval([genvarname('PRECISIONforTMLinitial', who)
'=PRECISIONforyinitial']);
    eval([genvarname('WforTMLinitial', who) '=Wforyinitial']);

    eval([genvarname('TMLinitialduo', who) '=yinitialduo']);

    eval([genvarname('dTMloververtimeinitial', who)
'=dyoververtimeinitial']);
    eval([genvarname('d2TMloverd2timeinitial', who)
'=d2yoverd2timeinitial']);
    eval([genvarname('d3TMloverd3timeinitial', who)
'=d3yoverd3timeinitial']);
    eval([genvarname('dTMloververtime', who) '=dyoververtime']);

    eval([genvarname('dTMloverdtimeduo', who) '=dyoverdtimeduo']);

    eval([genvarname('timenewTML', who) '=timenew']);
    eval([genvarname('TMLcounted', who) '=ycounted']);
    eval([genvarname('TMLinitcounted', who) '=yinitcounted']);

    eval([genvarname('TMLfit', who) '=yfit']);
    eval([genvarname('ERRORSforTML', who) '=ERRORSfory']);
    eval([genvarname('TotalErrorForTML', who) '=TotalErrorFory']);
    eval([genvarname('RELEERRORSforTML', who) '=RELEERRORSfory']);
    eval([genvarname('TotalRelErrorForTML', who)
'=TotalRelErrorFory']);
    eval([genvarname('ACCURACYforTML', who) '=ACCURACYfory']);
    eval([genvarname('PRECISIONforTML', who) '=PRECISIONfory']);
    eval([genvarname('WforTML', who) '=Wfory']);

    eval([genvarname('TMLinitfit', who) '=yinitfit']);
    eval([genvarname('ERRORSforTMLinit', who) '=ERRORSforyinit']);

```

```

    eval([genvarname('TotalErrorForTMLinit', who)
    '=TotalErrorForyinit' ]);
    eval([genvarname('RELEERRORSforTMLinit', who)
    '=RELEERRORSforyinit' ]);
    eval([genvarname('TotalRelErrorForTMLinit', who)
    '=TotalRelErrorForyinit' ]);
    eval([genvarname('ACCURACYforTMLinit', who)
    '=ACCURACYforyinit' ]);
    eval([genvarname('PRECISIONforTMLinit', who)
    '=PRECISIONforyinit' ]);
    eval([genvarname('WforTMLinit', who) '=Wforyinit' ]);

    eval([genvarname('TMLduo', who) '=yduo' ]);
    eval([genvarname('TMLinitduo', who) '=yinitduo' ]);

    eval([genvarname('timecompleteTML', who) '=timecomplete' ]);
    eval([genvarname('lentimecompleteTML', who) '=lentimecomplete' ]);
    eval([genvarname('TMLcomplete', who) '=ycomplete' ]);
    eval([genvarname('TMLinitcomplete', who) '=yinitcomplete' ]);

    eval([genvarname('tadjustmentTML', who) '=tadjustment' ]);
    eval([genvarname('TMLadjustment', who) '=yadjustment' ]);
    eval([genvarname('TMLinitadjustment', who) '=yinitadjustment' ]);

    eval([genvarname('TMLfitfirst', who) '=yfitfirst' ]);
    eval([genvarname('TMLfitlast', who) '=yfitlast' ]);
    eval([genvarname('TMLinitfitfirst', who) '=yinitfitfirst' ]);
    eval([genvarname('TMLinitfitlast', who) '=yinitfitlast' ]);
end;

lentimecompleteTML_vector = zeros(1,periods);
for i=1:periods,
    lentimecompleteTML_vector(i) = eval(strcat('lentimecompleteTML',
num2str(i)));
end;
maxlentimecompleteTML = max(lentimecompleteTML_vector);

timecomplete_TML = zeros(maxlentimecompleteTML,periods);
for i=1:periods,
    timecomplete_TML(1:lentimecompleteTML_vector(i),i) =
eval(strcat('timecompleteTML', num2str(i)));
end;

%-----

CVCM1fitveryinitialunscaled=0;
ERRORSforCVCM1initial=0;
TotalErrorForCVCM1initial=0;
RELEERRORSforCVCM1initial=0;
TotalRelErrorForCVCM1initial=0;
ACCURACYforCVCM1initial=0;
PRECISIONforCVCM1initial=0;
WforCVCM1initial=0;

CVCM1initialduo=0;

dCVCM1overdtimeinitial=0;
d2CVCM1overd2timeinitial=0;
d3CVCM1overd3timeinitial=0;

```

```

dCVCMLoverdtime=0;

dCVCMLoverdtimeduo=0;

timenewCVCML=0;
CVCMLcounted=0;
CVCMLinitcounted=0;

CVCMLfit=0;
ERRORSforCVCML=0;
TotalErrorForCVCML=0;
RELEERRORSforCVCML=0;
TotalRelErrorForCVCML=0;
ACCURACYforCVCML=0;
PRECISIONforCVCML=0;
WforCVCML=0;

CVCMLinitfit=0;
ERRORSforCVCMLinit=0;
TotalErrorForCVCMLinit=0;
RELEERRORSforCVCMLinit=0;
TotalRelErrorForCVCMLinit=0;
ACCURACYforCVCMLinit=0;
PRECISIONforCVCMLinit=0;
WforCVCMLinit=0;

CVCMLduo=0;
CVCMLinitduo=0;

timecompleteCVCML=0;
lentimecompleteCVCML=0;
CVCMLcomplete=0;
CVCMLinitcomplete=0;

tadjustmentCVCML=0;
CVCMLadjustment=0;
CVCMLinitadjustment=0;

CVCMLfitfirst=0;
CVCMLfitlast=0;
CVCMLinitfitfirst=0;
CVCMLinitfitlast=0;
for i=1:periods,

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoverdtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoverdtime,dyoverdtimeduo,timenew,yc
ounted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),CVCML(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU);
    eval([genvarname('CVCMLfitveryinitialunscaled', who)
'='yfitveryinitialunscaled']);

```

```

eval([genvarname('ERRORSforCVCMLinitial', who)
'=ERRORSforyinitial']);
eval([genvarname('TotalErrorForCVCMLinitial', who)
'=TotalErrorForyinitial']);
eval([genvarname('RELEERRORSforCVCMLinitial', who)
'=RELEERRORSforyinitial']);
eval([genvarname('TotalRelErrorForCVCMLinitial', who)
'=TotalRelErrorForyinitial']);
eval([genvarname('ACCURACYforCVCMLinitial', who)
'=ACCURACYforyinitial']);
eval([genvarname('PRECISIONforCVCMLinitial', who)
'=PRECISIONforyinitial']);
eval([genvarname('WforCVCMLinitial', who) '=Wforyinitial']);

eval([genvarname('CVCMLinitialduo', who) '=yinitialduo']);

eval([genvarname('dCVCMLoververtimeinitial', who)
'=dyoververtimeinitial']);
eval([genvarname('d2CVCMLoverd2timeinitial', who)
'=d2yoverd2timeinitial']);
eval([genvarname('d3CVCMLoverd3timeinitial', who)
'=d3yoverd3timeinitial']);
eval([genvarname('dCVCMLoververtime', who) '=dyoververtime']);

eval([genvarname('dCVCMLoverdtimeduo', who) '=dyoverdtimeduo']);

eval([genvarname('timenewCVCML', who) '=timenew']);
eval([genvarname('CVCMLcounted', who) '=ycounted']);
eval([genvarname('CVCMLinitcounted', who) '=yinitcounted']);

eval([genvarname('CVCMLfit', who) '=yfit']);
eval([genvarname('ERRORSforCVCML', who) '=ERRORSfory']);
eval([genvarname('TotalErrorForCVCML', who) '=TotalErrorFory']);
eval([genvarname('RELEERRORSforCVCML', who) '=RELEERRORSfory']);
eval([genvarname('TotalRelErrorForCVCML', who)
'=TotalRelErrorFory']);
eval([genvarname('ACCURACYforCVCML', who) '=ACCURACYfory']);
eval([genvarname('PRECISIONforCVCML', who) '=PRECISIONfory']);
eval([genvarname('WforCVCML', who) '=Wfory']);

eval([genvarname('CVCMLinitfit', who) '=yinitfit']);
eval([genvarname('ERRORSforCVCMLinit', who) '=ERRORSforyinit']);
eval([genvarname('TotalErrorForCVCMLinit', who)
'=TotalErrorForyinit']);
eval([genvarname('RELEERRORSforCVCMLinit', who)
'=RELEERRORSforyinit']);
eval([genvarname('TotalRelErrorForCVCMLinit', who)
'=TotalRelErrorForyinit']);
eval([genvarname('ACCURACYforCVCMLinit', who)
'=ACCURACYforyinit']);
eval([genvarname('PRECISIONforCVCMLinit', who)
'=PRECISIONforyinit']);
eval([genvarname('WforCVCMLinit', who) '=Wforyinit']);

eval([genvarname('CVCMLduo', who) '=yduo']);
eval([genvarname('CVCMLinitduo', who) '=yinitduo']);

eval([genvarname('timecompleteCVCML', who) '=timecomplete']);
eval([genvarname('lentimecompleteCVCML', who)
'=lentimecomplete']);

```

```

eval([genvarname('CVCM1complete', who) '=ycomplete']);
eval([genvarname('CVCM1initcomplete', who) '=yinitcomplete']);

eval([genvarname('tadjustmentCVCM1', who) '=tadjustment']);
eval([genvarname('CVCM1adjustment', who) '=yadjustment']);
eval([genvarname('CVCM1initadjustment', who)
'=yinitadjustment']);

eval([genvarname('CVCM1fitfirst', who) '=yfitfirst']);
eval([genvarname('CVCM1fitlast', who) '=yfitlast']);
eval([genvarname('CVCM1initfitfirst', who) '=yinitfitfirst']);
eval([genvarname('CVCM1initfitlast', who) '=yinitfitlast']);
end;

lentimecompleteCVCM1_vector = zeros(1,periods);
for i=1:periods,
    lentimecompleteCVCM1_vector(i) =
eval(strcat('lentimecompleteCVCM1', num2str(i)));
end;
maxlentimecompleteCVCM1 = max(lentimecompleteCVCM1_vector);

timecomplete_CVCM1 = zeros(maxlentimecompleteCVCM1,periods);
for i=1:periods,
    timecomplete_CVCM1(1:lentimecompleteCVCM1_vector(i),i) =
eval(strcat('timecompleteCVCM1', num2str(i)));
end;

%-----

CVCM2fitveryinitialunscaled=0;
ERRORSforCVCM2initial=0;
TotalErrorForCVCM2initial=0;
RELERRORSforCVCM2initial=0;
TotalRelErrorForCVCM2initial=0;
ACCURACYforCVCM2initial=0;
PRECISIONforCVCM2initial=0;
WforCVCM2initial=0;

CVCM2initialduo=0;

dCVCM2overdtimeinitial=0;
d2CVCM2overd2timeinitial=0;
d3CVCM2overd3timeinitial=0;
dCVCM2overdtime=0;

dCVCM2overdtimeduo=0;

timenewCVCM2=0;
CVCM2counted=0;
CVCM2initcounted=0;

CVCM2fit=0;
ERRORSforCVCM2=0;
TotalErrorForCVCM2=0;
RELERRORSforCVCM2=0;
TotalRelErrorForCVCM2=0;
ACCURACYforCVCM2=0;
PRECISIONforCVCM2=0;
WforCVCM2=0;

```

```

CVCM2initfit=0;
ERRORSforCVCM2init=0;
TotalErrorForCVCM2init=0;
RELEERRORSforCVCM2init=0;
TotalRelErrorForCVCM2init=0;
ACCURACYforCVCM2init=0;
PRECISIONforCVCM2init=0;
WforCVCM2init=0;

CVCM2duo=0;
CVCM2initduo=0;

timecompleteCVCM2=0;
lentimecompleteCVCM2=0;
CVCM2complete=0;
CVCM2initcomplete=0;

tadjustmentCVCM2=0;
CVCM2adjustment=0;
CVCM2initadjustment=0;

CVCM2fitfirst=0;
CVCM2fitlast=0;
CVCM2initfitfirst=0;
CVCM2initfitlast=0;
for i=1:periods,

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoverdtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoverdtime,dyoverdtimeduo,timenew,yco
unted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),CVCM2(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU);
    eval([genvarname('CVCM2fitveryinitialunscaled', who)
'='yfitveryinitialunscaled']);
    eval([genvarname('ERRORSforCVCM2initial', who)
'='ERRORSforyinitial']);
    eval([genvarname('TotalErrorForCVCM2initial', who)
'='TotalErrorForyinitial']);
    eval([genvarname('RELEERRORSforCVCM2initial', who)
'='RELEERRORSforyinitial']);
    eval([genvarname('TotalRelErrorForCVCM2initial', who)
'='TotalRelErrorForyinitial']);
    eval([genvarname('ACCURACYforCVCM2initial', who)
'='ACCURACYforyinitial']);
    eval([genvarname('PRECISIONforCVCM2initial', who)
'='PRECISIONforyinitial']);
    eval([genvarname('WforCVCM2initial', who) '='Wforyinitial']);

    eval([genvarname('CVCM2initialduo', who) '='yinitialduo']);

```



```

eval([genvarname('dCVC2oververtimeinitial', who)
'=dyoververtimeinitial' ]);
eval([genvarname('d2CVC2overd2timeinitial', who)
'=d2yoverd2timeinitial' ]);
eval([genvarname('d3CVC2overd3timeinitial', who)
'=d3yoverd3timeinitial' ]);
eval([genvarname('dCVC2oververtime', who) '=dyoververtime' ]);

eval([genvarname('dCVC2overdtime', who) '=dyoverdtime' ]);

eval([genvarname('timenewCVC2', who) '=timenew' ]);
eval([genvarname('CVC2counted', who) '=ycounted' ]);
eval([genvarname('CVC2initcounted', who) '=yinitcounted' ]);

eval([genvarname('CVC2fit', who) '=yfit' ]);
eval([genvarname('ERRORSforCVC2', who) '=ERRORSfory' ]);
eval([genvarname('TotalErrorForCVC2', who) '=TotalErrorFory' ]);
eval([genvarname('RELEERRORSforCVC2', who) '=RELEERRORSfory' ]);
eval([genvarname('TotalRelErrorForCVC2', who)
'=TotalRelErrorFory' ]);
eval([genvarname('ACCURACYforCVC2', who) '=ACCURACYfory' ]);
eval([genvarname('PRECISIONforCVC2', who) '=PRECISIONfory' ]);
eval([genvarname('WforCVC2', who) '=Wfory' ]);

eval([genvarname('CVC2initfit', who) '=yinitfit' ]);
eval([genvarname('ERRORSforCVC2init', who) '=ERRORSforyinit' ]);
eval([genvarname('TotalErrorForCVC2init', who)
'=TotalErrorForyinit' ]);
eval([genvarname('RELEERRORSforCVC2init', who)
'=RELEERRORSforyinit' ]);
eval([genvarname('TotalRelErrorForCVC2init', who)
'=TotalRelErrorForyinit' ]);
eval([genvarname('ACCURACYforCVC2init', who)
'=ACCURACYforyinit' ]);
eval([genvarname('PRECISIONforCVC2init', who)
'=PRECISIONforyinit' ]);
eval([genvarname('WforCVC2init', who) '=Wforyinit' ]);

eval([genvarname('CVC2duo', who) '=yduo' ]);
eval([genvarname('CVC2initduo', who) '=yinitduo' ]);

eval([genvarname('timecompleteCVC2', who) '=timecomplete' ]);
eval([genvarname('lentimecompleteCVC2', who)
'=lentimecomplete' ]);
eval([genvarname('CVC2complete', who) '=ycomplete' ]);
eval([genvarname('CVC2initcomplete', who) '=yinitcomplete' ]);

eval([genvarname('tadjustmentCVC2', who) '=tadjustment' ]);
eval([genvarname('CVC2adjustment', who) '=yadjustment' ]);
eval([genvarname('CVC2initadjustment', who)
'=yinitadjustment' ]);

eval([genvarname('CVC2fitfirst', who) '=yfitfirst' ]);
eval([genvarname('CVC2fitlast', who) '=yfitlast' ]);
eval([genvarname('CVC2initfitfirst', who) '=yinitfitfirst' ]);
eval([genvarname('CVC2initfitlast', who) '=yinitfitlast' ]);
end;

lentimecompleteCVC2_vector = zeros(1,periods);

```

```

for i=1:periods,
    lentimecompleteCVCM2_vector(i) =
eval(strcat('lentimecompleteCVCM2', num2str(i)));
end;
maxlentimecompleteCVCM2 = max(lentimecompleteCVCM2_vector);

timecomplete_CVCM2 = zeros(maxlentimecompleteCVCM2,periods);
for i=1:periods,
    timecomplete_CVCM2(1:lentimecompleteCVCM2_vector(i),i) =
eval(strcat('timecompleteCVCM2', num2str(i)));
end;

%-----

CVCM3fitveryinitialunscaled=0;
ERRORSforCVCM3initial=0;
TotalErrorForCVCM3initial=0;
RELEERRORSforCVCM3initial=0;
TotalRelErrorForCVCM3initial=0;
ACCURACYforCVCM3initial=0;
PRECISIONforCVCM3initial=0;
WforCVCM3initial=0;

CVCM3initialduo=0;

dCVCM3oververtimeinitial=0;
d2CVCM3overd2timeinitial=0;
d3CVCM3overd3timeinitial=0;
dCVCM3oververtime=0;

dCVCM3overdtimeduo=0;

timenewCVCM3=0;
CVCM3counted=0;
CVCM3initcounted=0;

CVCM3fit=0;
ERRORSforCVCM3=0;
TotalErrorForCVCM3=0;
RELEERRORSforCVCM3=0;
TotalRelErrorForCVCM3=0;
ACCURACYforCVCM3=0;
PRECISIONforCVCM3=0;
WforCVCM3=0;

CVCM3initfit=0;
ERRORSforCVCM3init=0;
TotalErrorForCVCM3init=0;
RELEERRORSforCVCM3init=0;
TotalRelErrorForCVCM3init=0;
ACCURACYforCVCM3init=0;
PRECISIONforCVCM3init=0;
WforCVCM3init=0;

CVCM3duo=0;
CVCM3initduo=0;

timecompleteCVCM3=0;
lentimecompleteCVCM3=0;

```

```

CVCM3complete=0;
CVCM3initcomplete=0;

tadjustmentCVCM3=0;
CVCM3adjustment=0;
CVCM3initadjustment=0;

CVCM3fitfirst=0;
CVCM3fitlast=0;
CVCM3initfitfirst=0;
CVCM3initfitlast=0;
for i=1:periods,

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoververtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoververtime,dyoverdtimeduo,timenew,yco
unted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),CVCM3(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU);
    eval([genvarname('CVCM3fitveryinitialunscaled', who)
'=yfitveryinitialunscaled']);
    eval([genvarname('ERRORSforCVCM3initial', who)
'=ERRORSforyinitial']);
    eval([genvarname('TotalErrorForCVCM3initial', who)
'=TotalErrorForyinitial']);
    eval([genvarname('RELEERRORSforCVCM3initial', who)
'=RELEERRORSforyinitial']);
    eval([genvarname('TotalRelErrorForCVCM3initial', who)
'=TotalRelErrorForyinitial']);
    eval([genvarname('ACCURACYforCVCM3initial', who)
'=ACCURACYforyinitial']);
    eval([genvarname('PRECISIONforCVCM3initial', who)
'=PRECISIONforyinitial']);
    eval([genvarname('WforCVCM3initial', who) '=Wforyinitial']);

    eval([genvarname('CVCM3initialduo', who) '=yinitialduo']);

    eval([genvarname('dCVCM3oververtimeinitial', who)
'=dyoververtimeinitial']);
    eval([genvarname('d2CVCM3overd2timeinitial', who)
'=d2yoverd2timeinitial']);
    eval([genvarname('d3CVCM3overd3timeinitial', who)
'=d3yoverd3timeinitial']);
    eval([genvarname('dCVCM3oververtime', who) '=dyoververtime']);

    eval([genvarname('dCVCM3overdtimeduo', who) '=dyoverdtimeduo']);

    eval([genvarname('timenewCVCM3', who) '=timenew']);
    eval([genvarname('CVCM3counted', who) '=ycounted']);
    eval([genvarname('CVCM3initcounted', who) '=yinitcounted']);

    eval([genvarname('CVCM3fit', who) '=yfit']);
    eval([genvarname('ERRORSforCVCM3', who) '=ERRORSfory']);

```

```

eval([genvarname('TotalErrorForCVCM3', who) '=TotalErrorFory']);
eval([genvarname('RELEERRORSforCVCM3', who) '=RELEERRORSfory']);
eval([genvarname('TotalRelErrorForCVCM3', who)
'=TotalRelErrorFory']);
eval([genvarname('ACCURACYforCVCM3', who) '=ACCURACYfory']);
eval([genvarname('PRECISIONforCVCM3', who) '=PRECISIONfory']);
eval([genvarname('WforCVCM3', who) '=Wfory']);

eval([genvarname('CVCM3initfit', who) '=yinitfit']);
eval([genvarname('ERRORSforCVCM3init', who) '=ERRORSforyinit']);
eval([genvarname('TotalErrorForCVCM3init', who)
'=TotalErrorForyinit']);
eval([genvarname('RELEERRORSforCVCM3init', who)
'=RELEERRORSforyinit']);
eval([genvarname('TotalRelErrorForCVCM3init', who)
'=TotalRelErrorForyinit']);
eval([genvarname('ACCURACYforCVCM3init', who)
'=ACCURACYforyinit']);
eval([genvarname('PRECISIONforCVCM3init', who)
'=PRECISIONforyinit']);
eval([genvarname('WforCVCM3init', who) '=Wforyinit']);

eval([genvarname('CVCM3duo', who) '=yduo']);
eval([genvarname('CVCM3initduo', who) '=yinitduo']);

eval([genvarname('timecompleteCVCM3', who) '=timecomplete']);
eval([genvarname('lentimecompleteCVCM3', who)
'=lentimecomplete']);
eval([genvarname('CVCM3complete', who) '=ycomplete']);
eval([genvarname('CVCM3initcomplete', who) '=yinitcomplete']);

eval([genvarname('tadjustmentCVCM3', who) '=tadjustment']);
eval([genvarname('CVCM3adjustment', who) '=yadjustment']);
eval([genvarname('CVCM3initadjustment', who)
'=yinitadjustment']);

eval([genvarname('CVCM3fitfirst', who) '=yfitfirst']);
eval([genvarname('CVCM3fitlast', who) '=yfitlast']);
eval([genvarname('CVCM3initfitfirst', who) '=yinitfitfirst']);
eval([genvarname('CVCM3initfitlast', who) '=yinitfitlast']);
end;

lentimecompleteCVCM3_vector = zeros(1,periods);
for i=1:periods,
    lentimecompleteCVCM3_vector(i) =
eval(strcat('lentimecompleteCVCM3', num2str(i)));
end;
maxlentimecompleteCVCM3 = max(lentimecompleteCVCM3_vector);

timecomplete_CVCM3 = zeros(maxlentimecompleteCVCM3,periods);
for i=1:periods,
    timecomplete_CVCM3(1:lentimecompleteCVCM3_vector(i),i) =
eval(strcat('timecompleteCVCM3', num2str(i)));
end;

%-----
%-----
%-----
%-----
%-----

```

```

timeCforTML=0;
tobezeroforTML=0;
exitflagforTML=0;
valueC1stforTML=0;
valueC2ndforTML=0;
for i=2:periods,

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforTML',num2s
tr(i-
1))),eval(strcat('WforTML',num2str(i))),eval(strcat('TMLadjustment',n
um2str(i-1))),eval(strcat('TMLadjustment',num2str(i))),SumDuration(i-
1)),eval(strcat('tadjustmentTML',num2str(i-
1))),eval(strcat('tadjustmentTML',num2str(i))),TAU,TAU);
    eval([genvarname('timeCforTML', who) '=tC']);
    eval([genvarname('tobezeroforTML', who) '=tobezero']);
    eval([genvarname('exitflagforTML', who) '=exitflag']);
    eval([genvarname('valueC1stforTML', who) '=yC1']);
    eval([genvarname('valueC2ndforTML', who) '=yC2']);
end;

timeCforCVCM1=0;
tobezeroforCVCM1=0;
exitflagforCVCM1=0;
valueC1stforCVCM1=0;
valueC2ndforCVCM1=0;
for i=2:periods,

[tC,tobezero,yC1,yC2]=FindPointC(eval(strcat('WforCVCM1',num2str(i-
1))),eval(strcat('WforCVCM1',num2str(i))),eval(strcat('CVCM1adjustmen
t',num2str(i-
1))),eval(strcat('CVCM1adjustment',num2str(i))),SumDuration(i-
1)),eval(strcat('tadjustmentCVCM1',num2str(i-
1))),eval(strcat('tadjustmentCVCM1',num2str(i))),TAU,TAU);
    eval([genvarname('timeCforCVCM1', who) '=tC']);
    eval([genvarname('tobezeroforCVCM1', who) '=tobezero']);
    eval([genvarname('exitflagforCVCM1', who) '=exitflag']);
    eval([genvarname('valueC1stforCVCM1', who) '=yC1']);
    eval([genvarname('valueC2ndforCVCM1', who) '=yC2']);
end;

timeCforCVCM2=0;
tobezeroforCVCM2=0;
exitflagforCVCM2=0;
valueC1stforCVCM2=0;
valueC2ndforCVCM2=0;
for i=2:periods,

[tC,tobezero,yC1,yC2]=FindPointC(eval(strcat('WforCVCM2',num2str(i-
1))),eval(strcat('WforCVCM2',num2str(i))),eval(strcat('CVCM2adjustmen
t',num2str(i-
1))),eval(strcat('CVCM2adjustment',num2str(i))),SumDuration(i-
1)),eval(strcat('tadjustmentCVCM2',num2str(i-
1))),eval(strcat('tadjustmentCVCM2',num2str(i))),TAU,TAU);
    eval([genvarname('timeCforCVCM2', who) '=tC']);
    eval([genvarname('tobezeroforCVCM2', who) '=tobezero']);
    eval([genvarname('exitflagforCVCM2', who) '=exitflag']);
    eval([genvarname('valueC1stforCVCM2', who) '=yC1']);
    eval([genvarname('valueC2ndforCVCM2', who) '=yC2']);
end;

```

```

timeCforCVCM3=0;
tobezeroforCVCM3=0;
exitflagforCVCM3=0;
valueC1stforCVCM3=0;
valueC2ndforCVCM3=0;
for i=2:periods,

[tC,tobezero,yC1,yC2]=FindPointC(eval(strcat('WforCVCM3',num2str(i-
1))),eval(strcat('WforCVCM3',num2str(i))),eval(strcat('CVCM3adjustmen
t',num2str(i-
1))),eval(strcat('CVCM3adjustment',num2str(i))),SumDuration(i-
1),eval(strcat('tadjustmentCVCM3',num2str(i-
1))),eval(strcat('tadjustmentCVCM3',num2str(i))),TAU,TAU);
    eval([genvarname('timeCforCVCM3', who) '=tC']);
    eval([genvarname('tobezeroforCVCM3', who) '=tobezero']);
    eval([genvarname('exitflagforCVCM3', who) '=exitflag']);
    eval([genvarname('valueC1stforCVCM3', who) '=yC1']);
    eval([genvarname('valueC2ndforCVCM3', who) '=yC2']);
end;

%-----
%-----

Ddifference = zeros(1,periods+1);
for i=1:periods,
    Ddifference(i+1) = positionadjusted(i+1)-PostForAverage(i+1);
end;

[maxDdifference,postDdifference] = max(Ddifference);

tmllastpart = zeros(maxDdifference+1,periods);
cvcml1lastpart = zeros(maxDdifference+1,periods);
cvcml2lastpart = zeros(maxDdifference+1,periods);
cvcml3lastpart = zeros(maxDdifference+1,periods);
for i=1:periods,
    tmllastpart(1:positionadjusted(i+1)-PostForAverage(i+1)+1,i) =
tml(PostForAverage(i+1):positionadjusted(i+1));
    cvcml1lastpart(1:positionadjusted(i+1)-PostForAverage(i+1)+1,i) =
cvcml(PostForAverage(i+1):positionadjusted(i+1));
    cvcml2lastpart(1:positionadjusted(i+1)-PostForAverage(i+1)+1,i) =
cvcml2(PostForAverage(i+1):positionadjusted(i+1));
    cvcml3lastpart(1:positionadjusted(i+1)-PostForAverage(i+1)+1,i) =
cvcml3(PostForAverage(i+1):positionadjusted(i+1));
end;

meantmllastpart =zeros(1,periods-1);
meancvcml1lastpart =zeros(1,periods-1);
meancvcml2lastpart =zeros(1,periods-1);
meancvcml3lastpart =zeros(1,periods-1);
for i=1:periods-1,
    meantmllastpart(i) = mean(tmllastpart(1:positionadjusted(i+1)-
PostForAverage(i+1)+1,i));
    meancvcml1lastpart(i) =
mean(cvcml1lastpart(1:positionadjusted(i+1)-PostForAverage(i+1)+1,i));
    meancvcml2lastpart(i) =
mean(cvcml2lastpart(1:positionadjusted(i+1)-PostForAverage(i+1)+1,i));
    meancvcml3lastpart(i) =
mean(cvcml3lastpart(1:positionadjusted(i+1)-PostForAverage(i+1)+1,i));
end;

```

```

%-----
%-----

timeCforTMLfinal=0;
tobezeroforTMLfinal=0;
exitflagforTMLfinal=0;
valueC1stforTMLfinal=0;
valueC2ndforTMLfinal=0;
ITforTML=0;
for i=2:periods-1,
    %if meantmllastpart(i-1) > eval(strcat('valueC1stforTML',
num2str(i-1))),
        %if meantmllastpart(i) > eval(strcat('valueC1stforTML',
num2str(i))),
            if eval(strcat('TMLfitfirst', num2str(i))) >
eval(strcat('TMLfitlast', num2str(i-1))) + 0,
                if eval(strcat('TMLfitfirst', num2str(i+1))) >
eval(strcat('TMLfitlast', num2str(i))) + 0,

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforTMLinit',n
um2str(i-
1))),eval(strcat('WforTMLinit',num2str(i))),eval(strcat('TMLinitadjus
tment',num2str(i-
1))),eval(strcat('TMLinitadjustment',num2str(i))),SumDuration(i-
1),eval(strcat('tadjustmentTML',num2str(i-
1))),eval(strcat('tadjustmentTML',num2str(i))),TAU,TAU);
                eval([genvarname('timeCforTMLfinal', who) '=tC']);
                eval([genvarname('tobezeroforTMLfinal', who)
'='tobezero']);
                eval([genvarname('exitflagforTMLfinal', who)
'='exitflag']);
                eval([genvarname('valueC1stforTMLfinal', who) '=yC1']);
                eval([genvarname('valueC2ndforTMLfinal', who) '=yC2']);
                eval([genvarname('ITforTML', who) '=3']);
            else

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforTMLinit',n
um2str(i-
1))),eval(strcat('WforTML',num2str(i))),eval(strcat('TMLinitadjustmen
t',num2str(i-
1))),eval(strcat('TMLadjustment',num2str(i))),SumDuration(i-
1),eval(strcat('tadjustmentTML',num2str(i-
1))),eval(strcat('tadjustmentTML',num2str(i))),TAU,TAU);
                eval([genvarname('timeCforTMLfinal', who) '=tC']);
                eval([genvarname('tobezeroforTMLfinal', who)
'='tobezero']);
                eval([genvarname('exitflagforTMLfinal', who)
'='exitflag']);
                eval([genvarname('valueC1stforTMLfinal', who) '=yC1']);
                eval([genvarname('valueC2ndforTMLfinal', who) '=yC2']);
                eval([genvarname('ITforTML', who) '=2']);
            end;
        else

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforTML',num2s
tr(i-
1))),eval(strcat('WforTML',num2str(i))),eval(strcat('TMLadjustment',n
um2str(i-1))),eval(strcat('TMLadjustment',num2str(i))),SumDuration(i-
1),eval(strcat('tadjustmentTML',num2str(i-
1))),eval(strcat('tadjustmentTML',num2str(i))),TAU,TAU);

```

```

        eval([genvarname('timeCforTMLfinal', who) '=tC']);
        eval([genvarname('tobezeroforTMLfinal', who) '=tobezero']);
        eval([genvarname('exitflagforTMLfinal', who) '=exitflag']);
        eval([genvarname('valueC1stforTMLfinal', who) '=yC1']);
        eval([genvarname('valueC2ndforTMLfinal', who) '=yC2']);
        eval([genvarname('ITforTML', who) '=1']);
    end;
    if i==periods-1,
        %if meantmllastpart(i) >
eval(strcat('valueC1stforTML', num2str(i))),
        if eval(strcat('TMLfitfirst', num2str(i+1))) >
eval(strcat('TMLfitlast', num2str(i))) + 0,

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforTMLinit',n
um2str(i))),eval(strcat('WforTML',num2str(i+1))),eval(strcat('TMLinit
adjustment',num2str(i))),eval(strcat('TMLadjustment',num2str(i+1))),S
umDuration(i),eval(strcat('tadjustmentTML',num2str(i))),eval(strcat('
tadjustmentTML',num2str(i+1))),TAU,TAU);
        eval([genvarname('timeCforTMLfinal', who) '=tC']);
        eval([genvarname('tobezeroforTMLfinal', who)
'=tobezero']);
        eval([genvarname('exitflagforTMLfinal', who)
'=exitflag']);
        eval([genvarname('valueC1stforTMLfinal', who) '=yC1']);
        eval([genvarname('valueC2ndforTMLfinal', who) '=yC2']);
        eval([genvarname('ITforTML', who) '=2']);
    else

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforTML',num2s
tr(i))),eval(strcat('WforTML',num2str(i+1))),eval(strcat('TMLadjustme
nt',num2str(i))),eval(strcat('TMLadjustment',num2str(i+1))),SumDurati
on(i),eval(strcat('tadjustmentTML',num2str(i))),eval(strcat('tadjustm
entTML',num2str(i+1))),TAU,TAU);
        eval([genvarname('timeCforTMLfinal', who) '=tC']);
        eval([genvarname('tobezeroforTMLfinal', who)
'=tobezero']);
        eval([genvarname('exitflagforTMLfinal', who)
'=exitflag']);
        eval([genvarname('valueC1stforTMLfinal', who) '=yC1']);
        eval([genvarname('valueC2ndforTMLfinal', who) '=yC2']);
        eval([genvarname('ITforTML', who) '=1']);
    end;
end;
end;

%-----

timeCforCVCM1final=0;
tobezeroforCVCM1final=0;
exitflagforCVCM1final=0;
valueC1stforCVCM1final=0;
valueC2ndforCVCM1final=0;
ITforCVCM1=0;
for i=2:periods-1,
    %if meancvcmlastpart(i-1) > eval(strcat('valueC1stforCVCM1',
num2str(i-1))),
        %if meancvcmlastpart(i) > eval(strcat('valueC1stforCVCM1',
num2str(i))),
            if eval(strcat('CVCM1fitfirst', num2str(i))) >
eval(strcat('CVCM1fitlast', num2str(i-1))) + 0,

```



```

        if eval(strcat('CVCMLfitfirst', num2str(i+1))) >
eval(strcat('CVCMLfitlast', num2str(i))) + 0,

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforCVCMLinit'
,num2str(i-
1))),eval(strcat('WforCVCMLinit',num2str(i))),eval(strcat('CVCMLinita
djustment',num2str(i-
1))),eval(strcat('CVCMLinitadjustment',num2str(i))),SumDuration(i-
1),eval(strcat('tadjustmentCVCML',num2str(i-
1))),eval(strcat('tadjustmentCVCML',num2str(i))),TAU,TAU);
        eval([genvarname('timeCforCVCMLfinal', who) '=tC']);
        eval([genvarname('tobezeroforCVCMLfinal', who)
'=tobezero']);
        eval([genvarname('exitflagforCVCMLfinal', who)
'=exitflag']);
        eval([genvarname('valueC1stforCVCMLfinal', who) '=yC1']);
        eval([genvarname('valueC2ndforCVCMLfinal', who) '=yC2']);
        eval([genvarname('ITforCVCML', who) '=3']);
    else

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforCVCMLinit'
,num2str(i-
1))),eval(strcat('WforCVCML',num2str(i))),eval(strcat('CVCMLinitadju
stment',num2str(i-
1))),eval(strcat('CVCMLadjustment',num2str(i))),SumDuration(i-
1),eval(strcat('tadjustmentCVCML',num2str(i-
1))),eval(strcat('tadjustmentCVCML',num2str(i))),TAU,TAU);
        eval([genvarname('timeCforCVCMLfinal', who) '=tC']);
        eval([genvarname('tobezeroforCVCMLfinal', who)
'=tobezero']);
        eval([genvarname('exitflagforCVCMLfinal', who)
'=exitflag']);
        eval([genvarname('valueC1stforCVCMLfinal', who) '=yC1']);
        eval([genvarname('valueC2ndforCVCMLfinal', who) '=yC2']);
        eval([genvarname('ITforCVCML', who) '=2']);
    end;
    else

[tC,tobezero,yC1,yC2]=FindPointC(eval(strcat('WforCVCML',num2str(i-
1))),eval(strcat('WforCVCML',num2str(i))),eval(strcat('CVCMLadjustmen
t',num2str(i-
1))),eval(strcat('CVCMLadjustment',num2str(i))),SumDuration(i-
1),eval(strcat('tadjustmentCVCML',num2str(i-
1))),eval(strcat('tadjustmentCVCML',num2str(i))),TAU,TAU);
        eval([genvarname('timeCforCVCMLfinal', who) '=tC']);
        eval([genvarname('tobezeroforCVCMLfinal', who) '=tobezero']);
        eval([genvarname('exitflagforCVCMLfinal', who) '=exitflag']);
        eval([genvarname('valueC1stforCVCMLfinal', who) '=yC1']);
        eval([genvarname('valueC2ndforCVCMLfinal', who) '=yC2']);
        eval([genvarname('ITforCVCML', who) '=1']);
    end;
    if i==periods-1,
        %if meancvcmllastpart(i) >
eval(strcat('valueC1stforCVCML',num2str(i))),
        if eval(strcat('CVCMLfitfirst', num2str(i+1))) >
eval(strcat('CVCMLfitlast', num2str(i))) + 0,

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforCVCMLinit'
,num2str(i))),eval(strcat('WforCVCML',num2str(i+1))),eval(strcat('CVC
MLinitadjustment',num2str(i))),eval(strcat('CVCMLadjustment',num2str(

```

```

i+1))),SumDuration(i),eval(strcat('tadjustmentCVCm1',num2str(i))),eval(
l(strcat('tadjustmentCVCm1',num2str(i+1))),TAU,TAU);
    eval([genvarname('timeCforCVCm1final', who) '=tC']);
    eval([genvarname('tobezeroforCVCm1final', who)
'=tobezero']);
    eval([genvarname('exitflagforCVCm1final', who)
'=exitflag']);
    eval([genvarname('valueC1stforCVCm1final', who) '=yC1']);
    eval([genvarname('valueC2ndforCVCm1final', who) '=yC2']);
    eval([genvarname('ITforCVCm1', who) '=2']);
else
[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforCVCm1',num
2str(i))),eval(strcat('WforCVCm1',num2str(i+1))),eval(strcat('CVCm1ad
justment',num2str(i))),eval(strcat('CVCm1adjustment',num2str(i+1))),S
umDuration(i),eval(strcat('tadjustmentCVCm1',num2str(i))),eval(strcat
('tadjustmentCVCm1',num2str(i+1))),TAU,TAU);
    eval([genvarname('timeCforCVCm1final', who) '=tC']);
    eval([genvarname('tobezeroforCVCm1final', who)
'=tobezero']);
    eval([genvarname('exitflagforCVCm1final', who)
'=exitflag']);
    eval([genvarname('valueC1stforCVCm1final', who) '=yC1']);
    eval([genvarname('valueC2ndforCVCm1final', who) '=yC2']);
    eval([genvarname('ITforCVCm1', who) '=1']);
end;
end;
end;

%-----

timeCforCVCm2final=0;
tobezeroforCVCm2final=0;
exitflagforCVCm2final=0;
valueC1stforCVCm2final=0;
valueC2ndforCVCm2final=0;
ITforCVCm2=0;
for i=2:periods-1,
    %if meancvcm2lastpart(i-1) > eval(strcat('valueC1stforCVCm2',
num2str(i-1))),
    %if meancvcm2lastpart(i) > eval(strcat('valueC1stforCVCm2',
num2str(i))),
    if eval(strcat('CVCm2fitfirst', num2str(i))) >
eval(strcat('CVCm2fitlast', num2str(i-1))) + 0,
        if eval(strcat('CVCm2fitfirst', num2str(i+1))) >
eval(strcat('CVCm2fitlast', num2str(i))) + 0,
[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforCVCm2init'
,num2str(i-
1))),eval(strcat('WforCVCm2init',num2str(i))),eval(strcat('CVCm2inita
djustment',num2str(i-
1))),eval(strcat('CVCm2initadjustment',num2str(i))),SumDuration(i-
1),eval(strcat('tadjustmentCVCm2',num2str(i-
1))),eval(strcat('tadjustmentCVCm2',num2str(i))),TAU,TAU);
        eval([genvarname('timeCforCVCm2final', who) '=tC']);
        eval([genvarname('tobezeroforCVCm2final', who)
'=tobezero']);
        eval([genvarname('exitflagforCVCm2final', who)
'=exitflag']);
        eval([genvarname('valueC1stforCVCm2final', who) '=yC1']);
        eval([genvarname('valueC2ndforCVCm2final', who) '=yC2']);

```

```

        eval([genvarname('ITforCVCM2', who) '=3']);
    else

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforCVCM2init',
num2str(i-1))),eval(strcat('WforCVCM2',num2str(i))),eval(strcat('CVCM2initadju
stment',num2str(i-1))),eval(strcat('CVCM2adjustment',num2str(i))),SumDuration(i-
1),eval(strcat('tadjustmentCVCM2',num2str(i-1))),eval(strcat('tadjustmentCVCM2',num2str(i-
1))),eval(strcat('tadjustmentCVCM2',num2str(i))),TAU,TAU);
        eval([genvarname('timeCforCVCM2final', who) '=tC']);
        eval([genvarname('tobezeroforCVCM2final', who)
'=tobezero']);
        eval([genvarname('exitflagforCVCM2final', who)
'=exitflag']);
        eval([genvarname('valueC1stforCVCM2final', who) '=yC1']);
        eval([genvarname('valueC2ndforCVCM2final', who) '=yC2']);
        eval([genvarname('ITforCVCM2', who) '=2']);
    end;
    else

[tC,tobezero,yC1,yC2]=FindPointC(eval(strcat('WforCVCM2',num2str(i-
1))),eval(strcat('WforCVCM2',num2str(i))),eval(strcat('CVCM2adjustmen
t',num2str(i-1))),eval(strcat('CVCM2adjustment',num2str(i))),SumDuration(i-
1),eval(strcat('tadjustmentCVCM2',num2str(i-1))),eval(strcat('tadjustmentCVCM2',num2str(i-
1))),eval(strcat('tadjustmentCVCM2',num2str(i))),TAU,TAU);
        eval([genvarname('timeCforCVCM2final', who) '=tC']);
        eval([genvarname('tobezeroforCVCM2final', who) '=tobezero']);
        eval([genvarname('exitflagforCVCM2final', who) '=exitflag']);
        eval([genvarname('valueC1stforCVCM2final', who) '=yC1']);
        eval([genvarname('valueC2ndforCVCM2final', who) '=yC2']);
        eval([genvarname('ITforCVCM2', who) '=1']);
    end;
    if i==periods-1,
        %if meancvcm2lastpart(i) >
eval(strcat('valueC1stforCVCM2',num2str(i))),
        if eval(strcat('CVCM2fitfirst', num2str(i+1))) >
eval(strcat('CVCM2fitlast', num2str(i))) + 0,

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforCVCM2init',
num2str(i))),eval(strcat('WforCVCM2',num2str(i+1))),eval(strcat('CVC
M2initadjustment',num2str(i))),eval(strcat('CVCM2adjustment',num2str(
i+1))),SumDuration(i),eval(strcat('tadjustmentCVCM2',num2str(i))),eva
l(strcat('tadjustmentCVCM2',num2str(i+1))),TAU,TAU);
        eval([genvarname('timeCforCVCM2final', who) '=tC']);
        eval([genvarname('tobezeroforCVCM2final', who)
'=tobezero']);
        eval([genvarname('exitflagforCVCM2final', who)
'=exitflag']);
        eval([genvarname('valueC1stforCVCM2final', who) '=yC1']);
        eval([genvarname('valueC2ndforCVCM2final', who) '=yC2']);
        eval([genvarname('ITforCVCM2', who) '=2']);
    else

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforCVCM2',num
2str(i))),eval(strcat('WforCVCM2',num2str(i+1))),eval(strcat('CVCM2ad
justment',num2str(i))),eval(strcat('CVCM2adjustment',num2str(i+1))),S
umDuration(i),eval(strcat('tadjustmentCVCM2',num2str(i))),eval(strcat
('tadjustmentCVCM2',num2str(i+1))),TAU,TAU);
        eval([genvarname('timeCforCVCM2final', who) '=tC']);

```

```

        eval([genvarname('tobezeroforCVCM2final', who)
'='tobezero' ]]);
        eval([genvarname('exitflagforCVCM2final', who)
'='exitflag' ]]);
        eval([genvarname('valueC1stforCVCM2final', who) '=yC1' ]]);
        eval([genvarname('valueC2ndforCVCM2final', who) '=yC2' ]]);
        eval([genvarname('ITforCVCM2', who) '=1' ]]);
    end;
end;
end;

%-----

timeCforCVCM3final=0;
tobezeroforCVCM3final=0;
exitflagforCVCM3final=0;
valueC1stforCVCM3final=0;
valueC2ndforCVCM3final=0;
ITforCVCM3=0;
for i=2:periods-1,
    %if meancvcm3lastpart(i-1) > eval(strcat('valueC1stforCVCM3',
num2str(i-1))),
    %if meancvcm3lastpart(i) > eval(strcat('valueC1stforCVCM3',
num2str(i))),
        if eval(strcat('CVCM3fitfirst', num2str(i))) >
eval(strcat('CVCM3fitlast', num2str(i-1))) + 0,
            if eval(strcat('CVCM3fitfirst', num2str(i+1))) >
eval(strcat('CVCM3fitlast', num2str(i))) + 0,

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforCVCM3init'
,num2str(i-
1))),eval(strcat('WforCVCM3init',num2str(i))),eval(strcat('CVCM3inita
djustment',num2str(i-
1))),eval(strcat('CVCM3initadjustment',num2str(i))),SumDuration(i-
1),eval(strcat('tadjustmentCVCM3',num2str(i-
1))),eval(strcat('tadjustmentCVCM3',num2str(i))),TAU,TAU);
            eval([genvarname('timeCforCVCM3final', who) '=tC' ]]);
            eval([genvarname('tobezeroforCVCM3final', who)
'='tobezero' ]]);
            eval([genvarname('exitflagforCVCM3final', who)
'='exitflag' ]]);
            eval([genvarname('valueC1stforCVCM3final', who) '=yC1' ]]);
            eval([genvarname('valueC2ndforCVCM3final', who) '=yC2' ]]);
            eval([genvarname('ITforCVCM3', who) '=3' ]]);
        else

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforCVCM3init'
,num2str(i-
1))),eval(strcat('WforCVCM3',num2str(i))),eval(strcat('CVCM3initadju
stment',num2str(i-
1))),eval(strcat('CVCM3adjustment',num2str(i))),SumDuration(i-
1),eval(strcat('tadjustmentCVCM3',num2str(i-
1))),eval(strcat('tadjustmentCVCM3',num2str(i))),TAU,TAU);
            eval([genvarname('timeCforCVCM3final', who) '=tC' ]]);
            eval([genvarname('tobezeroforCVCM3final', who)
'='tobezero' ]]);
            eval([genvarname('exitflagforCVCM3final', who)
'='exitflag' ]]);
            eval([genvarname('valueC1stforCVCM3final', who) '=yC1' ]]);
            eval([genvarname('valueC2ndforCVCM3final', who) '=yC2' ]]);
            eval([genvarname('ITforCVCM3', who) '=2' ]]);

```

```

        end;
    else

[tC,tobezero,yC1,yC2]=FindPointC(eval(strcat('WforCVCM3',num2str(i-
1))),eval(strcat('WforCVCM3',num2str(i))),eval(strcat('CVCM3adjustmen
t',num2str(i-
1))),eval(strcat('CVCM3adjustment',num2str(i)),SumDuration(i-
1),eval(strcat('tadjustmentCVCM3',num2str(i-
1))),eval(strcat('tadjustmentCVCM3',num2str(i)),TAU,TAU);
    eval([genvarname('timeCforCVCM3final', who) '=tC']);
    eval([genvarname('tobezeroforCVCM3final', who) '=tobezero']);
    eval([genvarname('exitflagforCVCM3final', who) '=exitflag']);
    eval([genvarname('valueC1stforCVCM3final', who) '=yC1']);
    eval([genvarname('valueC2ndforCVCM3final', who) '=yC2']);
    eval([genvarname('ITforCVCM3', who) '=1']);
    end;
    if i==periods-1,
        %if meancvcm3lastpart(i) >
eval(strcat('valueC1stforCVCM3',num2str(i))),
        if eval(strcat('CVCM3fitfirst', num2str(i+1))) >
eval(strcat('CVCM3fitlast', num2str(i))) + 0,

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforCVCM3init'
,num2str(i))),eval(strcat('WforCVCM3',num2str(i+1))),eval(strcat('CVC
M3initadjustment',num2str(i))),eval(strcat('CVCM3adjustment',num2str(
i+1))),SumDuration(i),eval(strcat('tadjustmentCVCM3',num2str(i))),eva
l(strcat('tadjustmentCVCM3',num2str(i+1))),TAU,TAU);
    eval([genvarname('timeCforCVCM3final', who) '=tC']);
    eval([genvarname('tobezeroforCVCM3final', who)
'=tobezero']);
    eval([genvarname('exitflagforCVCM3final', who)
'=exitflag']);
    eval([genvarname('valueC1stforCVCM3final', who) '=yC1']);
    eval([genvarname('valueC2ndforCVCM3final', who) '=yC2']);
    eval([genvarname('ITforCVCM3', who) '=2']);
    else

[tC,tobezero,exitflag,yC1,yC2]=FindPointC(eval(strcat('WforCVCM3',num
2str(i))),eval(strcat('WforCVCM3',num2str(i+1))),eval(strcat('CVCM3ad
justment',num2str(i))),eval(strcat('CVCM3adjustment',num2str(i+1))),S
umDuration(i),eval(strcat('tadjustmentCVCM3',num2str(i))),eval(strcat
('tadjustmentCVCM3',num2str(i+1))),TAU,TAU);
    eval([genvarname('timeCforCVCM3final', who) '=tC']);
    eval([genvarname('tobezeroforCVCM3final', who)
'=tobezero']);
    eval([genvarname('exitflagforCVCM3final', who)
'=exitflag']);
    eval([genvarname('valueC1stforCVCM3final', who) '=yC1']);
    eval([genvarname('valueC2ndforCVCM3final', who) '=yC2']);
    eval([genvarname('ITforCVCM3', who) '=1']);
    end;
    end;
end;

%-----
%-----
%-----
%-----
%-----

TMLcountedfinal=0;

```

```

TMLfitfinal=0;
ERRORSforTMLfinal=0;
TotalErrorForTMLfinal=0;
RELEERRORSforTMLfinal=0;
TotalRelErrorForTMLfinal=0;
ACCURACYforTMLfinal=0;
PRECISIONforTMLfinal=0;
WforTMLfinal=0;

TMLduofinal=0;

TMLcompletefinal=0;

TMLadjustmentfinal=0;
for i=1:periods-1,
    if eval(strcat('ITforTML', num2str(i)))==1,

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoververtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoververtime,dyoverdtimeduo,timenew,yc
ounted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TML(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU);
        eval([genvarname('TMLcountedfinal', who) '=ycounted']);

        eval([genvarname('TMLfitfinal', who) '=yfit']);
        eval([genvarname('ERRORSforTMLfinal', who) '=ERRORSfory']);
        eval([genvarname('TotalErrorForTMLfinal', who)
'=TotalErrorFory']);
        eval([genvarname('RELEERRORSforTMLfinal', who)
'=RELEERRORSfory']);
        eval([genvarname('TotalRelErrorForTMLfinal', who)
'=TotalRelErrorFory']);
        eval([genvarname('ACCURACYforTMLfinal', who)
'=ACCURACYfory']);
        eval([genvarname('PRECISIONforTMLfinal', who)
'=PRECISIONfory']);
        eval([genvarname('WforTMLfinal', who) '=Wfory']);

        eval([genvarname('TMLduofinal', who) '=yduo']);

        eval([genvarname('TMLcompletefinal', who) '=ycomplete']);

        eval([genvarname('TMLadjustmentfinal', who) '=yadjustment']);
    else

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoververtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoververtime,dyoverdtimeduo,timenew,yc
ounted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit

```

```

,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TML(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU);
    eval([genvarname('TMLcountedfinal', who) '=yinitcounted']);

    eval([genvarname('TMLfitfinal', who) '=yinitfit']);
    eval([genvarname('ERRORSforTMLfinal', who)
'=ERRORSforyinit']);
    eval([genvarname('TotalErrorForTMLfinal', who)
'=TotalErrorForyinit']);
    eval([genvarname('RELEERRORSforTMLfinal', who)
'=RELEERRORSforyinit']);
    eval([genvarname('TotalRelErrorForTMLfinal', who)
'=TotalRelErrorForyinit']);
    eval([genvarname('ACCURACYforTMLfinal', who)
'=ACCURACYforyinit']);
    eval([genvarname('PRECISIONforTMLfinal', who)
'=PRECISIONforyinit']);
    eval([genvarname('WforTMLfinal', who) '=Wforyinit']);

    eval([genvarname('TMLduofinal', who) '=yinitduo']);

    eval([genvarname('TMLcompletefinal', who) '=yinitcomplete']);

    eval([genvarname('TMLadjustmentfinal', who)
'=yinitadjustment']);
    end;
end;
for i=periods,

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoverdtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoverdtime,dyoverdtimeduo,timenew,yco
unted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TML(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU);
    eval([genvarname('TMLcountedfinal', who) '=ycounted']);

    eval([genvarname('TMLfitfinal', who) '=yfit']);
    eval([genvarname('ERRORSforTMLfinal', who) '=ERRORSfory']);
    eval([genvarname('TotalErrorForTMLfinal', who)
'=TotalErrorFory']);
    eval([genvarname('RELEERRORSforTMLfinal', who) '=RELEERRORSfory']);
    eval([genvarname('TotalRelErrorForTMLfinal', who)
'=TotalRelErrorFory']);
    eval([genvarname('ACCURACYforTMLfinal', who) '=ACCURACYfory']);
    eval([genvarname('PRECISIONforTMLfinal', who) '=PRECISIONfory']);
    eval([genvarname('WforTMLfinal', who) '=Wfory']);

```

```

eval([genvarname('TMLduofinal', who) '=yduo']);

eval([genvarname('TMLcompletedefinal', who) '=ycomplete']);

eval([genvarname('TMLadjustmentfinal', who) '=yadjustment']);
end;

TML_completedefinal = zeros(maxlentimecompleteTML,periods);
for i=1:periods,
    TML_completedefinal(1:lentimecompleteTML_vector(i),i) =
eval(strcat('TMLcompletedefinal', num2str(i)));
end;

%-----

CVCM1countedfinal=0;

CVCM1fitfinal=0;
ERRORSforCVCM1final=0;
TotalErrorForCVCM1final=0;
RELEERRORSforCVCM1final=0;
TotalRelErrorForCVCM1final=0;
ACCURACYforCVCM1final=0;
PRECISIONforCVCM1final=0;
WforCVCM1final=0;

CVCM1duofinal=0;

CVCM1completedefinal=0;

CVCM1adjustmentfinal=0;
for i=1:periods-1,
    if eval(strcat('ITforCVCM1', num2str(i)))==1,

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoverdtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoverdtime,dyoverdtimeduo,timenew,yco
unted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),CVCM1(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU);
        eval([genvarname('CVCM1countedfinal', who) '=ycounted']);

        eval([genvarname('CVCM1fitfinal', who) '=yfit']);
        eval([genvarname('ERRORSforCVCM1final', who) '=ERRORSfory']);
        eval([genvarname('TotalErrorForCVCM1final', who)
'=TotalErrorFory']);
        eval([genvarname('RELEERRORSforCVCM1final', who)
'=RELEERRORSfory']);
        eval([genvarname('TotalRelErrorForCVCM1final', who)
'=TotalRelErrorFory']);
        eval([genvarname('ACCURACYforCVCM1final', who)
'=ACCURACYfory']);

```



```

        eval([genvarname('PRECISIONforCVCMLfinal', who)
'=PRECISIONfory' ]);
        eval([genvarname('WforCVCMLfinal', who) '=Wfory' ]]);

        eval([genvarname('CVCMLduofinal', who) '=yduo' ]]);

        eval([genvarname('CVCMLcompletefinal', who) '=ycomplete' ]]);

        eval([genvarname('CVCMLadjustmentfinal', who)
'=yadjustment' ]]);
        else

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoververtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoververtime,dyoverdtimeduo,timenew,yc
ounted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),CVCML(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU);
        eval([genvarname('CVCMLcountedfinal', who) '=yinitcounted' ]]);

        eval([genvarname('CVCMLfitfinal', who) '=yinitfit' ]]);
        eval([genvarname('ERRORSforCVCMLfinal', who)
'=ERRORSforyinit' ]]);
        eval([genvarname('TotalErrorForCVCMLfinal', who)
'=TotalErrorForyinit' ]]);
        eval([genvarname('RELEERRORSforCVCMLfinal', who)
'=RELEERRORSforyinit' ]]);
        eval([genvarname('TotalRelErrorForCVCMLfinal', who)
'=TotalRelErrorForyinit' ]]);
        eval([genvarname('ACCURACYforCVCMLfinal', who)
'=ACCURACYforyinit' ]]);
        eval([genvarname('PRECISIONforCVCMLfinal', who)
'=PRECISIONforyinit' ]]);
        eval([genvarname('WforCVCMLfinal', who) '=Wforyinit' ]]);

        eval([genvarname('CVCMLduofinal', who) '=yinitduo' ]]);

        eval([genvarname('CVCMLcompletefinal', who)
'=yinitcomplete' ]]);

        eval([genvarname('CVCMLadjustmentfinal', who)
'=yinitadjustment' ]]);
        end;
end;
for i=periods,

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoververtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoververtime,dyoverdtimeduo,timenew,yc
ounted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf

```

```

oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),CVCM1(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU);
    eval([genvarname('CVCM1countedfinal', who) '=ycounted']);

    eval([genvarname('CVCM1fitfinal', who) '=yfit']);
    eval([genvarname('ERRORSforCVCM1final', who) '=ERRORSfory']);
    eval([genvarname('TotalErrorForCVCM1final', who)
'=TotalErrorFory']);
    eval([genvarname('RELEERRORSforCVCM1final', who)
'=RELEERRORSfory']);
    eval([genvarname('TotalRelErrorForCVCM1final', who)
'=TotalRelErrorFory']);
    eval([genvarname('ACCURACYforCVCM1final', who) '=ACCURACYfory']);
    eval([genvarname('PRECISIONforCVCM1final', who)
'=PRECISIONfory']);
    eval([genvarname('WforCVCM1final', who) '=Wfory']);

    eval([genvarname('CVCM1duofinal', who) '=yduo']);

    eval([genvarname('CVCM1completedefinal', who) '=ycomplete']);

    eval([genvarname('CVCM1adjustmentfinal', who) '=yadjustment']);
end;

CVCM1_completedefinal = zeros(maxlentimecompleteCVCM1,periods);
for i=1:periods,
    CVCM1_completedefinal(1:lentimecompleteCVCM1_vector(i),i) =
eval(strcat('CVCM1completedefinal', num2str(i)));
end;

%-----

CVCM2countedfinal=0;

CVCM2fitfinal=0;
ERRORSforCVCM2final=0;
TotalErrorForCVCM2final=0;
RELEERRORSforCVCM2final=0;
TotalRelErrorForCVCM2final=0;
ACCURACYforCVCM2final=0;
PRECISIONforCVCM2final=0;
WforCVCM2final=0;

CVCM2duofinal=0;

CVCM2completedefinal=0;

CVCM2adjustmentfinal=0;
for i=1:periods-1,
    if eval(strcat('ITforCVCM2', num2str(i)))==1,

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoverdtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoverdtime,dyoverdtime duo, timenew,yco

```

```

unted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),CVCM2(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU);
    eval([genvarname('CVCM2countedfinal', who) '=ycounted']);

    eval([genvarname('CVCM2fitfinal', who) '=yfit']);
    eval([genvarname('ERRORSforCVCM2final', who) '=ERRORSfory']);
    eval([genvarname('TotalErrorForCVCM2final', who)
'=TotalErrorFory']);
    eval([genvarname('RELEERRORSforCVCM2final', who)
'=RELEERRORSfory']);
    eval([genvarname('TotalRelErrorForCVCM2final', who)
'=TotalRelErrorFory']);
    eval([genvarname('ACCURACYforCVCM2final', who)
'=ACCURACYfory']);
    eval([genvarname('PRECISIONforCVCM2final', who)
'=PRECISIONfory']);
    eval([genvarname('WforCVCM2final', who) '=Wfory']);

    eval([genvarname('CVCM2duofinal', who) '=yduo']);

    eval([genvarname('CVCM2completefinal', who) '=ycomplete']);

    eval([genvarname('CVCM2adjustmentfinal', who)
'=yadjustment']);
    else

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoverdtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoverdtime,dyoverdtimeduo,timenew,yco
unted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),CVCM2(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU);
    eval([genvarname('CVCM2countedfinal', who) '=yinitcounted']);

    eval([genvarname('CVCM2fitfinal', who) '=yinitfit']);
    eval([genvarname('ERRORSforCVCM2final', who)
'=ERRORSforyinit']);
    eval([genvarname('TotalErrorForCVCM2final', who)
'=TotalErrorForyinit']);
    eval([genvarname('RELEERRORSforCVCM2final', who)
'=RELEERRORSforyinit']);
    eval([genvarname('TotalRelErrorForCVCM2final', who)
'=TotalRelErrorForyinit']);
    eval([genvarname('ACCURACYforCVCM2final', who)
'=ACCURACYforyinit']);

```

```

        eval([genvarname('PRECISIONforCVCM2final', who)
'='PRECISIONforyinit']);
        eval([genvarname('WforCVCM2final', who) '='Wforyinit']);

        eval([genvarname('CVCM2duofinal', who) '=yinitduo']);

        eval([genvarname('CVCM2completefinal', who)
'='yinitcomplete']);

        eval([genvarname('CVCM2adjustmentfinal', who)
'='yinitadjustment']);
    end;
end;
for i=periods,

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoverdtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoverdtime,dyoverdtimeduo,timenew,yco
unted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),CVCM2(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU));
        eval([genvarname('CVCM2countedfinal', who) '=ycounted']);

        eval([genvarname('CVCM2fitfinal', who) '=yfit']);
        eval([genvarname('ERRORSforCVCM2final', who) '=ERRORSfory']);
        eval([genvarname('TotalErrorForCVCM2final', who)
'='TotalErrorFory']);
        eval([genvarname('RELERRORSforCVCM2final', who)
'='RELERRORSfory']);
        eval([genvarname('TotalRelErrorForCVCM2final', who)
'='TotalRelErrorFory']);
        eval([genvarname('ACCURACYforCVCM2final', who) '=ACCURACYfory']);
        eval([genvarname('PRECISIONforCVCM2final', who)
'='PRECISIONfory']);
        eval([genvarname('WforCVCM2final', who) '=Wfory']);

        eval([genvarname('CVCM2duofinal', who) '=yduo']);

        eval([genvarname('CVCM2completefinal', who) '=ycomplete']);

        eval([genvarname('CVCM2adjustmentfinal', who) '=yadjustment']);
end;

CVCM2_completefinal = zeros(maxlentimecompleteCVCM2,periods);
for i=1:periods,
    CVCM2_completefinal(1:lentimecompleteCVCM2_vector(i),i) =
eval(strcat('CVCM2completefinal', num2str(i)));
end;

%-----

CVCM3countedfinal=0;

```

```

CVCM3fitfinal=0;
ERRORSforCVCM3final=0;
TotalErrorForCVCM3final=0;
RELEERRORSforCVCM3final=0;
TotalRelErrorForCVCM3final=0;
ACCURACYforCVCM3final=0;
PRECISIONforCVCM3final=0;
WforCVCM3final=0;

CVCM3duofinal=0;

CVCM3completefinal=0;

CVCM3adjustmentfinal=0;
for i=1:periods-1,
    if eval(strcat('ITforCVCM3', num2str(i)))==1,

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoverdtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoverdtime,dyoverdtime duo,timenew,yco
unted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),CVCM3(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU);
    eval([genvarname('CVCM3countedfinal', who) '=ycounted']);

    eval([genvarname('CVCM3fitfinal', who) '=yfit']);
    eval([genvarname('ERRORSforCVCM3final', who) '=ERRORSfory']);
    eval([genvarname('TotalErrorForCVCM3final', who)
'=TotalErrorFory']);
    eval([genvarname('RELEERRORSforCVCM3final', who)
'=RELEERRORSfory']);
    eval([genvarname('TotalRelErrorForCVCM3final', who)
'=TotalRelErrorFory']);
    eval([genvarname('ACCURACYforCVCM3final', who)
'=ACCURACYfory']);
    eval([genvarname('PRECISIONforCVCM3final', who)
'=PRECISIONfory']);
    eval([genvarname('WforCVCM3final', who) '=Wfory']);

    eval([genvarname('CVCM3duofinal', who) '=yduo']);

    eval([genvarname('CVCM3completefinal', who) '=ycomplete']);

    eval([genvarname('CVCM3adjustmentfinal', who)
'=yadjustment']);
    else

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoverdtimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoverdtime,dyoverdtime duo,timenew,yco
unted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total

```

```

RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),CVCM3(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU);
    eval([genvarname('CVCM3countedfinal', who) '=yinitcounted']);

    eval([genvarname('CVCM3fitfinal', who) '=yinitfit']);
    eval([genvarname('ERRORSforCVCM3final', who)
'=ERRORSforyinit']);
    eval([genvarname('TotalErrorForCVCM3final', who)
'=TotalErrorForyinit']);
    eval([genvarname('RELEERRORSforCVCM3final', who)
'=RELEERRORSforyinit']);
    eval([genvarname('TotalRelErrorForCVCM3final', who)
'=TotalRelErrorForyinit']);
    eval([genvarname('ACCURACYforCVCM3final', who)
'=ACCURACYforyinit']);
    eval([genvarname('PRECISIONforCVCM3final', who)
'=PRECISIONforyinit']);
    eval([genvarname('WforCVCM3final', who) '=Wforyinit']);

    eval([genvarname('CVCM3duofinal', who) '=yinitduo']);

    eval([genvarname('CVCM3completefinal', who)
'=yinitcomplete']);

    eval([genvarname('CVCM3adjustmentfinal', who)
'=yinitadjustment']);
    end;
end;
for i=periods,

[yfitveryinitialunscaled,ERRORSforyinitial,TotalErrorForyinitial,RELE
RRORSforyinitial,TotalRelErrorForyinitial,ACCURACYforyinitial,PRECISI
ONforyinitial,Wforyinitial,yinitialduo,dyoverdttimeinitial,d2yoverd2ti
meinitial,d3yoverd3timeinitial,dyoverdttime,dyoverdtimeduo,timenew,yco
unted,yinitcounted,yfit,ERRORSfory,TotalErrorFory,RELEERRORSfory,Total
RelErrorFory,ACCURACYfory,PRECISIONfory,Wfory,yinitfit,ERRORSforyinit
,TotalErrorForyinit,RELEERRORSforyinit,TotalRelErrorForyinit,ACCURACYf
oryinit,PRECISIONforyinit,Wforyinit,yduo,yinitduo,timecomplete,lentim
ecomplete,ycomplete,yinitcomplete,tadjustment,yadjustment,yinitadjust
ment,yfitfirst,yfitlast,yinitfitfirst,yinitfitlast] =
everystep(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),CVCM3(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),TAU);
    eval([genvarname('CVCM3countedfinal', who) '=ycounted']);

    eval([genvarname('CVCM3fitfinal', who) '=yfit']);
    eval([genvarname('ERRORSforCVCM3final', who) '=ERRORSfory']);
    eval([genvarname('TotalErrorForCVCM3final', who)
'=TotalErrorFory']);
    eval([genvarname('RELEERRORSforCVCM3final', who)
'=RELEERRORSfory']);
    eval([genvarname('TotalRelErrorForCVCM3final', who)
'=TotalRelErrorFory']);
    eval([genvarname('ACCURACYforCVCM3final', who) '=ACCURACYfory']);

```

```

    eval([genvarname('PRECISIONforCVCM3final', who)
    '=PRECISIONfory' ]);
    eval([genvarname('WforCVCM3final', who) '=Wfory' ]);

    eval([genvarname('CVCM3duofinal', who) '=yduo' ]);

    eval([genvarname('CVCM3completefinal', who) '=ycomplete' ]);

    eval([genvarname('CVCM3adjustmentfinal', who) '=yadjustment' ]);
end;

CVCM3_completefinal = zeros(maxlentimecompleteCVCM3,periods);
for i=1:periods,
    CVCM3_completefinal(1:lentimecompleteCVCM3_vector(i),i) =
    eval(strcat('CVCM3completefinal', num2str(i)));
end;

%-----
%-----
%-----
%-----
%-----

KaTML=0;
d1TML=0;
d2TML=0;
for i=2:periods,
    [k,dy1,dy2] =
    AccelFactor(eval(strcat('timeCforTMLfinal', num2str(i-
    1))),eval(strcat('tadjustmentTML', num2str(i-
    1))),eval(strcat('tadjustmentTML', num2str(i))),eval(strcat('WforTMLfi
    nal', num2str(i-1))),eval(strcat('WforTMLfinal', num2str(i))),TAU,TAU);
    eval([genvarname('KaTML', who) '=k' ]);
    eval([genvarname('d1TML', who) '=dy1' ]);
    eval([genvarname('d2TML', who) '=dy2' ]);
end;
Ka_TML=zeros(periods-1,1);
for i=1:periods-1,
    Ka_TML(i)= eval(strcat('KaTML', num2str(i)));
end;
Ka_TML

KaCVCM1=0;
d1CVCM1=0;
d2CVCM1=0;
for i=2:periods,
    [k,dy1,dy2] =
    AccelFactor(eval(strcat('timeCforCVCM1final', num2str(i-
    1))),eval(strcat('tadjustmentCVCM1', num2str(i-
    1))),eval(strcat('tadjustmentCVCM1', num2str(i))),eval(strcat('WforCVC
    M1final', num2str(i-
    1))),eval(strcat('WforCVCM1final', num2str(i))),TAU,TAU);
    eval([genvarname('KaCVCM1', who) '=k' ]);
    eval([genvarname('d1CVCM1', who) '=dy1' ]);
    eval([genvarname('d2CVCM1', who) '=dy2' ]);
end;
Ka_CVCM1=zeros(periods-1,1);
for i=1:periods-1,
    Ka_CVCM1(i)= eval(strcat('KaCVCM1', num2str(i)));
end;

```

```

Ka_CVCM1

KaCVCM2=0;
d1CVCM2=0;
d2CVCM2=0;
for i=2:periods,
    [k,dy1,dy2] =
    AccelFactor(eval(strcat('timeCforCVCM2final',num2str(i-
1))),eval(strcat('tadjustmentCVCM2',num2str(i-
1))),eval(strcat('tadjustmentCVCM2',num2str(i))),eval(strcat('WforCVC
M2final',num2str(i-
1))),eval(strcat('WforCVCM2final',num2str(i))),TAU,TAU);
    eval([genvarname('KaCVCM2', who) '=k']);
    eval([genvarname('d1CVCM2', who) '=dy1']);
    eval([genvarname('d2CVCM2', who) '=dy2']);
end;
Ka_CVCM2=zeros(periods-1,1);
for i=1:periods-1,
    Ka_CVCM2(i)= eval(strcat('KaCVCM2',num2str(i)));
end;
Ka_CVCM2

KaCVCM3=0;
d1CVCM3=0;
d2CVCM3=0;
for i=2:periods,
    [k,dy1,dy2] =
    AccelFactor(eval(strcat('timeCforCVCM3final',num2str(i-
1))),eval(strcat('tadjustmentCVCM3',num2str(i-
1))),eval(strcat('tadjustmentCVCM3',num2str(i))),eval(strcat('WforCVC
M3final',num2str(i-
1))),eval(strcat('WforCVCM3final',num2str(i))),TAU,TAU);
    eval([genvarname('KaCVCM3', who) '=k']);
    eval([genvarname('d1CVCM3', who) '=dy1']);
    eval([genvarname('d2CVCM3', who) '=dy2']);
end;
Ka_CVCM3=zeros(periods-1,1);
for i=1:periods-1,
    Ka_CVCM3(i)= eval(strcat('KaCVCM3',num2str(i)));
end;
Ka_CVCM3

%-----
%-----
%-----
%-----
%-----

EaTML=0;
for i=2:periods,
    E = ActivEnergy(eval(strcat('KaTML',num2str(i-
1))),AveragePeriodTemperature(i-
1),AveragePeriodTemperature(i),Rgasconst);
    eval([genvarname('EaTML', who) '=E']);
end;
Ea_TML=zeros(periods-1,1);
for i=1:periods-1,
    Ea_TML(i)= eval(strcat('EaTML',num2str(i)));
end;
Ea_TML

```



```

EaCVCM1=0;
for i=2:periods,
    E = ActivEnergy(eval(strcat('KaCVCM1',num2str(i-
1))),AveragePeriodTemperature(i-
1),AveragePeriodTemperature(i),Rgasconst);
    eval([genvarname('EaCVCM1', who) '=E']);
end;
Ea_CVCM1=zeros(periods-1,1);
for i=1:periods-1,
    Ea_CVCM1(i)= eval(strcat('EaCVCM1',num2str(i)));
end;
Ea_CVCM1

EaCVCM2=0;
for i=2:periods,
    E = ActivEnergy(eval(strcat('KaCVCM2',num2str(i-
1))),AveragePeriodTemperature(i-
1),AveragePeriodTemperature(i),Rgasconst);
    eval([genvarname('EaCVCM2', who) '=E']);
end;
Ea_CVCM2=zeros(periods-1,1);
for i=1:periods-1,
    Ea_CVCM2(i)= eval(strcat('EaCVCM2',num2str(i)));
end;
Ea_CVCM2

EaCVCM3=0;
for i=2:periods,
    E = ActivEnergy(eval(strcat('KaCVCM3',num2str(i-
1))),AveragePeriodTemperature(i-
1),AveragePeriodTemperature(i),Rgasconst);
    eval([genvarname('EaCVCM3', who) '=E']);
end;
Ea_CVCM3=zeros(periods-1,1);
for i=1:periods-1,
    Ea_CVCM3(i)= eval(strcat('EaCVCM3',num2str(i)));
end;
Ea_CVCM3

%-----
%-----
%-----
%-----
%-----

KeTML = KeCoefficient(Ka_TML,AveragePeriodTemperature)
KeCVCM1 = KeCoefficient(Ka_CVCM1,AveragePeriodTemperature)
KeCVCM2 = KeCoefficient(Ka_CVCM2,AveragePeriodTemperature)
KeCVCM3 = KeCoefficient(Ka_CVCM3,AveragePeriodTemperature)

%-----
%-----
%-----
%-----
%-----

acceleratedTimeTML = AccelTime(Time,positionadjusted,Ka_TML);
acceleratedTimeCVCM1 = AccelTime(Time,positionadjusted,Ka_CVCM1);
acceleratedTimeCVCM2 = AccelTime(Time,positionadjusted,Ka_CVCM2);

```

```

acceleratedTimeCVCM3 = AccelTime(Time,positionadjusted,Ka_CVCM3);

%-----

tau_TML=acceltau(duration,Ka_TML);
tau_CVCM1=acceltau(duration,Ka_CVCM1);
tau_CVCM2=acceltau(duration,Ka_CVCM2);
tau_CVCM3=acceltau(duration,Ka_CVCM3);

%-----

if b==3,
    disp('Constrained optimization is part of the optimization
toolbox');
    disp('The method used is "fmincon.m". Unfortunately, it usually
gets trapped in local minima')
    disp('because it is designed to work on problems where the
objective and constraint functions')
    disp('are both continuous and have continuous first
derivatives.');
```

disp('Method "fminsearch.m" performs unconstrained optimization.');

disp('It is suggested to use "fminsearch.m" - because it can often handle discontinuity - ');

disp('unless you notice negative values in the tau vector.');

```

    while (b ~= 1 && b ~= 2),
        disp('NOTE:');
        disp('(YES - acceptable only if the optimization toolbox is
installed in your computer)');
```

b = input('Do you want to use constrained optimization for the tau? [1: YES - 2: NO] :');

```

    end;
end;

if b==2,

[TMLforLongTermFit,ERRORSforLongTermFitTML,TotalErrorForLongTermFitTML,RELERRORSforLongTermFitTML,TotalRelativeErrorForLongTermFitTML,ACCURACYforLongTermFitTML,PRECISIONforLongTermFitTML,WforLongTermFitTML,TAUforLongTermFitTML,exitflagforLongTermFitTML]=forLongTermFit(acceleratedTimeTML,tml,tau_TML);

[CVCM1forLongTermFit,ERRORSforLongTermFitCVCM1,TotalErrorForLongTermFitCVCM1,RELERRORSforLongTermFitCVCM1,TotalRelativeErrorForLongTermFitCVCM1,ACCURACYforLongTermFitCVCM1,PRECISIONforLongTermFitCVCM1,WforLongTermFitCVCM1,TAUforLongTermFitCVCM1,exitflagforLongTermFitCVCM1]=forLongTermFit(acceleratedTimeCVCM1,cvcm1,tau_CVCM1);

[CVCM2forLongTermFit,ERRORSforLongTermFitCVCM2,TotalErrorForLongTermFitCVCM2,RELERRORSforLongTermFitCVCM2,TotalRelativeErrorForLongTermFitCVCM2,ACCURACYforLongTermFitCVCM2,PRECISIONforLongTermFitCVCM2,WforLongTermFitCVCM2,TAUforLongTermFitCVCM2,exitflagforLongTermFitCVCM2]=forLongTermFit(acceleratedTimeCVCM2,cvcm2,tau_CVCM2);

[CVCM3forLongTermFit,ERRORSforLongTermFitCVCM3,TotalErrorForLongTermFitCVCM3,RELERRORSforLongTermFitCVCM3,TotalRelativeErrorForLongTermFitCVCM3,ACCURACYforLongTermFitCVCM3,PRECISIONforLongTermFitCVCM3,WforLongTermFitCVCM3,TAUforLongTermFitCVCM3,exitflagforLongTermFitCVCM3]=forLongTermFit(acceleratedTimeCVCM3,cvcm3,tau_CVCM3);
elseif b==1,

```

```
[TMLforLongTermFit,ERRORSforLongTermFitTML,TotalErrorForLongTermFitTML,RELERRORSforLongTermFitTML,TotalRelativeErrorForLongTermFitTML,ACCURACYforLongTermFitTML,PRECISIONforLongTermFitTML,WforLongTermFitTML,TAUforLongTermFitTML]=forLongTermFit_alter(acceleratedTimeTML,tml,tau_TML);
```

```
[CVC1forLongTermFit,ERRORSforLongTermFitCVC1,TotalErrorForLongTermFitCVC1,RELERRORSforLongTermFitCVC1,TotalRelativeErrorForLongTermFitCVC1,ACCURACYforLongTermFitCVC1,PRECISIONforLongTermFitCVC1,WforLongTermFitCVC1,TAUforLongTermFitCVC1]=forLongTermFit_alter(acceleratedTimeCVC1,cvc1,tau_CVC1);
```

```
[CVC2forLongTermFit,ERRORSforLongTermFitCVC2,TotalErrorForLongTermFitCVC2,RELERRORSforLongTermFitCVC2,TotalRelativeErrorForLongTermFitCVC2,ACCURACYforLongTermFitCVC2,PRECISIONforLongTermFitCVC2,WforLongTermFitCVC2,TAUforLongTermFitCVC2]=forLongTermFit_alter(acceleratedTimeCVC2,cvc2,tau_CVC2);
```

```
[CVC3forLongTermFit,ERRORSforLongTermFitCVC3,TotalErrorForLongTermFitCVC3,RELERRORSforLongTermFitCVC3,TotalRelativeErrorForLongTermFitCVC3,ACCURACYforLongTermFitCVC3,PRECISIONforLongTermFitCVC3,WforLongTermFitCVC3,TAUforLongTermFitCVC3]=forLongTermFit_alter(acceleratedTimeCVC3,cvc3,tau_CVC3);
```

```
end;
```

```
%-----
```

```
TauLongTermPredictionsTML=LongTermTau(KeTML,Treference,Tpredictions,TAUforLongTermFitTML);
```

```
TauLongTermPredictionsCVC1=LongTermTau(KeCVC1,Treference,Tpredictions,TAUforLongTermFitCVC1);
```

```
TauLongTermPredictionsCVC2=LongTermTau(KeCVC2,Treference,Tpredictions,TAUforLongTermFitCVC2);
```

```
TauLongTermPredictionsCVC3=LongTermTau(KeCVC3,Treference,Tpredictions,TAUforLongTermFitCVC3);
```

```
%-----
```

```
timeLongTerm=[0:10:90000]';
```

```
TMLlongterm=LongTermPrediction(timeLongTerm,WforLongTermFitTML,TauLongTermPredictionsTML);
```

```
CVC1longterm=LongTermPrediction(timeLongTerm,WforLongTermFitCVC1,TauLongTermPredictionsCVC1);
```

```
CVC2longterm=LongTermPrediction(timeLongTerm,WforLongTermFitCVC2,TauLongTermPredictionsCVC2);
```

```
CVC3longterm=LongTermPrediction(timeLongTerm,WforLongTermFitCVC3,TauLongTermPredictionsCVC3);
```

```
%-----
```

```
%-----
```

```
%-----
```

```
%-----
```

```
%-----
```

```
if c==3,
    while (c ~= 1 && c ~= 2),
        c = input('Do you want to see the results in EXCEL files? [1: YES - 2: NO] :');
```

```

        if c==1,
            disp('MATLAB will proceed to a basic transmission of
information to EXCEL files.');
```

disp('This means that you will be able to find all the
information about the initial fit, the "final" fit,');

disp('the intersection times, the acceleration factors,
the activation energies, the Ke coefficients,');

disp('the long-term fit and the long-term predictions in
EXCEL files.');

disp('Any intermediate or further information might be
obtained by the MATLAB via commands and possible calculations.');

```

        end;
    end;
end;

    if c==1,
        xlswrite('vbqc.xls',meantemperatureQCM1,'general','B17');
        xlswrite('vbqc.xls',meantemperatureQCM2,'general','B18');
        xlswrite('vbqc.xls',meantemperatureQCM3,'general','B19');
%-----
        for i=1:periods,
            AveragePeriodTemperature(i) =
mean(PeriodTemperature(:,i));

            xlswrite('vbqc.xls',AveragePeriodTemperature(i),'general',strcat('E',
num2str(i+24)));

            xlswrite('tml.xls',AveragePeriodTemperature(i),strcat('period',num2st
r(i)), 'J1');

            xlswrite('cvcm1.xls',AveragePeriodTemperature(i),strcat('period',num2
str(i)), 'J1');

            xlswrite('cvcm2.xls',AveragePeriodTemperature(i),strcat('period',num2
str(i)), 'J1');

            xlswrite('cvcm3.xls',AveragePeriodTemperature(i),strcat('period',num2
str(i)), 'J1');

            %xlswrite('tml.xls',SampleName,strcat('period',num2str(i)), 'O1');

            %xlswrite('cvcm1.xls',SampleName,strcat('period',num2str(i)), 'O1');

            %xlswrite('cvcm2.xls',SampleName,strcat('period',num2str(i)), 'O1');

            %xlswrite('cvcm3.xls',SampleName,strcat('period',num2str(i)), 'O1');

            xlswrite('tml.xls',TIME(:,i),strcat('period',num2str(i)), 'A6');

            xlswrite('cvcm1.xls',TIME(:,i),strcat('period',num2str(i)), 'A6');

            xlswrite('cvcm2.xls',TIME(:,i),strcat('period',num2str(i)), 'A6');

            xlswrite('cvcm3.xls',TIME(:,i),strcat('period',num2str(i)), 'A6');

            xlswrite('tml.xls',TML(:,i),strcat('period',num2str(i)), 'B6');
```

```

xlswrite('cvcm1.xls',CVCM1(:,i),strcat('period',num2str(i)), 'B6');

xlswrite('cvcm2.xls',CVCM2(:,i),strcat('period',num2str(i)), 'B6');

xlswrite('cvcm3.xls',CVCM3(:,i),strcat('period',num2str(i)), 'B6');
%-----

xlswrite('vbqc.xls',eval(strcat('TotalErrorForTMLinitial',
num2str(i))), 'tmlneed',strcat('A', num2str(i+32)));

xlswrite('vbqc.xls',eval(strcat('TotalErrorForCVCM1initial',
num2str(i))), 'cvcm1need',strcat('A', num2str(i+32)));

xlswrite('vbqc.xls',eval(strcat('TotalErrorForCVCM2initial',
num2str(i))), 'cvcm2need',strcat('A', num2str(i+32)));

xlswrite('vbqc.xls',eval(strcat('TotalErrorForCVCM3initial',
num2str(i))), 'cvcm3need',strcat('A', num2str(i+32)));

xlswrite('vbqc.xls',eval(strcat('TotalRelErrorForTMLinitial',
num2str(i))), 'tmlneed',strcat('B', num2str(i+32)));

xlswrite('vbqc.xls',eval(strcat('TotalRelErrorForCVCM1initial',
num2str(i))), 'cvcm1need',strcat('B', num2str(i+32)));

xlswrite('vbqc.xls',eval(strcat('TotalRelErrorForCVCM2initial',
num2str(i))), 'cvcm2need',strcat('B', num2str(i+32)));

xlswrite('vbqc.xls',eval(strcat('TotalRelErrorForCVCM3initial',
num2str(i))), 'cvcm3need',strcat('B', num2str(i+32)));

xlswrite('vbqc.xls',eval(strcat('ACCURACYforTMLinitial',
num2str(i))), 'tmlneed',strcat('C', num2str(i+32)));

xlswrite('vbqc.xls',eval(strcat('ACCURACYforCVCM1initial',
num2str(i))), 'cvcm1need',strcat('C', num2str(i+32)));

xlswrite('vbqc.xls',eval(strcat('ACCURACYforCVCM2initial',
num2str(i))), 'cvcm2need',strcat('C', num2str(i+32)));

xlswrite('vbqc.xls',eval(strcat('ACCURACYforCVCM3initial',
num2str(i))), 'cvcm3need',strcat('C', num2str(i+32)));

xlswrite('vbqc.xls',eval(strcat('PRECISIONforTMLinitial',
num2str(i))), 'tmlneed',strcat('D', num2str(i+32)));

xlswrite('vbqc.xls',eval(strcat('PRECISIONforCVCM1initial',
num2str(i))), 'cvcm1need',strcat('D', num2str(i+32)));

xlswrite('vbqc.xls',eval(strcat('PRECISIONforCVCM2initial',
num2str(i))), 'cvcm2need',strcat('D', num2str(i+32)));

xlswrite('vbqc.xls',eval(strcat('PRECISIONforCVCM3initial',
num2str(i))), 'cvcm3need',strcat('D', num2str(i+32)));

```

```

        xlswrite('vbqc.xls',eval(strcat('WforTMLinitial',
num2str(i))), 'tmlneed',strcat('F', num2str(i+32)));
        xlswrite('vbqc.xls',eval(strcat('WforCVCM1initial',
num2str(i))), 'cvcm1need',strcat('F', num2str(i+32)));
        xlswrite('vbqc.xls',eval(strcat('WforCVCM2initial',
num2str(i))), 'cvcm2need',strcat('F', num2str(i+32)));
        xlswrite('vbqc.xls',eval(strcat('WforCVCM3initial',
num2str(i))), 'cvcm3need',strcat('F', num2str(i+32)));
%-----
xlswrite('vbqc.xls',eval(strcat('TotalErrorForTMLfinal',
num2str(i))), 'tmlneed',strcat('A', num2str(i+3)));

xlswrite('vbqc.xls',eval(strcat('TotalErrorForCVCM1final',
num2str(i))), 'cvcm1need',strcat('A', num2str(i+3)));

xlswrite('vbqc.xls',eval(strcat('TotalErrorForCVCM2final',
num2str(i))), 'cvcm2need',strcat('A', num2str(i+3)));

xlswrite('vbqc.xls',eval(strcat('TotalErrorForCVCM3final',
num2str(i))), 'cvcm3need',strcat('A', num2str(i+3)));

xlswrite('vbqc.xls',eval(strcat('TotalRelErrorForTMLfinal',
num2str(i))), 'tmlneed',strcat('B', num2str(i+3)));

xlswrite('vbqc.xls',eval(strcat('TotalRelErrorForCVCM1final',
num2str(i))), 'cvcm1need',strcat('B', num2str(i+3)));

xlswrite('vbqc.xls',eval(strcat('TotalRelErrorForCVCM2final',
num2str(i))), 'cvcm2need',strcat('B', num2str(i+3)));

xlswrite('vbqc.xls',eval(strcat('TotalRelErrorForCVCM3final',
num2str(i))), 'cvcm3need',strcat('B', num2str(i+3)));

        xlswrite('vbqc.xls',eval(strcat('ACCURACYforTMLfinal',
num2str(i))), 'tmlneed',strcat('C', num2str(i+3)));
        xlswrite('vbqc.xls',eval(strcat('ACCURACYforCVCM1final',
num2str(i))), 'cvcm1need',strcat('C', num2str(i+3)));
        xlswrite('vbqc.xls',eval(strcat('ACCURACYforCVCM2final',
num2str(i))), 'cvcm2need',strcat('C', num2str(i+3)));
        xlswrite('vbqc.xls',eval(strcat('ACCURACYforCVCM3final',
num2str(i))), 'cvcm3need',strcat('C', num2str(i+3)));

        xlswrite('vbqc.xls',eval(strcat('PRECISIONforTMLfinal',
num2str(i))), 'tmlneed',strcat('D', num2str(i+3)));
        xlswrite('vbqc.xls',eval(strcat('PRECISIONforCVCM1final',
num2str(i))), 'cvcm1need',strcat('D', num2str(i+3)));
        xlswrite('vbqc.xls',eval(strcat('PRECISIONforCVCM2final',
num2str(i))), 'cvcm2need',strcat('D', num2str(i+3)));
        xlswrite('vbqc.xls',eval(strcat('PRECISIONforCVCM3final',
num2str(i))), 'cvcm3need',strcat('D', num2str(i+3)));

        xlswrite('vbqc.xls',eval(strcat('WforTMLfinal',
num2str(i))), 'tmlneed',strcat('F', num2str(i+3)));
        xlswrite('vbqc.xls',eval(strcat('WforCVCM1final',
num2str(i))), 'cvcm1need',strcat('F', num2str(i+3)));
        xlswrite('vbqc.xls',eval(strcat('WforCVCM2final',
num2str(i))), 'cvcm2need',strcat('F', num2str(i+3)));
        xlswrite('vbqc.xls',eval(strcat('WforCVCM3final',
num2str(i))), 'cvcm3need',strcat('F', num2str(i+3)));

```

```

%-----
xlswrite('tml.xls',eval(strcat('TMLfitveryinitialunscaled',
num2str(i))),strcat('period',num2str(i)),'D6');

xlswrite('cvcml.xls',eval(strcat('CVCMLfitveryinitialunscaled',
num2str(i))),strcat('period',num2str(i)),'D6');

xlswrite('cvcml2.xls',eval(strcat('CVCML2fitveryinitialunscaled',
num2str(i))),strcat('period',num2str(i)),'D6');

xlswrite('cvcml3.xls',eval(strcat('CVCML3fitveryinitialunscaled',
num2str(i))),strcat('period',num2str(i)),'D6');

        xlswrite('tml.xls',eval(strcat('ERRORSforTMLinitial',
num2str(i))),strcat('period',num2str(i)),'E6');
        xlswrite('cvcml.xls',eval(strcat('ERRORSforCVCMLinitial',
num2str(i))),strcat('period',num2str(i)),'E6');
        xlswrite('cvcml2.xls',eval(strcat('ERRORSforCVCML2initial',
num2str(i))),strcat('period',num2str(i)),'E6');
        xlswrite('cvcml3.xls',eval(strcat('ERRORSforCVCML3initial',
num2str(i))),strcat('period',num2str(i)),'E6');

xlswrite('tml.xls',eval(strcat('RELEERRORSforTMLinitial',
num2str(i))),strcat('period',num2str(i)),'F6');

xlswrite('cvcml.xls',eval(strcat('RELEERRORSforCVCMLinitial',
num2str(i))),strcat('period',num2str(i)),'F6');

xlswrite('cvcml2.xls',eval(strcat('RELEERRORSforCVCML2initial',
num2str(i))),strcat('period',num2str(i)),'F6');

xlswrite('cvcml3.xls',eval(strcat('RELEERRORSforCVCML3initial',
num2str(i))),strcat('period',num2str(i)),'F6');
%-----
        xlswrite('tml.xls',eval(strcat('d3TMLoverd3timeinitial',
num2str(i))),strcat('period',num2str(i)),'H6');

xlswrite('cvcml.xls',eval(strcat('d3CVCMLoverd3timeinitial',
num2str(i))),strcat('period',num2str(i)),'H6');

xlswrite('cvcml2.xls',eval(strcat('d3CVCML2overd3timeinitial',
num2str(i))),strcat('period',num2str(i)),'H6');

xlswrite('cvcml3.xls',eval(strcat('d3CVCML3overd3timeinitial',
num2str(i))),strcat('period',num2str(i)),'H6');

        xlswrite('tml.xls',eval(strcat('d2TMLoverd2timeinitial',
num2str(i))),strcat('period',num2str(i)),'I6');

xlswrite('cvcml.xls',eval(strcat('d2CVCMLoverd2timeinitial',
num2str(i))),strcat('period',num2str(i)),'I6');

xlswrite('cvcml2.xls',eval(strcat('d2CVCML2overd2timeinitial',
num2str(i))),strcat('period',num2str(i)),'I6');

xlswrite('cvcml3.xls',eval(strcat('d2CVCML3overd2timeinitial',
num2str(i))),strcat('period',num2str(i)),'I6');

```

```

xlswrite('tml.xls',eval(strcat('dTMLoverdtimeinitial',
num2str(i))),strcat('period',num2str(i)),'J6');

xlswrite('cvcml.xls',eval(strcat('dCVCM1overdtimeinitial',
num2str(i))),strcat('period',num2str(i)),'J6');

xlswrite('cvcml2.xls',eval(strcat('dCVCM2overdtimeinitial',
num2str(i))),strcat('period',num2str(i)),'J6');

xlswrite('cvcml3.xls',eval(strcat('dCVCM3overdtimeinitial',
num2str(i))),strcat('period',num2str(i)),'J6');

        xlswrite('tml.xls',eval(strcat('dTMLoverdtime',
num2str(i))),strcat('period',num2str(i)),'K6');
        xlswrite('cvcml.xls',eval(strcat('dCVCM1overdtime',
num2str(i))),strcat('period',num2str(i)),'K6');
        xlswrite('cvcml2.xls',eval(strcat('dCVCM2overdtime',
num2str(i))),strcat('period',num2str(i)),'K6');
        xlswrite('cvcml3.xls',eval(strcat('dCVCM3overdtime',
num2str(i))),strcat('period',num2str(i)),'K6');
%-----
        xlswrite('tml.xls',eval(strcat('timenewTML',
num2str(i))),strcat('period',num2str(i)),'M6');
        xlswrite('cvcml.xls',eval(strcat('timenewCVCM1',
num2str(i))),strcat('period',num2str(i)),'M6');
        xlswrite('cvcml2.xls',eval(strcat('timenewCVCM2',
num2str(i))),strcat('period',num2str(i)),'M6');
        xlswrite('cvcml3.xls',eval(strcat('timenewCVCM3',
num2str(i))),strcat('period',num2str(i)),'M6');

        xlswrite('tml.xls',eval(strcat('TMLcountedfinal',
num2str(i))),strcat('period',num2str(i)),'N6');
        xlswrite('cvcml.xls',eval(strcat('CVCM1countedfinal',
num2str(i))),strcat('period',num2str(i)),'N6');
        xlswrite('cvcml2.xls',eval(strcat('CVCM2countedfinal',
num2str(i))),strcat('period',num2str(i)),'N6');
        xlswrite('cvcml3.xls',eval(strcat('CVCM3countedfinal',
num2str(i))),strcat('period',num2str(i)),'N6');
%-----
        xlswrite('tml.xls',eval(strcat('TMLfitfinal',
num2str(i))),strcat('period',num2str(i)),'P6');
        xlswrite('cvcml.xls',eval(strcat('CVCM1fitfinal',
num2str(i))),strcat('period',num2str(i)),'P6');
        xlswrite('cvcml2.xls',eval(strcat('CVCM2fitfinal',
num2str(i))),strcat('period',num2str(i)),'P6');
        xlswrite('cvcml3.xls',eval(strcat('CVCM3fitfinal',
num2str(i))),strcat('period',num2str(i)),'P6');

        xlswrite('tml.xls',eval(strcat('ERRORSforTMLfinal',
num2str(i))),strcat('period',num2str(i)),'Q6');
        xlswrite('cvcml.xls',eval(strcat('ERRORSforCVCM1final',
num2str(i))),strcat('period',num2str(i)),'Q6');
        xlswrite('cvcml2.xls',eval(strcat('ERRORSforCVCM2final',
num2str(i))),strcat('period',num2str(i)),'Q6');
        xlswrite('cvcml3.xls',eval(strcat('ERRORSforCVCM3final',
num2str(i))),strcat('period',num2str(i)),'Q6');

        xlswrite('tml.xls',eval(strcat('RELEERRORSforTMLfinal',
num2str(i))),strcat('period',num2str(i)),'R6');

```



```

xlswrite('cvcml.xls',eval(strcat('RELEERRORSforCVCMLfinal',
num2str(i))),strcat('period',num2str(i)),'R6');

xlswrite('cvcml2.xls',eval(strcat('RELEERRORSforCVCML2final',
num2str(i))),strcat('period',num2str(i)),'R6');

xlswrite('cvcml3.xls',eval(strcat('RELEERRORSforCVCML3final',
num2str(i))),strcat('period',num2str(i)),'R6');
%-----
        xlswrite('tml.xls',eval(strcat('timecompleteTML',
num2str(i))),strcat('period',num2str(i)),'T6');
        xlswrite('cvcml.xls',eval(strcat('timecompleteCVCML',
num2str(i))),strcat('period',num2str(i)),'T6');
        xlswrite('cvcml2.xls',eval(strcat('timecompleteCVCML2',
num2str(i))),strcat('period',num2str(i)),'T6');
        xlswrite('cvcml3.xls',eval(strcat('timecompleteCVCML3',
num2str(i))),strcat('period',num2str(i)),'T6');

        xlswrite('tml.xls',eval(strcat('TMLcompletefinal',
num2str(i))),strcat('period',num2str(i)),'U6');
        xlswrite('cvcml.xls',eval(strcat('CVCMLcompletefinal',
num2str(i))),strcat('period',num2str(i)),'U6');
        xlswrite('cvcml2.xls',eval(strcat('CVCML2completefinal',
num2str(i))),strcat('period',num2str(i)),'U6');
        xlswrite('cvcml3.xls',eval(strcat('CVCML3completefinal',
num2str(i))),strcat('period',num2str(i)),'U6');
        end;
%-----
        %xlswrite('tmlLongTerm.xls',SampleName,'general','G32');
        %xlswrite('cvcmlLongTerm.xls',SampleName,'general','G32');
        %xlswrite('cvcml2LongTerm.xls',SampleName,'general','G32');
        %xlswrite('cvcml3LongTerm.xls',SampleName,'general','G32');

xlswrite('cvcmlLongTerm.xls',meantemperatureQCM1,'general','K1');
xlswrite('cvcml2LongTerm.xls',meantemperatureQCM2,'general','K1');
xlswrite('cvcml3LongTerm.xls',meantemperatureQCM3,'general','K1');

        for i=1:periods-1,
            xlswrite('vbqc.xls',eval(strcat('ITforTML',
num2str(i))),'tmlneed',strcat('Z', num2str(i+3)));
            xlswrite('vbqc.xls',eval(strcat('ITforCVCML',
num2str(i))),'cvcmlneed',strcat('Z', num2str(i+3)));
            xlswrite('vbqc.xls',eval(strcat('ITforCVCML2',
num2str(i))),'cvcml2need',strcat('Z', num2str(i+3)));
            xlswrite('vbqc.xls',eval(strcat('ITforCVCML3',
num2str(i))),'cvcml3need',strcat('Z', num2str(i+3)));
%-----

xlswrite('tmlLongTerm.xls',eval(strcat('timeCforTMLfinal',num2str(i)
),'general',strcat('C', num2str(i+9))));

xlswrite('cvcmlLongTerm.xls',eval(strcat('timeCforCVCMLfinal',num2str
(i))),'general',strcat('C', num2str(i+9))));

xlswrite('cvcml2LongTerm.xls',eval(strcat('timeCforCVCML2final',num2str
(i))),'general',strcat('C', num2str(i+9))));

```

```

xlswrite('cvcm3LongTerm.xls',eval(strcat('timeCforCVCM3final',num2str(i))), 'general',strcat('C', num2str(i+9)));

xlswrite('tmlLongTerm.xls',eval(strcat('valueClstforTMLfinal',num2str(i))), 'general',strcat('D', num2str(i+9)));

xlswrite('cvcm1LongTerm.xls',eval(strcat('valueClstforCVCM1final',num2str(i))), 'general',strcat('D', num2str(i+9)));

xlswrite('cvcm2LongTerm.xls',eval(strcat('valueClstforCVCM2final',num2str(i))), 'general',strcat('D', num2str(i+9)));

xlswrite('cvcm3LongTerm.xls',eval(strcat('valueClstforCVCM3final',num2str(i))), 'general',strcat('D', num2str(i+9)));

xlswrite('tmlLongTerm.xls',eval(strcat('ITforTML',num2str(i))), 'general',strcat('E', num2str(i+9)));

xlswrite('cvcm1LongTerm.xls',eval(strcat('ITforCVCM1',num2str(i))), 'general',strcat('E', num2str(i+9)));

xlswrite('cvcm2LongTerm.xls',eval(strcat('ITforCVCM2',num2str(i))), 'general',strcat('E', num2str(i+9)));

xlswrite('cvcm3LongTerm.xls',eval(strcat('ITforCVCM3',num2str(i))), 'general',strcat('E', num2str(i+9)));
%-----

xlswrite('tmlLongTerm.xls',eval(strcat('d1TML',num2str(i))), 'general',strcat('G', num2str(i+9)));

xlswrite('cvcm1LongTerm.xls',eval(strcat('d1CVCM1',num2str(i))), 'general',strcat('G', num2str(i+9)));

xlswrite('cvcm2LongTerm.xls',eval(strcat('d1CVCM2',num2str(i))), 'general',strcat('G', num2str(i+9)));

xlswrite('cvcm3LongTerm.xls',eval(strcat('d1CVCM3',num2str(i))), 'general',strcat('G', num2str(i+9)));

xlswrite('tmlLongTerm.xls',eval(strcat('d2TML',num2str(i))), 'general',strcat('H', num2str(i+9)));

xlswrite('cvcm1LongTerm.xls',eval(strcat('d2CVCM1',num2str(i))), 'general',strcat('H', num2str(i+9)));

xlswrite('cvcm2LongTerm.xls',eval(strcat('d2CVCM2',num2str(i))), 'general',strcat('H', num2str(i+9)));

xlswrite('cvcm3LongTerm.xls',eval(strcat('d2CVCM3',num2str(i))), 'general',strcat('H', num2str(i+9)));

xlswrite('tmlLongTerm.xls',eval(strcat('KaTML',num2str(i))), 'general',strcat('I', num2str(i+9)));

```

```

xlswrite('cvcmlLongTerm.xls',eval(strcat('KaCVCm1',num2str(i))), 'gene
ral',strcat('I', num2str(i+9)));

xlswrite('cvcml2LongTerm.xls',eval(strcat('KaCVCm2',num2str(i))), 'gene
ral',strcat('I', num2str(i+9)));

xlswrite('cvcml3LongTerm.xls',eval(strcat('KaCVCm3',num2str(i))), 'gene
ral',strcat('I', num2str(i+9)));
%-----

xlswrite('tmlLongTerm.xls',eval(strcat('EaTmL',num2str(i))), 'general'
,strcat('K', num2str(i+9)));

xlswrite('cvcmlLongTerm.xls',eval(strcat('EaCVCm1',num2str(i))), 'gene
ral',strcat('K', num2str(i+9)));

xlswrite('cvcml2LongTerm.xls',eval(strcat('EaCVCm2',num2str(i))), 'gene
ral',strcat('K', num2str(i+9)));

xlswrite('cvcml3LongTerm.xls',eval(strcat('EaCVCm3',num2str(i))), 'gene
ral',strcat('K', num2str(i+9)));
    end;
%-----

    xlswrite('tmlLongTerm.xls',KeTmL, 'general', 'M9');
    xlswrite('cvcmlLongTerm.xls',KeCVCm1, 'general', 'M9');
    xlswrite('cvcml2LongTerm.xls',KeCVCm2, 'general', 'M9');
    xlswrite('cvcml3LongTerm.xls',KeCVCm3, 'general', 'M9');
%-----

    xlswrite('tmlLongTerm.xls',tau_TmL, 'fit', 'C3');
    xlswrite('cvcmlLongTerm.xls',tau_CVCm1, 'fit', 'C3');
    xlswrite('cvcml2LongTerm.xls',tau_CVCm2, 'fit', 'C3');
    xlswrite('cvcml3LongTerm.xls',tau_CVCm3, 'fit', 'C3');

xlswrite('tmlLongTerm.xls',TAUforLongTermFitTmL, 'fit', 'D3');

xlswrite('cvcmlLongTerm.xls',TAUforLongTermFitCVCm1, 'fit', 'D3');

xlswrite('cvcml2LongTerm.xls',TAUforLongTermFitCVCm2, 'fit', 'D3');

xlswrite('cvcml3LongTerm.xls',TAUforLongTermFitCVCm3, 'fit', 'D3');

xlswrite('tmlLongTerm.xls',WforLongTermFitTmL, 'fit', 'E3');

xlswrite('cvcmlLongTerm.xls',WforLongTermFitCVCm1, 'fit', 'E3');

xlswrite('cvcml2LongTerm.xls',WforLongTermFitCVCm2, 'fit', 'E3');

xlswrite('cvcml3LongTerm.xls',WforLongTermFitCVCm3, 'fit', 'E3');
%-----
xlswrite('tmlLongTerm.xls',acceleratedTimeTmL, 'fit', 'G7');

xlswrite('cvcmlLongTerm.xls',acceleratedTimeCVCm1, 'fit', 'G7');

xlswrite('cvcml2LongTerm.xls',acceleratedTimeCVCm2, 'fit', 'G7');

xlswrite('cvcml3LongTerm.xls',acceleratedTimeCVCm3, 'fit', 'G7');

    xlswrite('tmlLongTerm.xls',tml, 'fit', 'H7');
    xlswrite('cvcmlLongTerm.xls',cvcml, 'fit', 'H7');

```

```

xlswrite('cvcm2LongTerm.xls',cvcm2,'fit','H7');
xlswrite('cvcm3LongTerm.xls',cvcm3,'fit','H7');

xlswrite('tmlLongTerm.xls',TMLforLongTermFit,'fit','J7');
xlswrite('cvcm1LongTerm.xls',CVCM1forLongTermFit,'fit','J7');
xlswrite('cvcm2LongTerm.xls',CVCM2forLongTermFit,'fit','J7');
xlswrite('cvcm3LongTerm.xls',CVCM3forLongTermFit,'fit','J7');

xlswrite('tmlLongTerm.xls',ERRORSforLongTermFitTML,'fit','K7');

xlswrite('cvcm1LongTerm.xls',ERRORSforLongTermFitCVCM1,'fit','K7');

xlswrite('cvcm2LongTerm.xls',ERRORSforLongTermFitCVCM2,'fit','K7');

xlswrite('cvcm3LongTerm.xls',ERRORSforLongTermFitCVCM3,'fit','K7');

xlswrite('tmlLongTerm.xls',RELEERRORSforLongTermFitTML,'fit','L7');

xlswrite('cvcm1LongTerm.xls',RELEERRORSforLongTermFitCVCM1,'fit','L7')
;

xlswrite('cvcm2LongTerm.xls',RELEERRORSforLongTermFitCVCM2,'fit','L7')
;

xlswrite('cvcm3LongTerm.xls',RELEERRORSforLongTermFitCVCM3,'fit','L7')
;

xlswrite('tmlLongTerm.xls',TotalErrorForLongTermFitTML,'fit','K5');

xlswrite('cvcm1LongTerm.xls',TotalErrorForLongTermFitCVCM1,'fit','K5')
);

xlswrite('cvcm2LongTerm.xls',TotalErrorForLongTermFitCVCM2,'fit','K5')
);

xlswrite('cvcm3LongTerm.xls',TotalErrorForLongTermFitCVCM3,'fit','K5')
);

xlswrite('tmlLongTerm.xls',TotalRelativeErrorForLongTermFitTML,'fit',
'L5');

xlswrite('cvcm1LongTerm.xls',TotalRelativeErrorForLongTermFitCVCM1,'f
it','L5');

xlswrite('cvcm2LongTerm.xls',TotalRelativeErrorForLongTermFitCVCM2,'f
it','L5');

xlswrite('cvcm3LongTerm.xls',TotalRelativeErrorForLongTermFitCVCM3,'f
it','L5');

xlswrite('tmlLongTerm.xls',ACCURACYforLongTermFitTML,'fit','K3');

xlswrite('cvcm1LongTerm.xls',ACCURACYforLongTermFitCVCM1,'fit','K3');

xlswrite('cvcm2LongTerm.xls',ACCURACYforLongTermFitCVCM2,'fit','K3');

```

```

xlswrite('cvcm3LongTerm.xls',ACCURACYforLongTermFitCVCM3,'fit','K3');

xlswrite('tmlLongTerm.xls',PRECISIONforLongTermFitTML,'fit','L3');

xlswrite('cvcm1LongTerm.xls',PRECISIONforLongTermFitCVCM1,'fit','L3')
;

xlswrite('cvcm2LongTerm.xls',PRECISIONforLongTermFitCVCM2,'fit','L3')
;

xlswrite('cvcm3LongTerm.xls',PRECISIONforLongTermFitCVCM3,'fit','L3')
;
%-----
    xlswrite('tmlLongTerm.xls',Tpredictions,'fit','A27');
    xlswrite('cvcm1LongTerm.xls',Tpredictions,'fit','A27');
    xlswrite('cvcm2LongTerm.xls',Tpredictions,'fit','A27');
    xlswrite('cvcm3LongTerm.xls',Tpredictions,'fit','A27');

    xlswrite('tmlLongTerm.xls',Treference,'prediction','C6');
    xlswrite('cvcm1LongTerm.xls',Treference,'prediction','C6');
    xlswrite('cvcm2LongTerm.xls',Treference,'prediction','C6');
    xlswrite('cvcm3LongTerm.xls',Treference,'prediction','C6');

xlswrite('tmlLongTerm.xls',TauLongTermPredictionsTML,'prediction','D9
');

xlswrite('cvcm1LongTerm.xls',TauLongTermPredictionsCVCM1,'prediction'
,'D9');

xlswrite('cvcm2LongTerm.xls',TauLongTermPredictionsCVCM2,'prediction'
,'D9');

xlswrite('cvcm3LongTerm.xls',TauLongTermPredictionsCVCM3,'prediction'
,'D9');

xlswrite('tmlLongTerm.xls',timeLongTerm,'prediction','B32');

xlswrite('cvcm1LongTerm.xls',timeLongTerm,'prediction','B32');

xlswrite('cvcm2LongTerm.xls',timeLongTerm,'prediction','B32');

xlswrite('cvcm3LongTerm.xls',timeLongTerm,'prediction','B32');

xlswrite('tmlLongTerm.xls',TMLlongterm,'prediction','D32');

xlswrite('cvcm1LongTerm.xls',CVCM1longterm,'prediction','D32');

xlswrite('cvcm2LongTerm.xls',CVCM2longterm,'prediction','D32');

xlswrite('cvcm3LongTerm.xls',CVCM3longterm,'prediction','D32');
    end;

%-----
%-----
%-----

```

```

%-----
%-----

d=0;
e=0;
while (d ~= 1 && d ~= 2),
    d = input('Do you want to see graphs in MATLAB? [1: YES - 2: NO]
:');
    if d==1,
        while (e~=1 && e~=2 && e~=3 && e~=4 && e~=5 && e~=6 && e~=7
&& e~=8 && e~=9),
            disp('1: all graphs');
            disp('2: all graphs for TML');
            disp('3: all graphs for CVCM1');
            disp('4: all graphs for CVCM2');
            disp('5: all graphs for CVCM3');
            disp('6: all fits');
            disp('7: all long term fits');
            disp('8: all long term predictions');
            disp('9: exit');
            e = input('Which graphs do you want to see?');
            if e==1 || e==2,
                for i=1:periods,
                    figure(400+10*i+1)
                    axes1 = axes('Position',[0.12 0.2 0.75 0.72]);
                    [AX,H1,H2]=plotyy(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('TMLinitialduo',num2str(i))),TIME
E(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('RELEERRORSforTMLinitial',num2str
(i)))));
                    xlabel('time in
hours','FontSize',12,'FontAngle','italic')
                    set(get(AX(1),'Ylabel'),'String',strcat( '[' ,
symbolismos, ' ] Total Mass Loss'))
                    set(get(AX(2),'Ylabel'),'String',strcat('Relative
Errors between TML', num2str(i), ' and TMLinitialfit', num2str(i)))
                    title(strcat(SampleName, ' : TML', num2str(i)))
                    legend('relative errors','initial
fit','data','Location','Best')
                    text(0,-0.18,strcat('total error = ',
num2str(eval(strcat('TotalErrorForTMLinitial',
num2str(i))))),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'left')
                    text(0.5,-0.18,strcat('accuracy = ',
num2str(eval(strcat('ACCURACYforTMLinitial',
num2str(i))))),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'center')
                    text(1,-0.18,strcat('precision = ',
num2str(eval(strcat('PRECISIONforTMLinitial',
num2str(i))))),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'right')

                    figure(400+10*i+2)
                    [AX,H1,H2]=plotyy(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('TMLinitialduo',num2str(i))),TIME
E(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('dTMLoverdtimeduo',num2str(i))))
;
                    xlabel('time in
hours','FontSize',12,'FontAngle','italic')

```

```

        set(get(AX(1),'Ylabel'),'String',strcat( '[' ,
symbolismos, ' ] Total Mass Loss'))
        set(get(AX(2),'Ylabel'),'String',strcat('1st
Derivative', num2str(i),' and Virtual 1st Derivative', num2str(i)))
        title(strcat(SampleName, ' : TML', num2str(i)))
        legend('1st derivative','virtual 1st
derivative','initial fit','data','Location','Best')

        figure(400+10*i+3)
        axes2 = axes('Position',[0.12 0.2 0.75 0.72]);

[AX,H1,H2]=plotyy(eval(strcat('timenewTML',num2str(i))),eval(strcat('
TMLduofinal',num2str(i))),eval(strcat('timenewTML',num2str(i))),eval(
strcat('RELEERRORSforTMLfinal',num2str(i))));
        xlabel('time in
hours','FontSize',12,'FontAngle','italic')
        set(get(AX(1),'Ylabel'),'String',strcat( '[' ,
symbolismos, ' ] Total Mass Loss'))
        set(get(AX(2),'Ylabel'),'String',strcat('Relative
Errors between TML', num2str(i),' and TMLfit', num2str(i)))
        title(strcat(SampleName, ' : TML', num2str(i)))
        legend('relative
errors','fit','data','Location','Best')
        text(0,-0.18,strcat('total error = ',
num2str(eval(strcat('TotalErrorForTMLfinal',
num2str(i))))),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'left')
        text(0.5,-0.18,strcat('accuracy = ',
num2str(eval(strcat('ACCURACYforTMLfinal',
num2str(i))))),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'center')
        text(1,-0.18,strcat('precision = ',
num2str(eval(strcat('PRECISIONforTMLfinal',
num2str(i))))),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'right')
        end;
    end;

    if e==1 || e==3,
        for i=1:periods,
            figure(100+10*i+1)
            axes1 = axes('Position',[0.12 0.2 0.75 0.72]);
            [AX,H1,H2]=plotyy(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('CVCMLinitialduo',num2str(i))),T
IME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('RELEERRORSforCVCMLinitial',num2s
tr(i))));
            xlabel('time in
hours','FontSize',12,'FontAngle','italic')
            set(get(AX(1),'Ylabel'),'String',strcat( '[' ,
symbolismos, ' ] Collected Volatile Condensable Material 1'))
            set(get(AX(2),'Ylabel'),'String',strcat('Relative
Errors between CVCML-', num2str(i),' and CVCMLinitialfit',
num2str(i)))
            title(strcat(SampleName, ' : CVCML-', num2str(i)))
            legend('relative errors','initial
fit','data','Location','Best')
            text(0,-0.18,strcat('total error = ',
num2str(eval(strcat('TotalErrorForCVCMLinitial',
num2str(i))))),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'left')

```

```

        text(0.5,-0.18, strcat('accuracy = ',
num2str(eval(strcat('ACCURACYforCVCMLinitial',
num2str(i))))), 'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'center')
        text(1,-0.18, strcat('precision = ',
num2str(eval(strcat('PRECISIONforCVCMLinitial',
num2str(i))))), 'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'right')

        figure(100+10*i+2)
        [AX,H1,H2]=plotyy(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('CVCMLinitialduo',num2str(i))),T
IME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('dCVCMLoverdtimeduo',num2str(i))
));
        xlabel('time in
hours','FontSize',12,'FontAngle','italic')
        set(get(AX(1),'Ylabel'),'String',strcat(['',
symbolismos, '] Collected Volatile Condensable Material 1'))
        set(get(AX(2),'Ylabel'),'String',strcat('1st
Derivative', num2str(i), ' and Virtual 1st Derivative', num2str(i)))
        title(strcat(SampleName, ' : CVCML-', num2str(i)))
        legend('1st derivative','virtual 1st
derivative','initial fit','data','Location','Best')

        figure(100+10*i+3)
        axes2 = axes('Position',[0.12 0.2 0.75 0.72]);

[AX,H1,H2]=plotyy(eval(strcat('timenewCVCML',num2str(i))),eval(strcat
('CVCMLduofinal',num2str(i))),eval(strcat('timenewCVCML',num2str(i)))
,eval(strcat('RELERRORSforCVCMLfinal',num2str(i))));
        xlabel('time in
hours','FontSize',12,'FontAngle','italic')
        set(get(AX(1),'Ylabel'),'String',strcat(['',
symbolismos, '] Collected Volatile Condensable Material 1'))
        set(get(AX(2),'Ylabel'),'String',strcat('Relative
Errors between CVCML-', num2str(i), ' and CVCMLfit', num2str(i)))
        title(strcat(SampleName, ' : CVCML-', num2str(i)))
        legend('relative
errors','fit','data','Location','Best')
        text(0,-0.18, strcat('total error = ',
num2str(eval(strcat('TotalErrorForCVCMLfinal',
num2str(i))))), 'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'left')
        text(0.5,-0.18, strcat('accuracy = ',
num2str(eval(strcat('ACCURACYforCVCMLfinal',
num2str(i))))), 'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'center')
        text(1,-0.18, strcat('precision = ',
num2str(eval(strcat('PRECISIONforCVCMLfinal',
num2str(i))))), 'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'right')
        end;
    end;

    if e==1 || e==4,
        for i=1:periods,
            figure(200+10*i+1)
            axes1 = axes('Position',[0.12 0.2 0.75 0.72]);
            [AX,H1,H2]=plotyy(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('CVCML2initialduo',num2str(i))),T

```



```

IME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('RELERERRORSforCVCM2initial',num2s
tr(i)))));
        xlabel('time in
hours','FontSize',12,'FontAngle','italic')
        set(get(AX(1),'Ylabel'),'String',strcat( '[' ,
symbolismos, ' ] Collected Volatile Condensable Material 2'))
        set(get(AX(2),'Ylabel'),'String',strcat('Relative
Errors between CVCM2-', num2str(i), ' and CVCM2initialfit',
num2str(i)))
        title(strcat(SampleName, ' : CVCM2-', num2str(i)))
        legend('relative errors','initial
fit','data','Location','Best')
        text(0,-0.18,strcat('total error = ',
num2str(eval(strcat('TotalErrorForCVCM2initial',
num2str(i))))),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'left')
        text(0.5,-0.18,strcat('accuracy = ',
num2str(eval(strcat('ACCURACYforCVCM2initial',
num2str(i))))),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'center')
        text(1,-0.18,strcat('precision = ',
num2str(eval(strcat('PRECISIONforCVCM2initial',
num2str(i))))),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'right')

        figure(200+10*i+2)
        [AX,H1,H2]=plotyy(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('CVCM2initialduo',num2str(i))),T
IME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('dCVCM2overdtimeduo',num2str(i)
)));
        xlabel('time in
hours','FontSize',12,'FontAngle','italic')
        set(get(AX(1),'Ylabel'),'String',strcat( '[' ,
symbolismos, ' ] Collected Volatile Condensable Material 2'))
        set(get(AX(2),'Ylabel'),'String',strcat('1st
Derivative', num2str(i), ' and Virtual 1st Derivative', num2str(i)))
        title(strcat(SampleName, ' : CVCM2-', num2str(i)))
        legend('1st derivative','virtual 1st
derivative','initial fit','data','Location','Best')

        figure(200+10*i+3)
        axes2 = axes('Position',[0.12 0.2 0.75 0.72]);

[AX,H1,H2]=plotyy(eval(strcat('timenewCVCM2',num2str(i))),eval(strcat
('CVCM2duofinal',num2str(i))),eval(strcat('timenewCVCM2',num2str(i)))
,eval(strcat('RELERERRORSforCVCM2final',num2str(i)))));
        xlabel('time in
hours','FontSize',12,'FontAngle','italic')
        set(get(AX(1),'Ylabel'),'String',strcat( '[' ,
symbolismos, ' ] Collected Volatile Condensable Material 2'))
        set(get(AX(2),'Ylabel'),'String',strcat('Relative
Errors between CVCM2-', num2str(i), ' and CVCM2fit', num2str(i)))
        title(strcat(SampleName, ' : CVCM2-', num2str(i)))
        legend('relative
errors','fit','data','Location','Best')
        text(0,-0.18,strcat('total error = ',
num2str(eval(strcat('TotalErrorForCVCM2final',
num2str(i))))),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'left')

```

```

        text(0.5,-0.18, strcat('accuracy = ',
num2str(eval(strcat('ACCURACYforCVCM2final',
num2str(i))))), 'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'center')
        text(1,-0.18, strcat('precision = ',
num2str(eval(strcat('PRECISIONforCVCM2final',
num2str(i))))), 'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'right')
        end;
    end;

    if e==1 || e==5,
        for i=1:periods,
            figure(300+10*i+1)
            axes1 = axes('Position',[0.12 0.2 0.75 0.72]);
            [AX,H1,H2]=plotyy(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('CVCM3initialduo',num2str(i))),T
IME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('RELERRORSforCVCM3initial',num2s
tr(i)))));
            xlabel('time in
hours','FontSize',12,'FontAngle','italic')
            set(get(AX(1),'Ylabel'),'String',strcat(['[',
symbolismos, '] Collected Volatile Condensable Material 3']))
            set(get(AX(2),'Ylabel'),'String',strcat('Relative
Errors between CVCM3-', num2str(i), ' and CVCM3initialfit',
num2str(i)))
            title(strcat(SampleName, ' : CVCM3-', num2str(i)))
            legend('relative errors','initial
fit','data','Location','Best')
            text(0,-0.18, strcat('total error = ',
num2str(eval(strcat('TotalErrorForCVCM3initial',
num2str(i))))), 'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'left')
            text(0.5,-0.18, strcat('accuracy = ',
num2str(eval(strcat('ACCURACYforCVCM3initial',
num2str(i))))), 'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'center')
            text(1,-0.18, strcat('precision = ',
num2str(eval(strcat('PRECISIONforCVCM3initial',
num2str(i))))), 'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'right')

            figure(300+10*i+2)
            [AX,H1,H2]=plotyy(TIME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('CVCM3initialduo',num2str(i))),T
IME(1:positionadjusted(i+1)-
positionadjusted(i)+1,i),eval(strcat('dCVCM3overdtimeduo',num2str(i))
));
            xlabel('time in
hours','FontSize',12,'FontAngle','italic')
            set(get(AX(1),'Ylabel'),'String',strcat(['[',
symbolismos, '] Collected Volatile Condensable Material 3']))
            set(get(AX(2),'Ylabel'),'String',strcat('1st
Derivative', num2str(i), ' and Virtual 1st Derivative', num2str(i)))
            title(strcat(SampleName, ' : CVCM3-', num2str(i)))
            legend('1st derivative','virtual 1st
derivative','initial fit','data','Location','Best')

            figure(300+10*i+3)
            axes2 = axes('Position',[0.12 0.2 0.75 0.72]);

```

```

[AX,H1,H2]=plotyy(eval(strcat('timenewCVCM3',num2str(i))),eval(strcat
('CVCM3duofinal',num2str(i))),eval(strcat('timenewCVCM3',num2str(i)))
,eval(strcat('RELEERRORSforCVCM3final',num2str(i))));
    xlabel('time in
hours','FontSize',12,'FontAngle','italic')
    set(get(AX(1),'Ylabel'),'String',strcat(['',
symbolismos, '] Collected Volatile Condensable Material 3'))
    set(get(AX(2),'Ylabel'),'String',strcat('Relative
Errors between CVCM3-', num2str(i), ' and CVCM3fit', num2str(i)))
    title(strcat(SampleName, ' : CVCM3-', num2str(i)))
    legend('relative
errors','fit','data','Location','Best')
    text(0,-0.18,strcat('total error = ',
num2str(eval(strcat('TotalErrorForCVCM3final',
num2str(i))))),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'left')
    text(0.5,-0.18,strcat('accuracy = ',
num2str(eval(strcat('ACCURACYforCVCM3final',
num2str(i))))),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'center')
    text(1,-0.18,strcat('precision = ',
num2str(eval(strcat('PRECISIONforCVCM3final',
num2str(i))))),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'right')
    end;
end;

%-----

    if e==1 || e==2 || e==6,
        figure(40)
        axes3 = axes('Position',[0.12 0.2 0.75 0.72]);
        plot(Time,tml,timecomplete_TML,TML_completefinal,'--
');
        xlabel('time in
hours','FontSize',12,'FontAngle','italic')
        ylabel(strcat(['', symbolismos, '] Total Mass Loss'))
        title(strcat(SampleName, ' : TML'))
        legend('data','Location','Best')
        text(0,-0.18,strcat('acceleration factors = ',
mat2str(Ka_TML), ' and activation energies = ',
mat2str(Ea_TML)),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'left')
        end;

    if e==1 || e==3 || e==6,
        figure(10)
        axes3 = axes('Position',[0.12 0.2 0.75 0.72]);

plot(Time,cvcm1,timecomplete_CVCM1,CVCM1_completefinal,'--');
    xlabel('time in
hours','FontSize',12,'FontAngle','italic')
    ylabel(strcat(['', symbolismos, '] Collected Volatile
Condensable Material 1'))
    title(strcat(SampleName, ' : CVCM1'))
    legend('data','Location','Best')
    text(0,-0.18,strcat('acceleration factors = ',
mat2str(Ka_CVCM1), ' and activation energies = ',
mat2str(Ea_CVCM1)),'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'left')

```

```

end;

if e==1 || e==4 || e==6,
    figure(20)
    axes3 = axes('Position',[0.12 0.2 0.75 0.72]);

plot(Time,cvcm2,timecomplete_CVCM2,CVCM2_completefinal,'--');
    xlabel('time in
hours','FontSize',12,'FontAngle','italic')
    ylabel(strcat('[',symbolismos,'] Collected Volatile
Condensable Material 2'))
    title(strcat(SampleName,': CVCM2'))
    legend('data','Location','Best')
    text(0,-0.18,strcat('acceleration factors = ',
mat2str(Ka_CVCM2), ' and activation energies = ',
mat2str(Ea_CVCM2)), 'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'left')
end;

if e==1 || e==5 || e==6,
    figure(30)
    axes3 = axes('Position',[0.12 0.2 0.75 0.72]);

plot(Time,cvcm3,timecomplete_CVCM3,CVCM3_completefinal,'--');
    xlabel('time in
hours','FontSize',12,'FontAngle','italic')
    ylabel(strcat('[',symbolismos,'] Collected Volatile
Condensable Material 3'))
    title(strcat(SampleName,': CVCM3'))
    legend('data','Location','Best')
    text(0,-0.18,strcat('acceleration factors = ',
mat2str(Ka_CVCM3), ' and activation energies = ',
mat2str(Ea_CVCM3)), 'EdgeColor',[0 0 0], 'BackgroundColor', [1 1 1],
'Units', 'normalized', 'HorizontalAlignment', 'left')
end;

%-----

if e==1 || e==2 || e==7,
    figure(400)
    axes4 = axes('Position',[0.12 0.2 0.75 0.72]);

plot(acceleratedTimeTML,tml,acceleratedTimeTML,TMLforLongTermFit);
    xlabel('(accelerated)time in
hours','FontSize',12,'FontAngle','italic')
    ylabel(strcat('[',symbolismos,'] Total Mass Loss'))
    title(strcat(SampleName,': TML'))
    legend('data','fit','Location','Best')
    text(0,-0.18,strcat('W = ',
mat2str(WforLongTermFitTML), ' and TAU = ',
mat2str(TAUforLongTermFitTML)), 'EdgeColor',[0 0 0],
'BackgroundColor', [1 1 1], 'Units', 'normalized',
'HorizontalAlignment', 'left')
end;

if e==1 || e==3 || e==7,
    figure(100)
    axes4 = axes('Position',[0.12 0.2 0.75 0.72]);

```

```

plot(acceleratedTimeCVCM1,cvcm1,acceleratedTimeCVCM1,CVCM1forLongTerm
Fit);
    xlabel('(accelerated)time in
hours','FontSize',12,'FontAngle','italic')
    ylabel(strcat('[',symbolismos,'] Collected Volatile
Condensable Material 1'))
    title(strcat(SampleName,': CVCM1'))
    legend('data','fit','Location','Best')
    text(0,-0.18,strcat('W = ',
mat2str(WforLongTermFitCVCM1), ' and TAU = ',
mat2str(TAUforLongTermFitCVCM1)), 'EdgeColor',[0 0 0],
'BackgroundColor', [1 1 1], 'Units', 'normalized',
'HorizontalAlignment', 'left')
    end;

    if e==1 || e==4 || e==7,
        figure(200)
        axes4 = axes('Position',[0.12 0.2 0.75 0.72]);

plot(acceleratedTimeCVCM2,cvcm2,acceleratedTimeCVCM2,CVCM2forLongTerm
Fit);
    xlabel('(accelerated)time in
hours','FontSize',12,'FontAngle','italic')
    ylabel(strcat('[',symbolismos,'] Collected Volatile
Condensable Material 2'))
    title(strcat(SampleName,': CVCM2'))
    legend('data','fit','Location','Best')
    text(0,-0.18,strcat('W = ',
mat2str(WforLongTermFitCVCM2), ' and TAU = ',
mat2str(TAUforLongTermFitCVCM2)), 'EdgeColor',[0 0 0],
'BackgroundColor', [1 1 1], 'Units', 'normalized',
'HorizontalAlignment', 'left')
    end;

    if e==1 || e==5 || e==7,
        figure(300)
        axes4 = axes('Position',[0.12 0.2 0.75 0.72]);

plot(acceleratedTimeCVCM3,cvcm3,acceleratedTimeCVCM3,CVCM3forLongTerm
Fit);
    xlabel('(accelerated)time in
hours','FontSize',12,'FontAngle','italic')
    ylabel(strcat('[',symbolismos,'] Collected Volatile
Condensable Material 3'))
    title(strcat(SampleName,': CVCM3'))
    legend('data','fit','Location','Best')
    text(0,-0.18,strcat('W = ',
mat2str(WforLongTermFitCVCM3), ' and TAU = ',
mat2str(TAUforLongTermFitCVCM3)), 'EdgeColor',[0 0 0],
'BackgroundColor', [1 1 1], 'Units', 'normalized',
'HorizontalAlignment', 'left')
    end;

%-----

    if e==1 || e==2 || e==8,
        figure(4000)
        axes5 = axes('Position',[0.12 0.2 0.75 0.72]);
        plot(timeLongTerm,TMLLongterm);

```

```

        xlabel('time in
hours','FontSize',12,'FontAngle','italic')
        ylabel(strcat('[' , symbolismos, '] Total Mass Loss'))
        title(strcat(SampleName, ' : TML (Predictions for
Tref = ',num2str(Treference),' °C )'))
        text(0,-0.18,strcat('Predictions at = ',
mat2str(Tpredictions),' °C'),'EdgeColor',[0 0 0], 'BackgroundColor',
[1 1 1], 'Units', 'normalized', 'HorizontalAlignment', 'left')
        end;

        if e==1 || e==3 || e==8,
            figure(1000)
            axes5 = axes('Position',[0.12 0.2 0.75 0.72]);
            plot(timeLongTerm,CVCM1longterm);
            xlabel('time in
hours','FontSize',12,'FontAngle','italic')
            ylabel(strcat('[' , symbolismos, '] Collected Volatile
Condensable Material 1'))
            title(strcat(SampleName, ' : CVCM1 (Predictions for
Tref = ',num2str(Treference),' °C )'))
            text(0,-0.18,strcat('Predictions at = ',
mat2str(Tpredictions),' °C'),'EdgeColor',[0 0 0], 'BackgroundColor',
[1 1 1], 'Units', 'normalized', 'HorizontalAlignment', 'left')
            end;

        if e==1 || e==4 || e==8,
            figure(2000)
            axes5 = axes('Position',[0.12 0.2 0.75 0.72]);
            plot(timeLongTerm,CVCM2longterm);
            xlabel('time in
hours','FontSize',12,'FontAngle','italic')
            ylabel(strcat('[' , symbolismos, '] Collected Volatile
Condensable Material 2'))
            title(strcat(SampleName, ' : CVCM2 (Predictions for
Tref = ',num2str(Treference),' °C )'))
            text(0,-0.18,strcat('Predictions at = ',
mat2str(Tpredictions),' °C'),'EdgeColor',[0 0 0], 'BackgroundColor',
[1 1 1], 'Units', 'normalized', 'HorizontalAlignment', 'left')
            end;

        if e==1 || e==5 || e==8,
            figure(3000)
            axes5 = axes('Position',[0.12 0.2 0.75 0.72]);
            plot(timeLongTerm,CVCM3longterm);
            xlabel('time in
hours','FontSize',12,'FontAngle','italic')
            ylabel(strcat('[' , symbolismos, '] Collected Volatile
Condensable Material 3'))
            title(strcat(SampleName, ' : CVCM3 (Predictions for
Tref = ',num2str(Treference),' °C )'))
            text(0,-0.18,strcat('Predictions at = ',
mat2str(Tpredictions),' °C'),'EdgeColor',[0 0 0], 'BackgroundColor',
[1 1 1], 'Units', 'normalized', 'HorizontalAlignment', 'left')
            end;

%-----

        if e==1 || e==2 || e==3 || e==4 || e==5 || e==6 || e==7
|| e==8,
            e = 0;

```

```

        end;

        if e==9,
            disp('-----');
        end;
    end;
end;
end;

%-----
%-----
%-----
%-----
%-----

disp('----That is all folks----');
disp('-----THE END-----');

```

XX

### 9.15. FindPointC.m

Βρίσκει το σημείο τομής δυο διαδοχικών περιόδων.

```

function
[tC,tobezero,exitflag,yC1,yC2]=FindPointC(w1,w2,adjustment1,adjustmen
t2,tstart,t1,t2,tau1,tau2);

if nargin==8, tau2=[50 20 5 1 0.5 0.1]; end;
if nargin==7, tau2=[50 20 5 1 0.5 0.1]; tau1=[50 20 5 1 0.5 0.1];
end;
if nargin==6, tau2=[50 20 5 1 0.5 0.1]; tau1=[50 20 5 1 0.5 0.1];
t2=24; end;
if nargin==5, tau2=[50 20 5 1 0.5 0.1]; tau1=[50 20 5 1 0.5 0.1];
t2=24; t1=0; end;

[tC,tobezero,exitflag] = fzero(@(t)
forPointC(t,t1,t2,adjustment1,adjustment2,w1,w2,tau1,tau2),tstart);
% tobezero =
forPointC(tC,t1,t2,adjustment1,adjustment2,w1,w2,tau1,tau2);

yC1 = forValueC(tC,t1,adjustment1,w1,tau1);
yC2 = forValueC(tC,t2,adjustment2,w2,tau2);

```

XX

### 9.16. fitdata.m

Προσαρμόζει τα δεδομένα προσδιορίζοντας το διάνυσμα  $w$  χωρίς περιορισμό μη αρνητικότητας, με τη μέθοδο Gauss.

```

function [yout,err,terr,relerr,trelerr,acc,prec,w]=fitdata(t,y,tau);

```

```

if nargin==2, tau= [50 20 5 1 0.5 0.1]'; end;

lent= length(t);
leny = length(y);
if lent ~= leny, disp ('vectors t and y (tml or cvcm) should be of
equal size'); end;

d= length(tau);
w0= zeros(1,d)';
lenw= length(w0);

a = zeros(lenw,lent);
b = zeros(lenw,lent);
yout = zeros(1,lent);
err = zeros(1,lent);
D = zeros(lent,lenw);

for i=1:lenw;
    for j=1:lent;
        a(i,j)= 1-exp(-t(j)/tau(i));
        b(i,j)= w0(i)*a(i,j);
        yout(j)= sum(b(1:lenw,j));
        err(j)= y(j)-yout(j);
        D(j,i)= 1-exp(-t(j)/tau(i));
    end;
end;

errcol = err';
transD = D';
A = transD*D;
B = transD*errcol;

%status indicator [norm(A)*norm(inverse(A))]
%si = cond(A);

dw = gauss(A,B);
%dw = lsqnonneg(A,B);
w = w0+dw;

terr=0;
relerr = zeros(1,lent);
treterr=0;
acc=0;
prec=0;
for i=1:lenw;
    for j=1:lent;
        a(i,j)= 1-exp(-t(j)/tau(i));
        b(i,j)= w(i)*a(i,j);
        yout(j)= sum(b(1:lenw,j));
        err(j)= abs(y(j)-yout(j));
        terr= sum(err(:));
        acc= -log10(terr);
    end;
end;
for j=2:lent;
    relerr(j)= err(j)/abs(y(j));
    treterr= sum(relerr(:));
end;

```



```

    prec= -log10(trelerr);
end;

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

### 9.17. forLongTermFit.m

Κάνει την μακροπρόθεσμη προσαρμογή, προσδιορίζοντας το διάνυσμα  $w$  και βελτιστοποιώντας το διάνυσμα  $\tau$  χωρίς περιορισμούς μη αρνητικότητας.

```

function
[yout,err,terr,relerr,trelerr,acc,prec,w,taunew,exitflag]=forLongTerm
Fit(t,y,tau);

```

```

count=1;
terrold=1;
terr=1;
while terr>terrold-0.001*terrold,
    terrold=terr;
    [yout_,err_,terr,relerr_,trelerr,acc,prec,w]=definew(t,y,tau);

[yout_,err_,terr,relerr_,trelerr,acc,prec,taunew,exitflag]=definetau(
t,y,w,tau);
    count=count+1;
    if count>10,
        break;
    end;
end;

```

```

%-----
% rows --> columns
yout = yout_';
err = err_';
relerr = relerr_';
%-----

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

### 9.18. forLongTermFit\_alter.m

Κάνει την μακροπρόθεσμη προσαρμογή, προσδιορίζοντας το διάνυσμα  $w$  και βελτιστοποιώντας το διάνυσμα  $\tau$  υπό περιορισμούς μη αρνητικότητας.

```

function
[yout,err,terr,relerr,trelerr,acc,prec,w,taunew]=forLongTermFit_alter
(t,y,tau);

```

```

count=1;
terrold=1;
terr=1;

```

```

while terr>terrold-0.001*terrold,
    terrold=terr;
    [yout_,err_,terr,relerr_,treterr,acc,prec,w]=definew(t,y,tau);

[yout_,err_,terr,relerr_,treterr,acc,prec,taunew]=definetau_alter(t,y
,w,tau);
    count=count+1;
    if count>10,
        break;
    end;
end;

%-----
% rows --> columns
yout = yout_';
err = err_';
relerr = relerr_';
%-----

```

XX

### 9.19. forPointC.m

Επεκτείνει τις συναρτήσεις των καμπυλών δυο διαδοχικών περιόδων προς εύρεση του σημείου τομής τους.

```

function
yout=forPointC(t,t1,t2,adjustment1,adjustment2,w1,w2,taul,tau2);

if nargin==8,tau2=[50 20 5 1 0.5 0.1]'; end;
if nargin==7,tau2=[50 20 5 1 0.5 0.1]'; taul=[50 20 5 1 0.5 0.1]';
end;

lenw1=length(w1);
lenw2=length(w2);

a= zeros(lenw1,1);
b= zeros(lenw1,1);
yout1= 0;
for i=1:lenw1;
    a(i)= 1-exp(-(t-t1)/taul(i));
    b(i)= w1(i)*a(i);
    yout1= sum(b(1:lenw1));
end;

c= zeros(lenw2,1);
d= zeros(lenw2,1);
yout2= 0;
for i=1:lenw2;
    c(i)= 1-exp(-(t-t2)/tau2(i));
    d(i)= w2(i)*c(i);
    yout2= sum(d(1:lenw2));
end;

yout=(adjustment1+yout1)-(adjustment2+yout2);

```

XX

### 9.20. forValueC.m

Επεκτείνει τις συναρτήσεις των καμπυλών δυο διαδοχικών περιόδων προς εύρεση της τιμής του σημείου τομής τους.

```
function yout=forValueC(t,tin,adjustment,w,tau);

if nargin==4,tau=[50 20 5 1 0.5 0.1]'; end;

lenw=length(w);

a= zeros(lenw,1);
b= zeros(lenw,1);
youtpart= 0;
for i=1:lenw;
    a(i)= 1-exp(-(t-tin)/tau(i));
    b(i)= w(i)*a(i);
    youtpart= sum(b(1:lenw));
end;

yout=adjustment+youtpart;
```

XX

### 9.21. gauss.m

Προσαρμόζει τα δεδομένα με την μέθοδο των ελαχίστων τετραγώνων χωρίς περιορισμούς.

```
function x=gauss(a,b);
%Solving ax=b, where a in R^(nxn) and x,b in R^(n)
n = length(b);
%Problem's dimension
for i=1:n-1,
    %Finding max in column
    [amax,imax] = max(abs(a(i:n,i)));
    if amax<eps,
        %Non peculiar matrix
        disp('Singular Matrix');
        return;
    end;
    imax = imax+i-1;
    if imax~=i,
        sa = a(imax,i:n);
        sb = b(imax);
        a(imax,i:n) = a(i,i:n);
        b(imax) = b(i); %Line alternation
        a(i,i:n) = sa;
        b(i) = sb;
    end;
    b(i+1:n) = b(i+1:n)-b(i)*a(i+1:n,i)/a(i,i);
```

```

        a(i+1:n,i+1:n) = a(i+1:n,i+1:n)-a(i+1:n,i)*a(i,i+1:n)/a(i,i);
        %effacement of j-column
    end;
    if abs(a(n,n))<eps,
        disp('Singular Matrix');
        return;
    end;
    %Backward replacement
    x(n,1) = b(n)/a(n,n);
    for i=n-1:-1:1,
        x(i,1) = (b(i)-a(i,i+1:n)*x(i+1:n,1))/a(i,i);
    end,

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

## 9.22. generate.m

Παράγει δεδομένα για συγκεκριμένο χρονικό διάστημα σύμφωνα με τη συνάρτηση προσαρμογής, με βάση τα διανύσματα  $\mathbf{w}$  και  $\boldsymbol{\tau}$ .

```

function yout=generate(t,w,tau);

if nargin==2, tau=[50 20 5 1 0.5 0.1]; end;

lent = length(t);

lenw = length(w);
lentau = length(tau);
if lenw~=lentau, disp('vectors w and tau should be of equal size');
end;

a = ones(lenw,lent);
b = ones(lenw,lent);
yout = ones(1,lent);

for i=1:lenw;
    for j=1:lent;
        a(i,j)= 1-exp(-t(j)/tau(i));
        b(i,j)= w(i)*a(i,j);
        yout(j)= sum(b(1:lenw,j));
    end;
end;

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

## 9.23. KeCoefficient.m

Υπολογίζει τον συντελεστή ke για μακροπρόθεσμες προβλέψεις.

```

function Kefit = KeCoefficient(k,T);

```

```

lenk=length(k);
lenT=length(T);

if lenk~=lenT-1,
    disp('The number of acceleration factors is not compatible with
the number of temperature steps: #Kij=#T-1');
    return;
end;

prodk=zeros(1,lenk);
lnk=zeros(1,lenk);
DT=zeros(1,lenk);
for i=1:lenk,
    prodk(i)=prod(k(1:i));
    lnk(i)=log(prodk(i));
    DT(i)=T(i+1)-T(1);
end;
%-----
meanlnk=mean(lnk);
meanDT=mean(DT);

difflnk=zeros(1,lenk);
diffDT=zeros(1,lenk);
numerator=zeros(1,lenk);
denominator=zeros(1,lenk);
Snumerator=0;
Sdenominator=0;
for i=1:lenk,
    difflnk(i)=lnk(i)-meanlnk;
    diffDT(i)=DT(i)-meanDT;
    numerator(i)=difflnk(i)*diffDT(i);
    denominator(i)=diffDT(i)^2;
    Snumerator=sum(numerator(i));
    Sdenominator=sum(denominator(i));
end

Kelsq=Snumerator/Sdenominator;
c=meanlnk-Kelsq*meanDT;
%-----
b=zeros(1,lenk);
a=zeros(1,lenk);
B=0;
A=0;
for i=1:lenk,
    b(i)=lnk(i)*DT(i);
    a(i)=DT(i)^2;
    B=sum(b(i));
    A=sum(a(i));
end;

Kefit=B/A;
%-----
multiKe=zeros(1,lenk);
for i=1:lenk,
    multiKe(i)=lnk(i)/DT(i);
end;
Ke=mean(multiKe);

```

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

### 9.24. LongTermPrediction.m

Κάνει την μακροπρόθεσμη πρόβλεψη με βάση τα επιταχυνθέντα διανύσματα χρόνου,  $\tau$  και  $w$ .

```
function ltw=LongTermPrediction(timelt, accelw, newtau);

lentimelt=length(timelt);

[m,n]=size(newtau);

lenw=length(accelw);

if lenw~=m,
    disp('the lengths of w and tau are incompatible');
    return;
end;

a=zeros(lentimelt,n,m);
ltw=zeros(lentimelt,n);
for k=1:n,
    for j=1:lentimelt,
        for i=1:m,
            a(j,k,i)=accelw(i)*(1-exp(-timelt(j)/newtau(i,k)));
            ltw(j,k)=sum(a(j,k,:));
        end;
    end;
end;
```

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

### 9.25. LongTermTau.m

Επιταχύνει το διάνυσμα  $\tau$ , για μακροπρόθεσμες προβλέψεις.

```
function newtau=LongTermTau(Ke, Tref, T, tau);

lenT=length(T);
lentau=length(tau);

newtau=zeros(lentau, lenT);
for i=1:lentau,
    for j=1:lenT,
        newtau(i,j)=tau(i)*exp(-Ke*(T(j)-Tref));
    end;
end;
```

XX

## 9.26. tsqerror.m

Υπολογίζει το συνολικό σφάλμα μιας προσαρμογής.

```
function error=tsqerror(t,y,w,tau);

if nargin==3, tau=[50 20 5 1 0.5 0.1]'; end;

lent = length(t);
leny = length(y);
if lent~=leny, disp('vectors time and TML/CVCM should be of equal
size');
    return;
end;

lenw = length(w);
lentau = length(tau);
if lenw~=lentau, disp('vectors w and tau should be of same size');
    return;
end;

a = zeros(lenw,lent);
b = zeros(lenw,lent);
c = zeros(1,lent);
err = zeros(1,lent);
error = 0;

for j=1:lent,
    for i=1:lenw,
        a(i,j)= 1-exp(-t(j)/tau(i));
        b(i,j)= w(i)*a(i,j);
        c(j)= sum(b(:,j));
        err(j)= (y(j)-c(j))^2;
        error= sum(err(:));
    end;
end;
```

XX