

REKENTECHNISCHE VERANDERINGEN IN HET MICROWEATHER-SIMULATIE-MODEL

Een aanzet tot het geschikt maken van het microweather-model voor
korte gewassen.

W. Bouten

Bosboom Toussaintstraat 3

Utrecht

INHOUD

Blz.

1.1.1.	Inleiding	1
2.1.1.	Werking van het microweather-simulatiemodel	2
2.1.2.	De grootte van de integratietijdstap	2
2.2.1.	Veranderingen in het microweather model	4
2.2.2.	De berekening van de bladweerstand	6
2.2.3.	Andere wijzigingen die aangebracht werden	7
3.1.	Vergelijking van de verschillende programma's	
3.1.1.	De resultaten	8
3.1.2.	De rekentijden	8
3.2.1.	Conclusies	9
	Appendix 1 Het gebruik van IMSL-subroutines in CSMP	10
	Appendix 2 Enige opmerkingen bij het gebruik van LEQT1F	11
	Appendix 3 Aanvulling op de symbolenlijst	12
	Appendix 4 Listing van het programma	13

Inleiding

In dit verslag wordt een eerste aanzet tot verandering van het " Micro-waether-simulatiemodel " (Goudriaan 1977) besproken. Deze verandering heeft ten doel, het programma zo aan te passen dat het ook voor berekeningen in korte gewassen (bv. gras) te gebruiken is. Naarmate een gewas korter is, zijn de gradienten groter en zullen transportprocessen sneller verlopen. Rekentechnisch heeft dit tot gevolg dat er met dunnere lagen gewerkt moet worden, zodat de tijdstap waarover geïntegreerd wordt kleiner moet zijn. Om zeer lange rekentijden te voorkomen is in het veranderde programma gekozen voor een andere rekentechniek waarmee de temperatuur en de dampspanning van de lucht in de gewaslaag berekend worden. Het voordeel hiervan is, dat de integratietijdstap aanzienlijk groter gekozen kan worden, omdat de tijdscoëfficiënten van de overblijvende processen groter zijn.

In dit verslag wordt eerst zeer kort dat deel van het oorspronkelijke programma besproken dat veranderd moest worden. Daarna volgt een beschrijving van de veranderingen en vergelijking van de resultaten. In de appendices is een listing van het complete programma opgenomen, omdat door verandering van de rekenwijze ook extra dimensionering en nog enkele andere wijzigingen noodzakelijk waren. Verder bevatten zij een aanvulling op de symbolenlijst en een toelichting op het gebruik van de in het programma gebruikte subroutine uit het IMSL-pakket.

$$\text{DEBIET} = \Delta T * \rho C_p * d \quad (2.1.2.)$$

maar ook:

$$\text{DEBIET} = TC * \text{DELT} \quad (2.1.3.)$$

waarbij: TC : de tijd nodig voor temperatuursvereffening bij const. flux

$$\frac{\Delta T * \rho C_p * TC}{R(d)} \approx \Delta T * \rho C_p * d \quad (2.1.4.)$$

$$TC \approx d^2 \quad (2.1.5.)$$

De tijdstap waarover geïntegreerd wordt is ook evenredig met het kwadraat van de laagdikte. Wanneer een gewas in 8 i.p.v. 3 lagen wordt opgedeeld, zal de rekentijd dus ongeveer $8^3/3^3 \approx 20$ keer zo lang zijn.

2.2.1. Veranderingen in het microweather model

In het veranderde model worden de temperatuur en dampspanning van de lucht binnen de gewaslaag niet langer door integratie berekend. Het bladerdek wordt beschouwd als een lineair systeem waarvan op elke tijdstap de evenwichtsituatie berekend wordt. De onafhankelijke en veranderlijke drijvende factoren zijn : de instraling, windsnelheid, luchttemperatuur en vochtigheid en de bodemtemperatuur.

Wanneer de warmtecapaciteit van de luchtlagen kleiner wordt neemt de tijdconstante af en de rekentijd toe (zie 2.1.2.). De fout die gemaakt wordt door te stellen dat de energiebalans van de laag altijd in evenwicht is, wordt echter ook kleiner. We verwaarlozen dan de capacitieve, bufferende eigenschap van de lagen en bereiken het schema van figuur 1 waarin de capaciteiten op de knooppunten zijn weggelaten. Deze figuur is in een wat gewijzigde vorm overgenomen uit Goudriaan et al. 1972. We kunnen dan voor n lagen de volgende vergelijkingen opstellen.

- voor elke gewaslaag:

$$S_i = H_i + V_i + M_i \quad (3.1.1)$$

$$\rho C_p (\theta_i - T_i) = r_i H_i \quad (3.1.2)$$

$$\rho C_p (\epsilon_i - e_i) = q_i \gamma V_i \quad (3.1.3)$$

$$\epsilon_i = e_{sa} + \Delta_i (\theta_i - T_a) \quad (3.1.4)$$

met M_i : energie gebruikt voor CO_2 -assimilatie $(Jm^{-2}s^{-1})$

ρC_p : volumetrische warmtecap. van de lucht $(Jm^{-3}K^{-1})$

γ : psychrometerconstante $(mbar K^{-1})$

Δ_i : helling verzadigde dampspanningscurve $(mbar K^{-1})$

e_{sa} : verzadigde dampspanning van de lucht $(mbar)$

- voor uitwisseling met de bovenlucht

$$\rho C_p (T_1 - T_a) = R_1 (H_1 + H_2 + H_3 + \dots + H_n + H_{n+1}) \quad (3.1.5)$$

$$\rho C_p (e_1 - e_a) = Q_1 \gamma (V_1 + V_2 + \dots + V_n + V_{n+1}) \quad (3.1.6)$$

- voor uitwisseling tussen de lagen onderling

$$\rho C_p (T_i - T_{i-1}) = R_i (H_i + H_{i+1} + \dots + H_n + H_{n+1}) \quad (3.1.7)$$

$$\rho C_p (e_i - e_{i-1}) = Q_i \gamma (V_i + V_{i+1} + \dots + V_n + V_{n+1}) \quad (3.1.8)$$

- voor uitwisseling met de bodemgrenslaag

$$\rho C_p (T_{n+1} - T_n) = R_{n+1} H_{n+1} \quad (3.1.9)$$

$$\rho C_p (e_{n+1} - e_n) = Q_{n+1} \gamma V_{n+1} \quad (3.1.10)$$

2.1.1. De werking van het Microweather-simulatiemodel

Zeer kort wordt hier het principe van dat deel van het simulatiemodel beschreven dat veranderd is. Voor nadere informatie over het oorspronkelijke programma wordt verwezen naar Goudriaan (1977).

Het microweather-model berekent micrometeorologische grootheden in de gewaslaag en verder de energie- en vochtuitwisseling van het bladerdek met de lucht erboven en de onderliggende bodem.

Als input zijn hiervoor nodig:

- weergegevens op referentiehoogte
- geometrische en fysiologische plantgegevens
- thermische bodemgegevens

Uitgaande van de zonhoogte, een temperatuurprofiel in het gewas en de oppervlaktetemperatuur van de bodem, wordt bij bewolkte hemel de netto geabsorbeerde straling voor elke gewaslaag apart berekend. Vervolgens wordt de bladweerstand uit de gegeven cuticulaire weerstand, de in- en uitwendige CO₂-concentratie, de netto CO₂-assimilatie en de grenslaagweerstand berekend. Uit het initieel dampspanningsprofiel volgt verder de drogende kracht van de lucht, zodat het energieverlies door verdamping en convectief transport en ook de bladtemperatuur berekend kunnen worden. De gehele berekening wordt herhaald voor een heldere hemel, opgesplitst in een bijdrage van diffuse straling en een van directe straling, waarna het gewogen gemiddelde genomen wordt. Vervolgens worden de fluxen tussen de lagen, en de uitwisseling met de bodem en de lucht boven het gewas berekend. Samen met de capaciteit van de luchtlaag geeft dit de veranderingssnelheid van temperatuur en dampdruk, zodat de feitelijke temperatuur en dampdruk door integratie verkregen worden, waarna de berekening herhaald wordt voor de volgende tijdstap.

2.1.2. De grootte van de integratie-tijdstap

De energieflex tussen de gewaslagen is recht evenredig met het temperatuurverschil (ΔT) tussen de lagen en de warmtecapaciteit (ρC_p) van de lucht en verder omgekeerd evenredig met de weerstand $R(d)$.

$$FLUX = \frac{\Delta T * \rho C_p}{R(d)} \quad (2.1.1.)$$

Voor een energiedebiet dat optreedt bij vereffening van een temperatuurverschil tussen twee lagen geldt:

- voor de bodemgrenslaag en uitwisseling met de bodem

$$S_{n+1} + H_{n+2} = H_{n+1} + V_{n+1} \quad (3.1.11)$$

$$\rho C_p (\theta_s - T_{n+1}) = r_{n+1} H_{n+1} / D_s \quad (3.1.12)$$

$$\rho C_p (\epsilon_s - e_{n+1}) = q_{n+1} \gamma V_{n+1} / D_s \quad (3.1.13)$$

$$\epsilon_s = e_{sa} + \Delta_i (T_s - T_a) \quad (3.1.14)$$

$$H_{n+2} = (TEMP1 - T_s) * Cond * 2 / DIK \quad (3.1.15)$$

$$G = -H_{n+2} \quad (3.1.16)$$

met Cond: geleidbaarheid van de bodemlaag $(Jm^{-1}s^{-1}K^{-1})$

DIK: dikte van de bovenste bodemlaag (m)

Ds: oppervlaktevergrotingsfactor van het bodemopp. (m^2/m^2)

Wanneer we deze vergelijkingen tellen komen we op $6n+8$ vergelijkingen met evenzoveel onbekenden. Deze kunnen dus in principe opgelost worden. Om echter de rekentijd te verkorten en de benodigde geheugenruimte te verminderen is eerst het aantal vergelijkingen en onbekenden teruggebracht door substitutie.

Dit kan op twee verschillende manieren gebeuren:

- De eerste gedachte was, het aantal vergelijkingen tot het minimum te beperken door alle onbekenden als functie van H te schrijven. Uit de overgebleven $n+2$ vergelijkingen kunnen de onbekenden $H_1 \dots H_{n+2}$ opgelost worden. Om dit met behulp van de subroutine LEQT1F uit het IMSL-pakket te doen, worden de vergelijkingen eerst in matrix-vorm geschreven. Zo ontstaat de volgende matrix-vergelijking:

$$A * H = B * S$$

waarbij A: matrix met coëf. van $H_1 \dots H_{n+2}$ (zie fig. 2a)

H: vector $H_1 \dots H_{n+2}$ (onbekenden)

B: matrix met coëf. van $S_1 \dots S_{n+2}$ (zie fig. 2b)

S: vector $S_1 \dots S_{n+2}$ ($S_{n+2} = 1$)

Uit de op deze wijze verkregen fluxen kunnen vervolgens de andere onbekenden berekend worden.

Wanneer deze rekenmethode gebruikt wordt doet er zich echter nog een probleem voor, waardoor hij onbruikbaar wordt.

Wanneer er sprake is van extreme stabiliteit ($Ri > 0.24$) is er geen uitwisseling tussen het gewas en de bovenliggende lucht. Dit kan gesimuleerd worden door de weerstand (ABTURR) tussen de bovenkant van het gewas en de referentiehoogte zeer groot te maken in vergelijking tot de weerstanden in het gewas. Dit heeft echter een singular stelsel vergelijkingen tot gevolg, dat dus niet opgelost kan worden.

- Een andere mogelijkheid om het grote aantal $(6n+8)$ vergelijkingen terug te brengen is om alle variabelen te schrijven als functie van temperatuur en dampspanning van de luchtlagen in het gewas. Er blijven dan $2n+2$ vergelijkingen over waaruit weer na omschrijving in matrix-vorm m.b.v. LEQT1F de onbekenden $T_1 \dots T_{n+1}$ en $e_1 \dots e_{n+1}$ opgelost kunnen worden. De matrixvergelijking wordt:

$$A \cdot X = B$$

waarin A: matrix van coeff. van T en e (zie fig. 3)

X: vector waarvan $X_1 \dots X_{n+1} = T_1 \dots T_{n+1}, X_{n+2} \dots X_{2n+2} = e_1 \dots e_{n+1}$

B: vector van constanten (zie fig. 3)

Wanneer T en e bekend zijn, kunnen de andere onbekenden m.b.v. de vergelijkingen 3.1.1 - 3.1.16 berekend worden.

Als nu de weerstand boven het gewas zeer groot is, blijft het stelsel oplosbaar en volgt er uit 3.1.5 en 3.1.6 dat

$$H_1 = -H_2 - H_3 - H_4 \quad (3.1.17)$$

$$V_1 = -V_2 - V_3 - V_4 \quad (3.1.18)$$

De temperatuur en dampspanning van de bovenlucht zijn dus inderdaad niet meer van invloed op de onderliggende lagen.

Toch deden zich in het geval van extreme stabiliteit nog moeilijkheden voor. De matrixvergelijking zelf is goed oplosbaar, maar de iteratie divergeert zeer langzaam. Om dit probleem te omzeilen, zodat de rest van het programma getest kon worden, werd het Richardsons getal gelijk gesteld aan nul.

2.2.2. De berekening van de bladweerstand

In figuur 1 is te zien dat één van de grootheden welke bekend worden verondersteld de bladweerstand is. Het is echter niet mogelijk voor alle bladeren, elk met een andere ligging ten opzichte van invallende straling, de zelfde bladweerstand te gebruiken. Ook kan van deze verschillende bladweerstand geen gewogen gemiddelde genomen worden zoals dit voor de energietransporten in 2.1.1. gebeurd is. Er wordt daarom gewerkt met een soort fictieve bladweerstand welke geldt voor een gehele laag. Deze wordt als volgt berekend. Op de zelfde manier als in 2.1.1. wordt de totale verdamping per laag berekend, zodat voor de fictieve bladweerstand geldt:

$$r_{fl} = (\epsilon - e) \cdot \rho C_p / V \cdot \gamma - r_b \cdot 0.93 \quad (3.1.19)$$

(verg. 3.1.3)

waarin r_b : grenslaagweerstand warmtetransport

Omdat deze weerstand echter berekend wordt m.b.v. een aangenomen temperatuurprofiel, worden in feite verkeerde waarden gebruikt. Met deze waarden wordt uit de matrixvergelijking opnieuw een temperatuurprofiel verkregen welke op zijn beurt gebruikt kan worden voor de berekening van de bladweerstand. Deze iteratie kan verder herhaald worden totdat het procentuele verschil tussen de herberekende waarde en de vorige binnen gestelde grenzen valt.

Wanneer er toch gebruik gemaakt wordt van een iteratief proces, kunnen ook gemakkelijk andere grootheden herberekend worden. Zo is er niet langer gewerkt met de helling van de verzadigde dampspanningscurve bij luchttemperatuur, maar bij de gemiddelde temperatuur van blad en lucht (zie bv. verg. 3.1.4.). Hiervoor is een extra berekening nodig welke uiteindelijk nauwelijks van invloed is op de berekende resultaten.

Doordat de berekening zelf echter iets nauwkeuriger is, convergeert de iteratie iets sneller zodat uiteindelijk de rekentijd toch korter is (verg. tabel I)

2.2.3. Andere wijzigingen die aangebracht werden

Er werden naast de verandering van de rekenwijze nog drie andere kleine wijzigingen aangebracht.

- Er bleken verschillen in vooral bladtemperatuur op te treden, wanneer het gewas in een verschillend aantal lagen werd onderverdeeld (zie fig. 4cend). Dit werd veroorzaakt doordat de grenslaagweerstand werd berekend m.b.v. de windsnelheid aan de bovenkant van de beschouwde gewaslaag. De berekening geschiedt nu met de gemiddelde waarde tussen de boven- en de onderkant van de laag.
- Bij de berekening van het waterverlies van de plant werd gerekend met de evapotranspiratie. Dit werd veranderd in de transpiratie.
- Bij de berekening van de evaporatie is een oppervlaktevergrotingsfactor DS gebruikt welke de verhouding weergeeft tussen het grondoppervlak en de verticale projectie daarvan. Naarmate een grond ruwer is, is het verdampend oppervlak groter.

3.1. Vergelijking van de verschillende programma's

3.1.1. De resultaten

Wanneer we de resultaten van de verschillende programma's **vergelijken**, vallen de volgende dingen op.

- De verschillen in bladtemperatuur tussen het oorspronkelijke programma en dat waarin de rekenwijze veranderd is, zijn zeer klein (verg. fig. **4a** en **4b**). Ook andere grootheden vertonen nauwelijks verschillen. Ze zijn niet verder uitgezet.
- Wanneer het gewas werd ingedeeld in meer lagen, verschilden de bladtemperaturen nogal bij lagen met hoge windsnelheids-gradienten. Na de windsnelheidscorrectie (zie 2.2.3.) komen de profielen vrijwel overeen (fig. **4c** en **4d**). Ook de evapotranspiratie was gelijk.
- Invoering van de oppervlaktevergrotingsfactor is slechts van weinig invloed (fig. **4e**).

3.1.2. De rekentijden

Het grote voordeel van de verandering van de rekenwijze wordt het best weergegeven door tabel I waarin de rekentijden zijn weergegeven welke nodig waren voor de berekening gedurende één hele dag.

TABEL I Vergelijking van de rekentijden van de programma's

NUMLL	3	8
WIB000	57.38	
WIB001	15.46	1:25.92
WIB002	13.15	
WIB003	7.96	24.30

WIB000 : oorspronkelijk microweather programma

WIB001 : vernieuwde rekenwijze (DELT = 180)

WIB002 : nieuwe programma waarbij de herberekening van de helling van de dampspanningscurve in de iteratie (DELT = 180)

WIB003 : nieuwe programma met DELT = 480

3.2.1. Conclusies

De rekentechnische verandering, welke in het programma is aangebracht, is niet van invloed op de berekende grootheden. Wel is de rekentijd voor het gewijzigde programma korter, zelfs in het geval dat een gewas in drie lagen verdeeld wordt. Vooral wanneer het bladerdek in een groot aantal lagen verdeeld wordt zal het gewijzigde programma aanzienlijk lagere kosten met zich meebrengen. Het nadeel van deze rekenmethode is echter wel, dat het programma zelf slechter leesbaar en procesmatig moeilijk te begrijpen is. Ook de moeilijkheden welke zich voordeden bij extreme stabiliteit konden nog niet overwonnen worden. Om dit probleem te omzeilen werd er steeds gerekend met een Richardsonsgetal gelijk aan nul. Het was ook nog de bedoeling het model te testen aan de gegevens van grasland. Door tijdgebrek is dit echter niet gebeurd.

Literatuur

Goudriaan J. 1977 Crop micrometeorologie : a simulation study
Pudoc

Goudriaan J., Waggoner P.E. 1972 Simulation both aerial microclimate
and soil temperature from observations above the
foliar canopy Neth. J. Agric. Sci. 20

Het gebruik van IMSL-subroutines in CSMP

Voor het runnen van een CSMP programma worden de volgende commando's gebruikt:

```
R SETLIB
```

```
EXIT
```

```
.R CSMP
```

```
*FILE.CSM
```

In de procedure die gevolgd wordt, wordt eerst het programma op CSMP-fouten gecontroleerd, vervolgens vertaald in FORTRAN, op Fortran-fouten gecontroleerd en uiteindelijk automatisch geëxecuteerd met behulp van het CSMP.CCL-file. Na executie worden alle overbodige files verwijderd. Indien echter in het CSMP programma subroutines uit de IMSL-library opgeroepen worden, moet in het execute commando aangegeven worden waar deze subroutine gevonden kan worden. Hiertoe moet het CSMP.CCL-file uitgebreid worden met DSK:IMSL.REL/SEARCH,. Dit gebeurt op de volgende wijze.

Copieer de CSMP.CCL-file uit het SYS-gebied naar het eigen schijfgebied:

```
.COPY CSMP.CCL=SYS:CSMP.CCL
```

Voeg vervolgens m.b.v. Teco DSK:IMSL.REL/SEARCH in zodat de nieuwe file wordt:

```
DSK:UPDATE.REL,DSK:IMSL.REL/SEARCH,SYS:CSMP00.REL,SYS:CSMP02.REL,SYS:CSMP03.REL/SEARCH,SYS:CSMP03.REL/SEARCH,SYS:PLT.REL/SEARCH,DSKD:TCSID.REL [5,11]/SEARCH,DSK:MACRO.REL [50,76]  
/RUNAME:CSMP/GO/EXECUTE
```

Omdat deze file op het eigen schijfgebied staat, wordt deze niet automatisch gebruikt bij de executie van het programma. Om dit te bewerkstelligen, wordt de executie afgebroken met dubbel .C tijdens het link loading, waarna vervolgens de executie wordt voortgezet met de gewijzigde CSMP.CCL-file door de commando's:

```
R LINK
```

```
*CSMP.CCL
```

Appendix 2

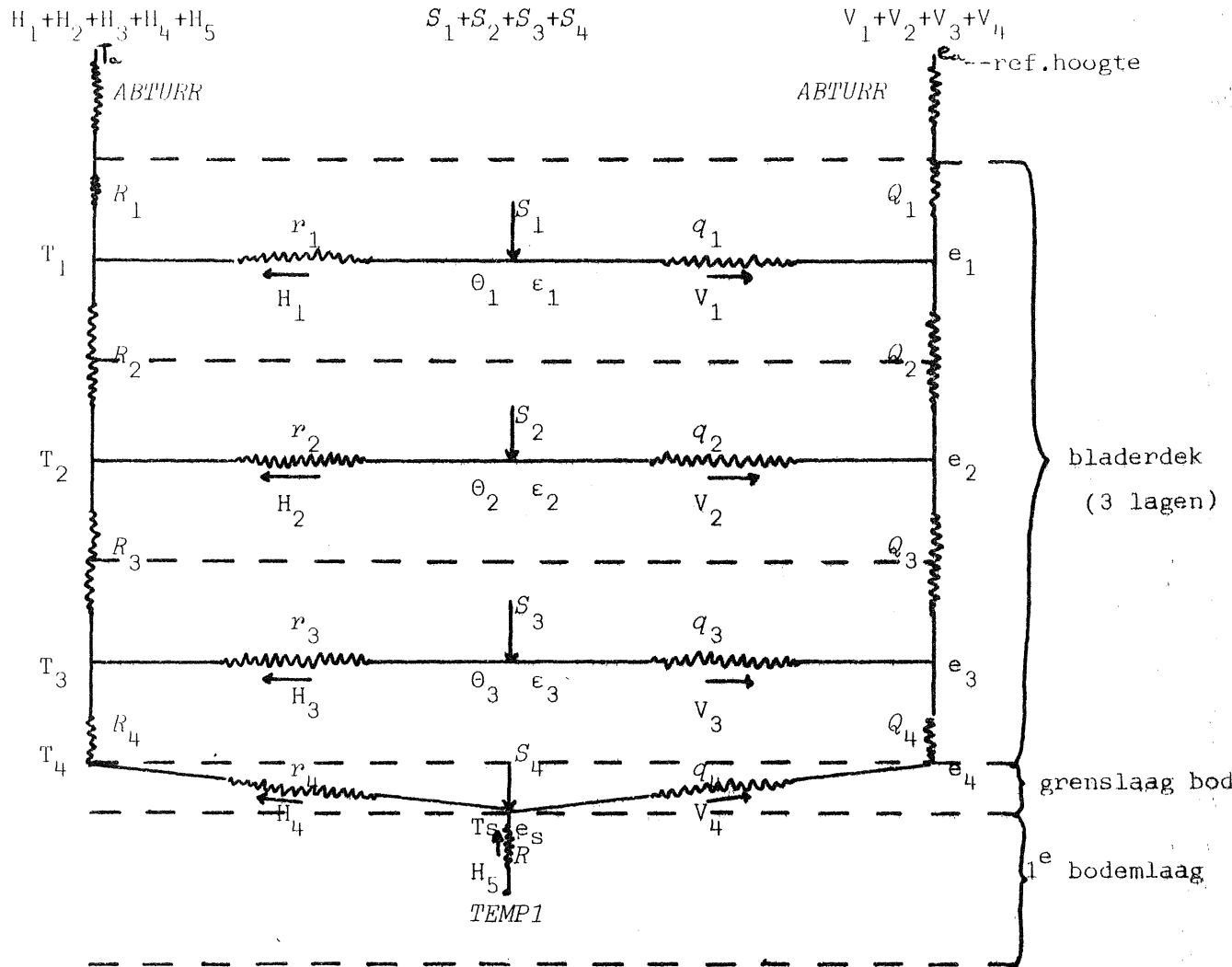
Enige opmerkingen bij het gebruik van LEQT1F

Een matrix-vergelijking van de vorm $A \cdot X = B$ oplossen kan m.b.v. de subroutine LEQT1F uit het IMSL-pakket. De handleiding uit het IMSL manual van het rekencentrum is in deze appendix opgenomen.

Doordat zowel matrix A als B in de output overschreven worden, moeten deze in de iteratieprocedure steeds opnieuw berekend worden.

Wanneer voor de simulatie het gewas is ingedeeld in een groot aantal lagen, ontstaat er een matrix met veel nullen. Het is zeer waarschijnlijk dat het gebruik van de subroutine LEQT1B dan goedkoper is. Deze transformeert de coëfficiënten-matrix zo, dat de nulplaatsen er voor het grootste gedeelte invallen zodat er minder geheugenruimte nodig is.

Figuur 1 Doorsnede door het gewas



- Bekende grootheden:

S : instraling per laag

$ABTURR$: turbulentieweerstand tussen bovenkant v.h.gewas en ref.niveau

R : turbulentieweerstand voor warmtetransport

Q : turbulentieweerstand voor damptransport

$TEMP1$: temperatuur 1^e bodemlaag

r : grenslaagweerstand voor warmtetransport

q : som grenslaag- plus bladweerstand damptransport

- Te berekenen grootheden

H : energieflex convectief warmtetransport per laag

V : energieflex door verdamping per laag

T : luchttemperatuur

T_s : oppervlaktetemperatuur bodem

e : dampspanning van de lucht

e_s : oppervlaktedampspanning van de bodem (verzadigd)

θ : bladtemperatuur

ϵ : dampspanning in het blad (verzadigd)

Figuur 2a Matrix A

$\frac{2 * \text{Cond} * \gamma}{d * \rho C_p * \Delta_4} * Q_1$	$\frac{2 * \text{Cond} * \gamma}{d * \rho C_p * \Delta_4} (Q_1 + Q_2)$	$\frac{2 * \text{Cond} * \gamma}{d * \rho C_p * \Delta_4} (Q_1 + Q_2 + Q_3)$	$\frac{2 * \text{Cond} * \gamma}{d * \rho C_p * \Delta_4} (Q_1 + Q_2 + Q_3 + Q_4 + q_4)$	$\frac{2 * \text{Cond} * \gamma}{d * \rho C_p * \Delta_4} (Q_1 + Q_2 + Q_3 + Q_4 + q_4) - 1$
$-\frac{\gamma}{R_4 \Delta_4} * Q_1$ $+\frac{\gamma}{R_4 \Delta_3} * Q_1$	$-\frac{\gamma}{R_4 \Delta_4} (Q_1 + Q_2)$ $+\frac{\gamma}{R_4 \Delta_3} (Q_1 + Q_2)$	$-\frac{\gamma}{R_4 \Delta_4} (Q_1 + Q_2 + Q_3)$ $+\frac{\gamma}{R_4 \Delta_3} (Q_1 + Q_2 + Q_3 + q_3)$ $+\frac{\gamma}{R_4 \Delta_2}$	$-\frac{\gamma}{R_4 \Delta_4} (Q_1 + Q_2 + Q_3 + Q_4 + q_4)$ $+\frac{\gamma}{R_4 \Delta_3} (Q_1 + Q_2 + Q_3)$ $-\frac{\gamma}{R_4} - 1$	$+\frac{\gamma}{R_4 \Delta_4} (Q_1 + Q_2 + Q_3 + Q_4 + q_4)$ $-\frac{\gamma}{R_4 \Delta_3} (Q_1 + Q_2 + Q_3)$
$-\frac{\gamma}{R_3 \Delta_3} Q_1$ $+\frac{\gamma}{R_3 \Delta_2} Q_1$	$-\frac{\gamma}{R_3 \Delta_3} (Q_1 + Q_2)$ $+\frac{\gamma}{R_3 \Delta_2} (Q_1 + Q_2 + q_2)$ $+\frac{\gamma}{R_3} - 1$	$-\frac{\gamma}{R_3 \Delta_3} (Q_1 + Q_2 + Q_3 + q_3)$ $+\frac{\gamma}{R_3 \Delta_2} (Q_1 + Q_2)$ $-\frac{\gamma}{R_3} - 1$	$-\frac{\gamma}{R_3 \Delta_3} (Q_1 + Q_2 + Q_3)$ $+\frac{\gamma}{R_3 \Delta_2} (Q_1 + Q_2)$ $- 1$	$+\frac{\gamma}{R_3 \Delta_3} (Q_1 + Q_2 + Q_3)$ $-\frac{\gamma}{R_3 \Delta_2} (Q_1 + Q_2)$
$-\frac{\gamma}{R_2 \Delta_2} Q_1$ $+\frac{\gamma}{R_2 \Delta_1} (Q_1 + q_1)$ $+\frac{\gamma}{R_2} - 1$	$-\frac{\gamma}{R_2 \Delta_2} (Q_1 + Q_2 + q_2)$ $+\frac{\gamma}{R_2 \Delta_1} (Q_1)$ $-\frac{\gamma}{R_2} - 1$	$-\frac{\gamma}{R_2 \Delta_2} (Q_1 + Q_2)$ $+\frac{\gamma}{R_2 \Delta_1} Q_1$ $- 1$	$-\frac{\gamma}{R_2 \Delta_2} (Q_1 + Q_2)$ $+\frac{\gamma}{R_2 \Delta_1} Q_1$ $- 1$	$+\frac{\gamma}{R_2 \Delta_2} (Q_1 + Q_2)$ $-\frac{\gamma}{R_2 \Delta_1} Q_1$
$-\frac{\gamma}{R_1 \Delta_1} (Q_1 + q_1)$ $-\frac{\gamma}{R_1} - 1$	$-\frac{\gamma}{R_1 \Delta_1} (Q_1)$ $- 1$	$-\frac{\gamma}{R_1 \Delta_1} Q_1$ $- 1$	$-\frac{\gamma}{R_1 \Delta_1} Q_1$ $- 1$	$+\frac{\gamma}{R_1 \Delta_1} Q_1$

Figuur 2b Matrix B. De open plaatsen bevatten de zelfde elementen als matrix A

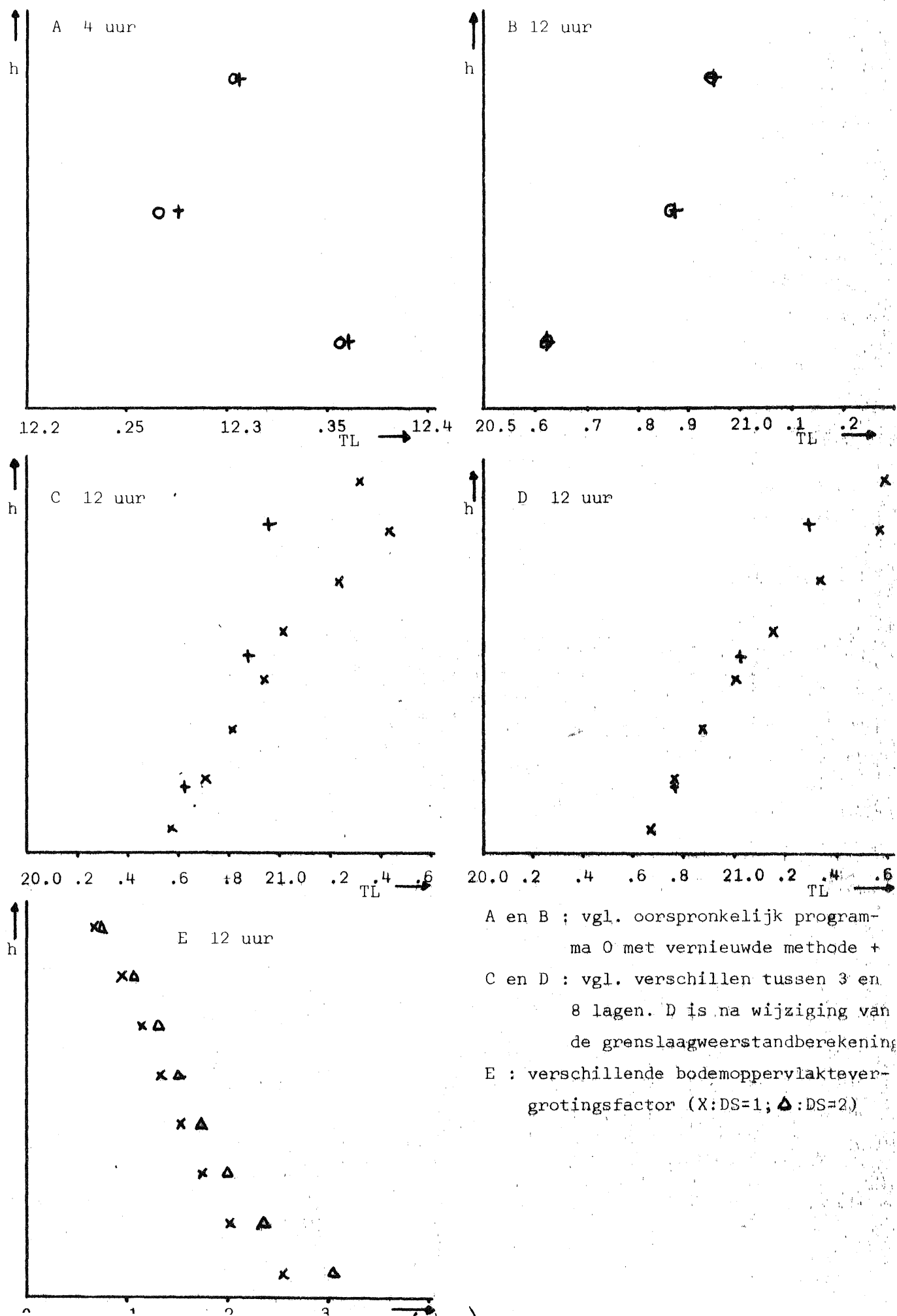
				$\frac{2 * \text{Cond} * (T_a - T_s - e_{sa} - e_a)}{d}$
				$\frac{\rho C_p (e_{sa} - e_a)}{R_4 \Delta_4} - \frac{e_{sa} - e_a}{\Delta_3}$
				$\frac{\rho C_p (e_{sa} - e_a)}{R_3 \Delta_3} - \frac{e_{sa} - e_a}{\Delta_2}$
				$\frac{\rho C_p (e_{sa} - e_a)}{R_2 \Delta_2} - \frac{e_{sa} - e_a}{\Delta_1}$
				$\frac{\rho C_p (e_{sa} - e_a)}{R_1 \Delta_1}$

Figuur 3 Coëfficiënten matrix A en vector B

$\frac{\Delta_1 r_1}{DL \cdot R_1} + \frac{\Delta_1 r_1}{DL \cdot R_2} + \Delta_1$	$\frac{-\Delta_1 r_1}{DL \cdot R_2}$	0	0	$\frac{-q_1}{DL \cdot Q_2} - \frac{q_1}{DL \cdot Q_1} - 1$	$\frac{q_1}{DL \cdot Q_2}$	0	0
$\frac{-\Delta_2 r_2}{DL \cdot R_2}$	$\frac{\Delta_2 r_2}{DL \cdot R_2} + \frac{\Delta_2 r_2}{DL \cdot R_3} + \Delta_2$	$\frac{-\Delta_2 r_2}{DL \cdot R_3}$	0	$\frac{q_2}{DL \cdot Q_2}$	$-\frac{q_2}{DL \cdot Q_3} - \frac{q_2}{DL \cdot Q_2} - 1$	$\frac{q_2}{DL \cdot Q_3}$	0
0	$\frac{-\Delta_3 r_3}{DL \cdot R_3}$	$\frac{\Delta_3 r_3}{DL \cdot R_3} + \frac{\Delta_3 r_3}{DL \cdot R_4} + \Delta_3$	$\frac{-\Delta_3 r_3}{DL \cdot R_4}$	0	$\frac{q_3}{DL \cdot Q_3}$	$-\frac{q_3}{DL \cdot Q_4} - \frac{q_3}{DL \cdot Q_3} - 1$	$\frac{q_3}{DL \cdot Q_4}$
0	0	$\frac{-\Delta_4 r_4}{DS \cdot R_4}$	$\frac{\Delta_4 r_4}{DS \cdot R_4} + \Delta_4$	0	0	$\frac{q_4}{DS \cdot Q_4}$	$-\frac{q_4}{DS \cdot Q_4} - 1$
$\frac{p_{CP}}{R_1} + \frac{p_{CP}}{R_2}$	$\frac{-p_{CP}}{R_2}$	0	0	$\frac{p_{CP}}{Q_1 \gamma} + \frac{p_{CP}}{Q_2 \gamma}$	$-\frac{p_{CP}}{Q_2 \gamma}$	0	0
$\frac{-p_{CP}}{R_2}$	$\frac{p_{CP}}{R_2} + \frac{p_{CP}}{R_3}$	$\frac{-p_{CP}}{R_3}$	0	$\frac{-p_{CP}}{Q_2 \gamma}$	$\frac{p_{CP}}{Q_2 \gamma} + \frac{p_{CP}}{Q_3 \gamma}$	$\frac{-p_{CP}}{Q_3 \gamma}$	0
0	$\frac{-p_{CP}}{R_3}$	$\frac{p_{CP}}{R_3} + \frac{p_{CP}}{R_4}$	$\frac{-p_{CP}}{R_4}$	0	$\frac{-p_{CP}}{Q_3 \gamma}$	$\frac{p_{CP}}{Q_3 \gamma} + \frac{p_{CP}}{Q_4 \gamma}$	$\frac{-p_{CP}}{Q_4 \gamma}$
0	0	$\frac{-p_{CP}}{R_4} - \frac{r_4}{R_4} \cdot \frac{Cond}{\frac{1}{2}d}$	$\frac{p_{CP}}{R_4} + \frac{(1+r_4)}{R_4} \cdot \frac{Cond}{\frac{1}{2}d}$	0	0	$\frac{-p_{CP}}{Q_4 \gamma}$	$\frac{p_{CP}}{Q_4 \gamma}$

$T_a \left(\frac{\Delta_1 r_1}{DL \cdot R_1} + \Delta_1 \right) - \frac{q_1}{DL \cdot Q_1} e_a - e_{sa}$
$\Delta_2 T_a - e_{sa}$
$\Delta_3 T_a - e_{sa}$
$\Delta_4 T_a - e_{sa}$
$S_1 + T_a \frac{p_{CP}}{R_1} + e_a \cdot \frac{p_{CP}}{Q_1 \cdot \gamma}$
S_2
S_3
$S_4 + T_5 \cdot \frac{Cond}{\frac{1}{2}d}$

Figuur 4 Berekende bladtemperaturen voor een maïsgewas



A en B : vgl. oorspronkelijk programma 0 met vernieuwde methode +
 C en D : vgl. verschillen tussen 3 en 8 lagen. D is na wijziging van de grenslaagweerstandberekening
 E : verschillende bodemoppervlaktevergrotingsfactor (X:DS=1; Δ:DS=2)

IMSL ROUTINE NAME - LEQTLF

PURPOSE - LINEAR EQUATION SOLUTION - FULL STORAGE
MODE - SPACE ECONOMIZER SOLUTION.

USAGE - CALL LEQTLF (A,M,N,IA,B,IDGT,WKAREA,IER)

ARGUMENTS

- A - INPUT MATRIX OF DIMENSION N BY N CONTAINING THE COEFFICIENT MATRIX OF THE EQUATION $AX = B$.
ON OUTPUT, A IS REPLACED BY THE LU DECOMPOSITION OF A ROWWISE PERMUTATION OF A.
- M - NUMBER OF RIGHT-HAND SIDES. (INPUT)
- N - ORDER OF A AND NUMBER OF ROWS IN B. (INPUT)
- IA - ROW DIMENSION OF A AND B EXACTLY AS SPECIFIED IN THE DIMENSION STATEMENT OF THE CALLING PROGRAM. (INPUT)
- B - INPUT MATRIX OF DIMENSION N BY M CONTAINING RIGHT-HAND SIDES OF THE EQUATION $AX = B$.
ON OUTPUT, THE N BY M SOLUTION X REPLACES B.
- IDGT - INPUT OPTION.
IF IDGT IS GREATER THAN 0, THE ELEMENTS OF A AND B ARE ASSUMED TO BE CORRECT TO IDGT DECIMAL DIGITS AND THE ROUTINE PERFORMS AN ACCURACY TEST.
IF IDGT EQUALS ZERO, THE ACCURACY TEST IS BYPASSED.
- WKAREA - WORK AREA OF DIMENSION GREATER THAN OR EQUAL TO N.
- IER - ERROR PARAMETER. (OUTPUT)
 TERMINAL ERROR
 IER = 129 INDICATES THAT MATRIX A IS ALGORITHMICALLY SINGULAR. (SEE THE CHAPTER L PRELUDE).
 WARNING ERROR
 IER = 34 INDICATES THAT THE ACCURACY TEST FAILED. THE COMPUTED SOLUTION MAY BE IN ERROR BY MORE THAN CAN BE ACCOUNTED FOR BY THE UNCERTAINTY OF THE DATA. THIS WARNING CAN BE PRODUCED ONLY IF IDGT IS GREATER THAN 0 ON INPUT. (SEE CHAPTER L PRELUDE FOR FURTHER DISCUSSION).

PRECISION/HARDWARE - SINGLE AND DOUBLE/H32
- SINGLE/H36,H48,H60

REQD. IMSL ROUTINES - LUDATF,LUELMF,UERTST,UGETIO

NOTATION - INFORMATION ON SPECIAL NOTATION AND CONVENTIONS IS AVAILABLE IN THE MANUAL INTRODUCTION OR THROUGH IMSL ROUTINE UHELP

Algorithm

LEQTLF solves the set of linear equations $AX=B$ for X , given the N by N matrix A (in full storage mode). The N by M solution X overwrites B . The chief advantage of this program is that it uses less space than LEQT2F.

This routine performs Gaussian elimination (Crout algorithm) with equilibration and partial pivoting.

See reference: Forsythe, George and Moler, Cleve B., Computer Solution of Linear Algebraic Systems, Englewood Cliffs, N. J., Prentice-Hall, Inc., 1967, Chapter 9.

Note: This routine uses a Crout algorithm for decomposition (as opposed to the elimination routine in the above reference).

Accuracy

If IDGT is greater than zero, elements of A are assumed to be correct to IDGT decimal digits. The solution X will be the exact solution - without any roundoff error - to a matrix $\bar{A}$ which agrees with A in the first IDGT decimal digits. If such a solution is not obtainable in the accuracy with which we are working, a warning is given (IER=34). If IDGT equals zero, this accuracy test is bypassed.

Example

This example solves the set of linear equations $AX=B$ for X where A is a 3 by 3 matrix and B is a 3 by 4 matrix. Since $IDGT \neq 0$, an accuracy test is performed.

Input:

```
REAL A(4,4),B(4,4),WKAREA(16)
INTEGER M,N,IA,IDGT,IER
```

```
N      =    3
M      =    4
IA     =    4
IDGT   =    3
```

```
A      =  [ 33.000    16.0    72.0    x ]
           [ -24.000   -10.0   -57.0    x ]
           [ - 8.000    - 4.0   -17.0    x ]
           [    x          x          x    x ]
```

```
B      =  [ 1.0      0.0      0.0   -359.0 ]
           [ 0.0      1.0      0.0    281.0 ]
           [ 0.0      0.0      1.0    85.0 ]
           [    x          x          x     x ]
```

```
CALL LEQT1F (A,M,N,IA,B,IDGT,WKAREA,IER)
```

```
.  
.  
.
```

```
END
```

Output:

```
IER    =    0
```

```
A      =
```

$$\begin{bmatrix} -8.0 & -4.0 & -17.0 & x \\ 3.0 & 2.0 & -6.0 & x \\ -4.1250 & -0.25 & 0.375 & x \\ x & x & x & x \end{bmatrix}$$

```
B      =
```

$$\begin{bmatrix} -9.6666 & -2.6666 & -32.0 & 1.0 \\ 8.0 & 2.5 & 25.5 & -2.0 \\ 2.6666 & .6666 & 9.0 & -5.0 \\ x & x & x & x \end{bmatrix}$$

Note: x indicates elements not used by LEQT1F.

Appendix 3Aanvulling van de symbolen lijst

ABSC	see ABSCL but for sunlit leaves
ABSCL	see ABSO but for a clear sky
ABSO	total absorbed energy per leaf area for an overcast sky
ABSS	see ABSC but for shaded leaves
DELTA	slope of saturated vapour pressure curve
DELTAM	DELTA in macro
DS	soil surface roughness factor
EPS	vapour pressure at leaf surface
HTOP	height at the top of a layer
IER	error parameter
L	runner in do-loop
LBR	resistance to vapour caused by boundary layer and leaf resistance
LEQT1F	name of IMSL- subroutine
LTL	last leaf temperature
M	runner in do-loop
MAT	coefficient matrix
N	runner in do-loop
NUML2	NUMLL plus two
NUML12	2 * NUML2
RAM	RA in macro
RRM	RR in macro
TADTM	TADT in macro
TEMP1	temperature of toplayer of the soil
TL4	TL in Kelvin raised to the power 4
TM	mean of leaf temperature and air temperature
TSM	mean of soil temperature and air temperature
VEC	unknown vector to be solved
VPAC	vapour pressure of air in canopy
WKAREA	work area for subroutine
X	runner in iteration

TITLE MICROWEATHER SIMULATION

***** SECTION 1 *****

* FIXED AND ARRAY DECLARATIONS

FIXED N,I,NUMLL,NUML1,NUML2,IS,IL,SN,L,ISUN,J,INVL,IER,M
FIXED NUML2

/ DIMENSION CO2(20),CO2FL(21),RINC(21)
/ DIMENSION DEWL(9),NCO2A(9),NLWR(21),NLWRM(9,9)
/ DIMENSION WIND(21),HTOP(21),NSF(20),NLF(20),MAT(20,20),VEC(20)
/ DIMENSION S(9,10),Z(9,10),LHLL(15),SHLL(15),ZA(5),ZS(5)
/ DIMENSION ABSRAD(10),RA(10),PR(10)
/ DIMENSION TADT(10),VPAC(10),DT(10),DV(10),EPS(10),LBR(10),TM(10)
/ DIMENSION DELTA(10),LTL(10),WKAREA(400),IDT(10),IDV(10)

STORAGE K(21),TL(15),TCOND(20),VCOND(20),ITCAP(20),IVCAP(20)
STORAGE LHFL(21),DIK(20),SHFL(21),RFN(11),KDN(11),KB(11),TCOM(10)
STORAGE HFL(11),TEMP(10),DIST(11),RCHI(9),TI(10)
STORAGE F(9),BU(9),GAV(9),RFV(11),KDV(11)

***** SECTION 2 *****

* MACRO FOR NUMERICAL INTEGRATION OF THE INVERSE OF
* THE EXCHANGE COEFFICIENT WITH HEIGHT

MACRO INT,A,B,C,D,E=NUMINT(COEF1,COEF2,POWER,ZS,MONOBL)

* SEE EQN 4.22 AND 4.23
F1=COEF1*(1.-COEF2*ZS(1)/MONOBL)**POWER
F2=COEF1*(1.-COEF2*ZS(2)/MONOBL)**POWER
F3=COEF1*(1.-COEF2*ZS(3)/MONOBL)**POWER
F4=COEF1*(1.-COEF2*ZS(4)/MONOBL)**POWER
F5=COEF1*(1.-COEF2*ZS(5)/MONOBL)**POWER
A=F5
E=(F1-4.*F2+6.*F3-4.*F4+F5)/(24.*DZS4)
D=(F2-3.*F3+3.*F4-F5)/(6.*DZS3)-6.*E*DZS
C=(F3-2.*F4+F5)/(2.*DZS2)-3.*D*DZS-7.*E*DZS2
B=(F4-F5)/DZS-C*DZS-D*DZS2-E*DZS3
* SEE EQN 4.42
INT=A*INTN+B*Z1S+C*Z2S+D*Z3S+E*Z4S

ENDMAC

* MACRO DESCRIBING THE CO2 ASSIMILATION, TRANSPIRATION
* AND LEAF TEMPERATURE OF THE INDIVIDUAL LEAVES

MACRO ENL,SHL,TL,NCO2A,LRES,ABSRAD= ...
TRPH(VIS,NIR,LWR)
ABSRAD=VIS+NIR+LWR
AMAX =AMAX1(AFGEN(AMTB,TADTM),0.001)
* SEE EQN 3.8
NCO2A =(AMAX+DPL)*(1.-EXP(-VIS*EFF/AMAX))-DPL
CO2F =68.4*(ECO2C-RCO2I)
* SEE EQN 3.9

```

SRESL =CO2F/(AMAX1(.001,NCO2A)*1.66)-RAM*0.783
IF (SRESL.GT.SRW) GO TO 700
SRESL =SRW
NCO2A =AMIN1(CO2F/((1.66*SRW+1.3*RAM),NCO2A)
700 LRES =RESCW*SRESL/((SRESL+RESCW)
IF (DEW(1).GT.0. .OR. LHL(1).LT.0.) LRES=0.
ENP =0.3*NCO2A
* SEE EQN 3.11
EHL =(DELTAM*(ABSRAD=ENP)+DRYP)/(PSCH*(RAM*0.93+LRES)/RAM+DELTAM)
SHL =ABSRAD-EHL-ENP
TL =TADTM+SHL*RRM
ENDMAC

```

***** SECTION 3 *****

* FUNCTIONS, PARAMETERS AND TABLES

```

FUNCTION SRADTB=0.,-1.E6, 1.E6,-1.E6
FUNCTION NRADTB=0.,-1.E6, 1.E6,-1.E6

```

***** USER MUST DEFINE THE FOLLOWING FUNCTIONS, PARAMETERS AND TABLES ****

* OBSERVED WEATHER DATA AS A FUNCTION OF HOUR OF THE DAY IN STANDARD SOLAR

```

FUNCTION TATB = 0.,13.5, 6.,12.9, 10.,16.5, 14.,20., 16.,20.6, ...
18.,14.1, 24.,13.5
FUNCTION VPATB =0.,13., 6.,12., 8.5,13., 13.,11.2, 18.,15., 24.,13.
FUNCTION WINDRB=0.,.7, 7.,.7, 8.,.7, 9.,2., 11.,3., 16.,1., 17.,.7, ...
24.,.7
FUNCTION NRADTB=0.,-40., 6.,-30., 6.5,0., 8.5,105., 9.8,400., ...
11.2,660., 12.1,690., 13.2,630., 14.1,530., 15.1,380., ...
16.1,200., 17.1,14., 18.1,-84., 24.,-40.

```

```

PARAM ECO2C=330.
PARAM CROPHT=2.5,REFHT=3.
PARAM DFACT=0.8,ZNOTF=0.07
* ZNOTF RATIO OF ROUGHNESS LENGTH AND CROP HEIGHT
PARAM LAI=3.73
PARAM LAT=45.,DLONG=0.
PARAM DAY=270.
PARAM START=0.

```

*** PHYSICAL DATA

```

PARAM GR=9.81
CONST SIGMA=5.6696E-8
CONST PI=3.1415927
PARAM KARMAN=0.35
PARAM RHOC=1240.,PSCH=0.67
PARAM LHVAP=2.5E9
PARAM SCOF=4.7

```

```

FUNCTION FRDIFT=0.,1., 5.,1., 15.,0.323, 25.,.219, 35.,.176, ...
45.,.16, 55.,.145, 65.,.136, 90.,.133

```

```

TABLE BU(1-9)=.03,.087,.133,.163,.174,.163,.133,.087,.03

```

*** PLANT DATA

PARAM SCN=0.85,SCV=0.2
 PARAM WIDTH=0.05
 FUNCTION AMTB=10.,.001, 15.,25., 20.,50., 25.,60., 35.,60., 40.,20.
 PARAM PCO2I=90.
 PARAM EFF=0.62,RESCW=2000.
 FUNCTION SRNTB=.5,1.E4, 0.6,3.E3, 0.7,800., 0.8,600., 0.9,130., 1.5,130.
 FUNCTION WSTTB=.5,-50., .7,-17., .8,-14., .84,-12.5, .88,-10., ...
 .90,-8.1, 1.,0., 1.5,40.5
 TABLE F(1-9)=9*0.
 ** UNLESS SPECIFIED BY USER, A SPHERICAL LEAF ANGLE DISTRIBUTION IS ASSUMED
 FUNCTION HTB=0.,0., 100.,0.
 ** IF HTB IS NOT SPECIFIED BY USER, A PARABOLIC DISTRIBUTION OF LEAF
 * AREA DENSITY WITH HEIGHT IS ASSUMED BY THE FOLLOWING FUNCTION
 FUNCTION RHTB=0.,0., 0.028,0.1, 0.104,0.2, 0.216,0.3, 0.352,0.4, ...
 0.5,0.5, 0.648,0.6, 0.784,0.7, 0.896,0.8, 0.972,0.9, 1.,1.
 PARAM WRESPL=10.E6
 PARAM SCRS=3.5E-2
 FUNCTION TREDTB=0.,.08, 10.,.08, 20.,.29, 33.,.94, 37.,1., 48.,0.87
 PARAM WSTSL=-0.1
 PARAM DRAGC=0.3

*** SOIL DATA, SOIL IS ALWAYS ASSUMED AT FIELD CAPACITY

PARAM DS=2.
 PARAM TCOM1=0.02
 PARAM MULT=1.2
 PARAM LAMBDA=1.3
 PARAM HRES=0.05
 PARAM PRESS=0.
 ** WHEN SOIL SURFACE IS WET, THIS RESISTANCE IS ZERO,
 * WHEN IT IS DRY, THIS RESISTANCE IS VERY LARGE, E.G., 1.E6
 PARAM VHCAP=2.E6
 PARAM SRESP=10.
 TABLE TI(1-10)=12.2,12.9,13.6,14.3,15.0,15.5,15.7,15.7,15.4,15.2
 * TI INITIAL TEMPERATURE OF THE DIFFERENT SOIL LAYERS

*** CONTROL VARIABLES

PARAM NUMLL=3
 * NUMLL NUMBER OF LAYERS IN WHICH CANOPY IS DIVIDED
 ** CAUTION: THIS NUMBER APPEARS IN WRITE STATEMENTS SEVERAL TIMES.
 * AND IN END NUMBER OF INTEGRAL STATEMENTS OF DT,DV,DEW,DEWT.
 * ADVICE: NUMLL=3 IS SUFFICIENTLY ACCURATE FOR TOTAL FLUXES.
 * SPECIFY TIMER CARD IN SECONDS (ONE DAY=86400 SECONDS)
 TIMER TIME=0.,FINTIM=43200.,PRDEL=3600.,OUTDEL=10800.,DELT=480.
 METHOD RECT

INITIAL

SECTION 4

* STRATIFICATION IN AIR AND SOIL

* ASSIGNING INITIAL VALUES TO SOME CONTROL VARIABLES

PROCD WCCPI,SRADF=CONTR(LAI)

WCCPI=2.5E-3*LAI*0.975

DO 72 I=1,NUMLL

IDT(I)=4.

IDV(I)=-2.

DT(I)=IDT(I)

DV(I)=IDV(I)

IDEW(I)=0.

72 IDEWT(I)=0.

TEMP1=TI(1)

TS=TI(1)

SRADF =1.

TEF0 =0.

TELN =0.

ENDPRO

* AERODYNAMICS AND STRATIFICATION ABOVE THE CANOPY

PROCD D,ZNOT,ZS1,ZA1,DZS,INTN,INTAN=ABL(SRADF)

* BETTER FORMULATIONS FOR D AND ZNOT GIVEN IN SECTION 4.4(LATER DEVELOP)

D =DFACT*CROPHT

ZNOT =0.1*INSW(CROPHT-1.,CROPHT,SQRT(CROPHT))

ZA1 =REFHT-D

ZA(1)=ZA1

ZA(5)=CROPHT-D

Z1A =ZA1-ZA(5)

Z2A =(ZA1*ZA1-ZA(5)*ZA(5))*0.50

Z3A =(ZA1**3-ZA(5)**3)/3.

Z4A =(ZA1**4-ZA(5)**4)*0.25

ZS1 =ZA1

ZS(1)=ZS1

ZS(5)=ZNOT

Z1S =ZS1-ZS(5)

Z2S =(ZS1*ZS1-ZS(5)*ZS(5))*0.50

Z3S =(ZS1**3-ZS(5)**3)/3.

Z4S =(ZS1**4-ZS(5)**4)*0.25

DZS =Z1S*0.25

DZS2 =DZS*DZS

DZS3 =DZS*DZS2

DZS4 =DZS*DZS3

DO 40 J=2,4

ZA(J)=ZA1-(J-1)*Z1A*0.25

40 ZS(J)=ZS1-(J-1)*DZS

INTN=ALOG(ZS1/ZS(5))

INTAN=ALOG(ZA1/ZA(5))

ENDPRO

* AERODYNAMICS AND STRATIFICATION IN THE CANOPY

PROCD NUML1,DL,LMIX,ALPHA=LAYER(D,NUMLL)

NUML1=NUMLL+1

NUML12=NUML1*2

NUML2=NUMLL+2

DL=LAI/NUMLL

* ACCORDING TO EQN 4.44 A FACTOR 4/PI MUST BE ADDED (LATER DEVELOP)
LMIX=SQRT(WIDTH*CROPHT/LAI)

```

* SEE EQN 4.49 , ADDITION OF IW IS LATER DEVELOPMENT
ALPHAK=SQRT(DRAGC*LAI*CROPHT/(2.*LMIX))
* IF NO MEASURED LEAF AREA DISTRIBUTION IS AVAILABLE (HTB), A ...
* PARABOLIC ONE (RHTB) IS ASSUMED
IF (AFGEN(HTB,100.).EQ.0.) GO TO 22
  DO 20 I=1,NUMLL
20  HTOP(I)=AFGEN(HTB,LAI-(I-1)*DL)
  GO TO 21
22  DO 23 I=1,NUMLL
23  HTOP(I)=CROPHT*AFGEN(RHTB,1.-(I-1.)/NUMLL)
21  HTOP(NUMLL)=0.
  DO 24 I=1,NUMLL
    DIK(I)=HTOP(I)-HTOP(I+1)
    ITCAP(I)=1./(DIK(I)*RHOCF)
24  IVCAP(I)=ITCAP(I)*PSCH
ENDPRO

```

```

* INITIALIZATION AND STRATIFICATION IN THE SOIL
PROCD DIST1=SOILI(XL)
  DIST1 =0.5*TCOM1
  DIST(1)=DIST1
  TCOM(1)=TCOM1
  HCI(1) =TCOM1*VHCAP*TI(1)
  DO 70 I=2,10
    TCOM(I)=TCOM(I-1)*MULT
    DIST(I)=0.5*(TCOM(I)+TCOM(I-1))
70  HCI(I) =TI(I)*TCOM(I)*VHCAP
ENDPRO

```

***** SECTION 5 *****

* RADIATION CHARACTERISTICS

```

* AVERAGE PROJECTION OF LEAVES (OAV)
* AND LIGHT DISTRIBUTION FUNCTIONS Z AND S , ACCORDING TO APPENDIX B
PROCD SUMF,ZISSN,RAD,DEC,SSIN,CCOS=GEOM(DL)
  RAD =PI/180.
  DEC =-23.45*COS(PI*(DAY+10.173)/182.621)
  COSDEC=COS(RAD*DEC)
  SINDEC=SIN(RAD*DEC)
  SNLT=SIN(RAD*LAT)
  CSLT=COS(RAD*LAT)
  SSIN=SNLT*SINDEC
  CCOS=CSLT*COSDEC
  SUMF=F(1)+F(2)+F(3)+F(4)+F(5)+F(6)+F(7)+F(8)+F(9)
  IF (SUMF.EQ.0.) GO TO 90
  DO 64 IS=1,9
    FLIS=(10*IS-5)*RAD
    SI=SIN(FLIS)
    CO=COS(FLIS)
    DD=0.
    DO 60 IL=1,9
      FLIL=(10*IL-5)*RAD
      AA=SI*COS(FLIL)

```

```

      BB=CO*SIN(FLIL)
      CC=AA
      IF (IS.GE.IL) GO TO 36
      SQ=SQRT(BB*BB-AA*AA)
      CC=2.*(AA*ATAN(AA/SQ)+SQ)/PI
      * CC IS 0 (SEE EQN 2.3), AND DD IS OAV (SEE EQN 2.4)
36      DD=DD+CC*F(IL)
      DO 56 SN=1,9
      FLSN=SN/10.
      FA=FLSN-AA
      CC=1.
      IF (IS.LT.IL) GO TO 57
      IF (FLSN=BB.GE.AA) GO TO 56
      IF (FLSN=BB.GT.AA) GO TO 61
      CC=0.
      GO TO 56
61      SQ=SQRT(BB*BB-FA*FA)
      CC=ATAN(FA/SQ)/PI+0.5
      GO TO 56
57      IF (FLSN=AA.GE.BB) GO TO 56
      IF (FLSN=AA.GT.BB) GO TO 61
      SQ=SQRT(BB*BB-FA*FA)
      CC=ATAN(FA/SQ)
      FA=FLSN-AA
      SQ=SQRT(BB*BB-FA*FA)
      CC=(ATAN(FA/SQ)+CC)/PI
56      S(IL,SN)=CC
60      S(IL,10)=1.
      EE=0.
      DO 63 SN=1,10
      CC=0.
      DO 62 IL=1,9
62      CC=CC+F(IL)*S(IL,SN)
      Z(IS,SN)=CC-EE
63      EE=CC
64      OAV(IS)=DD
      GO TO 91

      *      A SPHERICAL LEAF ANGLE DISTRIBUTION IS ASSUMED
90      ZISSN=0.1
      DO 92 IS=1,9
92      OAV(IS)=0.5
91      CONTINUE
ENDPRO

```

```

      *      SIMPLIFIED EQUATIONS FOR REFLECTION AND EXTINCTION
      *      COEFFICIENTS UNDER DIRECT RADIATION
      PROCED REFV,REFN=EXTI(RAD)
      SQNI=SQRT(1.-SCN)
      SQV =SQRT(1.-SCV)
      REFV=(1.-SQV)/(1.+SQV)
      REFN=(1.-SQNI)/(1.+SQNI)
      *      SEE EQN 2.21
      DO 65 IS=1,9
      KB(IS+1)=OAV(IS)/SIN((10*IS-5)*RAD)
      *      SEE EQN 2.34
      KDN(IS+1)=KB(IS+1)*SQNI+0.94623+0.03533
65      KDV(IS+1)=KB(IS+1)*SQV +0.94623+0.03533
      *      SEE EQN 2.42

```

```

KB(1)=KB(2)
KB(11)=KB(10)
KDV(1) =KDV(2)
KDV(11) =KDV(10)
KDN(1) =KDN(2)
KDN(11) =KDN(10)
DO 66 IS=1,11
  RFN(IS)=REFN*2.*KB(IS)/(KB(IS)+1.)
  RFV(IS)=PEFV*2.*KB(IS)/(KB(IS)+1.)
* SEE EQN 2.45
  RFV(IS)=AMAX1(0.,1.117*(1.-EXP(-RFV(IS) ) )=0.0111)
66  RFN(IS)=AMAX1(0.,1.117*(1.-EXP(-RFN(IS) ) )=0.0111)
* SEE EQN 2.46
ENDPRO

```

```

* REFLECTION AND EXTINCTION OF DIFFUSE RADIATION(SEE EQN 2.37)
PROCED XL,XVDF,XNDF,RFOVV,RFOVN=EXDIF(REFV)
RFOVV=0.
RFOVN=0.
SUMBL=0.
SUMVI=0.
SUMNI=0.
DO 67 J=1,9
  RFOVV=RFOVV+BU(J)*RFV(J+1)
  RFOVN=RFOVN+BU(J)*RFN(J+1)
  SUMBL=SUMBL+BU(J)*EXP(-KB(J+1)*LAI)
  SUMVI=SUMVI+BU(J)*EXP(-KDV(J+1)*LAI)
67  SUMNI=SUMNI+BU(J)*EXP(-KDN(J+1)*LAI)
KBDF=-ALOG(SUMBL)/LAI
KDFV=-ALOG(SUMVI)/LAI
KDFN=-ALOG(SUMNI)/LAI
XNDF=EXP(-DL*KDFN)
XVDF=EXP(-DL*KDFV)
XL=EXP(-DL*KBDF)
XLN=XL**NUMDL
* SEE EQN 2.41
ENDPRO

```

DYNAMIC

***** SECTION 6 *****

* CONTROL OF EXECUTION

* TIMING AND PROGRAM CONTROL

```

PROCED HOUR,SRADM,NRADM,DIFTL,DIFDT=CONTR1(TIME)
HOUR=AMOD(TIME/3600,+START,24.)
SRADM=AFGEN(SRADTB,HOUR)
NRADM=AFGEN(NRADTB,HOUR)

```

ENDPRO

```

PROCED ENLOSS=CONTR2(DT1,DV1,TELP0,RSS,RSSE)

```

```
PRT=IMPULS(0.,PRDEL)
```

```
ENLOSS=SHFL(1)+LNFL(1)+G+0.3*TNC02A
```

```
ENDPRO
```

```
*****
```

SECTION 7

```
*****
```

```
* WEATHER CONDITIONS ABOVE THE CANOPY
```

```
* OBSERVED WEATHER CONDITIONS
```

```
PROCD SVPA,TA,TABS,VPD,DELTAM,WINDR,WINDR2,CT=WEER(HOUR,DZS)
```

```
VPA =AFGEN(VPATB,HOUR)
```

```
TA =AFGEN(TATB,HOUR)
```

```
TA239=TA+239.
```

```
TABS =TA+273.
```

```
SVPA =6.11*EXP(17.4*TA/TA239)
```

```
DELTAM=4158.6*SVPA/(TA239*TA239)
```

```
VPD =SVPA-VPA
```

```
WINDR=AFGEN(WINDRB,HOUR)
```

```
WINDR2=WINDR*WINDR
```

```
CT=INTN*WINDR2*TABS/(GR*4.*DZS)
```

```
ENDPRO
```

```
* CALCULATION OF SUN HEIGHT
```

```
PROCD HSUN,SNHS,SNHSS=GLOBE(HOUR,RAD,DEC,SSIN,CCOS,SRW)
```

```
* SEE EQN 2.107
```

```
SNHSS=SSIN+CCOS*COS(RAD*(HOUR+12,-DLONG)*15.)
```

```
SNHS =AMAX1(0.,SNHSS)
```

```
HSUN =ATAN(SNHS/SQRT(1.-SNHS*SNHS) )/RAD
```

```
ENDPRO
```

```
* INTERPOLATION OF REFLECTION AND EXTINCTION COEFFICIENTS
```

```
PROCD RDRV,RDRN,KDRV,KDRN,KDR=INTERP(HSUN)
```

```
FISUN=(HSUN+15.)*0.1
```

```
ISUN =FISUN
```

```
IS=FISUN-0.5
```

```
FI =FISUN-ISUN
```

```
RDRV =RFV(ISUN) *(1.-FI)+RFV(ISUN+1) *FI
```

```
RDRN =RFN(ISUN) *(1.-FI)+RFN(ISUN+1) *FI
```

```
KDR =KB(ISUN) *(1.-FI)+KB(ISUN+1) *FI
```

```
KDRN =KDN(ISUN) *(1.-FI)+KDN(ISUN+1) *FI
```

```
KDRV =KDV(ISUN) *(1.-FI)+KDV(ISUN+1) *FI
```

```
ENDPRO
```

```
* CALCULATION OF FRACTION OVERCAST AND APPARENT SKY TEMPERATURE
```

```
PROCD DIFOV,DIFCL,SUNDCL,FOV,FCL,LFOV,LFCL,NRAD,SRAD,SKT,LWRI,TELN=...
```

```
CURRAD(KDR)
```

```
LASTDT=DT(1)
```

```
LASTDV=DV(1)
```

```
LASTTL=TL(1)
```

```
LASTNR=NRADH
```

```

LASTSF=SRADM
TELM=TELM+1.
* SEE TABLE 2.1
FRDIF =AFGEH(FRDIFT,HSUN)
DIFCL =580.*SNHS*FRDIF
DIFOV =116.*SNHS
SUNDCL=580.*SNHS*(1.-FRDIF)
NSRO =(1.-RFOVV+0.7*(1.-RFOVN))*DIFOV
NSRC =(2.-RDRV-RDRH)*SUNDCL+(2.-RFOVV-RFOVN)*DIFCL
SRC =2.*(SUNDCL+DIFCL)
SRO =1.7*DIFOV
SKTCL =1.2*TA-21.
* SEE EQN 2.67
SKTOV =TA-2.
SKTCL4=(SKTCL+273.)*4
SKTOV4=(SKTOV+273.)*4
LWRCI =SIGMA*(SKTCL4-ETS4)
LWROI =SIGMA*(SKTOV4-ETS4)
IF (SRADM.LT.-1000.) GO TO 300
LFOV =1.
LFCL =0.
SUNDCL=0.
DIFCL=0.
DIFOV=SRADM/1.7
IF (SNHS.EQ.0. .OR. SRADM.EQ.0.) GO TO 302
FOV=(SRC-SRADM)/(NOT(SRC-SRO)+SRC-SRO)
FCL=1.-FOV
LFOV=LIMIT(0.,1.,FOV)
LFCL=1.-LFOV
IF (LFOV.EQ.FOV) GO TO 302
SRADF =SRADM/INSW(LFOV-FOV,SRO,SRC)
SUNDCL=SRADF*SUNDCL
DIFCL =SRADF*DIFCL
DIFOV =SRADF*DIFOV
302 IF(NRADM.LT.-1000.) GO TO 304
310 LWRI =NRADM-LFCL*((2.-RDRV-RDRH)*SUNDCL+(2.-RFOVV-RFOVN)*
DIFCL)-LFOV*(1.7-RFOVV-0.7*RFOVN)*DIFOV
311 SKT =(ABS(LWRI/SIGMA+ETS4))*0.25-273.
GO TO 306
304 SKT =FOV*SKTOV +(1.-FOV)*SKTCL
SKT4 =(SKT+273.)*4
LWRI =SIGMA*(SKT4-ETS4)
GO TO 306
300 FOV =(NSRC+LWRCI-NRADM)/(NSRC-NSRO+LWRCI-LWROI)
FCL =1.-FOV
LFOV =LIMIT(0.,1.,FOV)
LFCL =1.-LFOV
LWRI =LFOV*LWROI+LFCL*LWRCI
NRADOV=NSRO+LWRI
NRADCL=NSRC+LWRI
IF (LFOV.EQ.FOV) GO TO 311
IF (SNHS.GT.0.) GO TO 309
LWRI =NRADM
GO TO 311
309 SRADF=(NRADM-LWRI)/INSW(LFOV-FOV,NSRO,NSRC)
IF(SRADF.GT.0.) GO TO 307
WRITE(6,807) SRADF
807 FORMAT(1H,20H WARNING SRADF IS E15.5,18H ,BUT SET AT ZERO /)
SRADF=0.
307 IF(SRADF.LT.2.) GO TO 308

```

```

WRITE(6,808) SRADF
808 FORMAT(1H ,20H WARNING  SRADF IS  E15.5,18H ,BUT SET AT TWO /)
SRADF=2.
308 SUNDCL=SUNDCL*SRADF
DIFCL =DIFCL *SRADF
DIFOV =DIFOV *SRADF
GO TO 310
306 NRADCL=(2.-RDRV-RDRN)*SUNDCL+(2.-RFOVV-RFOVN)*DIFCL+LWRI
NRADOV=(1.-RFOVV+0.7*(1.-RFOVN) )*DIFOV+LWRI
NRAD =LFOV*NRADOV + LFCL*NRADCL
SRAD =LFOV*DIFOV*1.7+2.*LFCL*(SUNDCL+DIFCL)
SKT4 =(SKT+273.)**4
ENDPRO

```

***** SECTION 8 *****

* WATER STATUS OF THE CANOPY (SECTION 3.4.4)

```

PROCED SRW,RWCP,WSTCP=STRESS(CT)
RWCP =WCCP/(LAI*2.5E-3)
WSTCP=AFGEN(WSTTB,RWCP)
SRW =AFGEN(SRWTB,RWCP)
ENDPRO

```

* WATER BALANCE OF THE CANOPY

```

PROCED WLOSS,WUPTCP,WCCPRT,TRZ,ACRS=WABAL(WSTCP,TS)
WLOSS =(LHFL(1)-LHFL(NUML1))/LHVAP
TRZ =(TEMP(3)+TEMP(4)+TEMP(5) )/3.
ACRS =SCRS*1.E-6*AFGEN(TREDTB,TRZ)
WUPTCP=(WSTSL-WSTCP)/(WRESPL+1./ACRS)
WCCPRT=WUPTCP-WLOSS
WCCP =INTGRL(WCCPI,WCCPRT)
ENDPRO

```

***** SECTION 9 *****

* TURBULENCE AND WIND

* ABOVE VEGETATION AERODYNAMICS (SECTION 4.2)

```

PROCED RICHN,MONOBL,INTN,WINDC,USTAR,ABTURR,DZETA= ...
ABOVE(WINDR,TABS,INTN,INTAN,ZA1,DIFOV,SRW)
RICHN=-(REFHT-D-ZNOT)*GR*(LASTDT+0.1*LASTDV)/(WINDR2*TABS)
* SEE EQN 4.40
**** VOOR TIJDELIJKE VEREENVOLDIGING WORDT GESTELD RICHN=0
RICHN=0
IF (RICHN.GT.+0.001) GO TO 41
IF (RICHN.LT.=0.001) GO TO 43

```

* NEUTRAL CONDITIONS. (SECTION 4.2.1)

```

42 USTAR =WINDR*KARMAN/INTN
ABTURR=0.74*INTAN/(KARMAN*USTAR)
WINDC =WINDR -USTAR*INTAN/KARMAN
GO TO 50

```

* STABLE CONDITIONS WHEN RICHN IS POSITIVE.

```

41 AA =SCOF*(1.-SCOF*RICHN)
   IF (AA.LT.0.) GO TO 45
   BB =INTN*(0.74-2.*SCOF*RICHN)
   CC =-INTN*INTN*RICHN
*   SEE EQNS 4.37,4.38,4.39
   DZETA=(SQRT(BB*BB-4.*AA*CC)-BB)/(AA+AA)
*   SEE EQN 4.36
   GO TO 46

```

* EXTREME STABILITY, LAMINAR FLOW

```

45 DZETA =10
46 INTM =INTN+SCOF*DZETA
*   SEE EQN 4.33
   USTAR =KARMAN*WINDR/INTM
   MONOBL=(ZA1-ZNOT)/DZETA
   ABTURR=(0.74*INTN+Z1A*SCOF/MONOBL)/(KARMAN*USTAR)
*   SEE EQN 4.29
   WINDC =WINDR -USTAR*(INTN+Z1A*SCOF/MONOBL)/KARMAN
   GO TO 50

```

* UNSTABLE CONDITIONS WHEN RICHN IS NEGATIVE.

```

43 CONTINUE
   DZETA=IMPL(-0.001,1.E-3,DZETA1)
   MONOBL=Z1S/DZETA
   INTM,AM,BM,CM,DM,EM=NUMINT(1.,15.0,-0.25,ZS,MONOBL)
   INTN,AH,BH,CH,DH,EH=NUMINT(0.74, 9.0,-0.50,ZS,MONOBL)
   DZETA1=RICHN*INTM*INTM/INTN
*   SEE EQN 4.43
   USTAR =KARMAN*WINDR/INTM
   MONOBL=(ZA1-ZNOT)/DZETA
   KU =KARMAN*USTAR
   ABTURR=(AH*INTN+BH*Z1A+CH*Z2A+DH*Z3A+EH*Z4A)/KU
*   SEE EQN 4.29
   INTAM=AM*INTN+BM*Z1A+CM*Z2A+DM*Z3A+EM*Z4A
   WINDC =WINDR -USTAR*INTAM/KARMAN

```

50 CONTINUE

ENDPRO

* INSIDE VEGETATIVE CANOPY AERODYNAMICS, EDDY DIFFUSIVITY AND WINDSPEED
 PROCED PHI=INSIDE(ALPHAK,LMIX,TABS,WINDC)

```

   WIND(1) =WINDC
   GRLMIX =GR*LMIX*LMIX
   RICH(1)=-GRLMIX*DT(1)/(TABS*(0.5*DIK(1)+Z1A)*WIND(1)*WIND(1) )
*   SEE EQN 4.50
   I =1
   GO TO 121

```

* DO 123 I=2,NUMLL

```

120 WIND(I) =AMAX1(WINDC*EXP(-ALPHAK*(1.-HTOP(I)/CROFHT) ),0.0001)

```

* SEE EQN 4.48

```

   RICH(I)=GRLMIX*(DT(I-1)-DT(I))/...
           (TABS*(0.5*DIK(I-1)+DIK(I) )*WIND(I)**2)

```

* SEE EQN 4.50

```

121 IF (RICH(I).GT.0.) GO TO 122

```

```

   PHI =0.74/SQRT(1.-9.*RICH(I) )

```

```

   GO TO 123

```

```

122 PHI =0.74+(0.74-9.4*RICH(I)-0.74*SQRT(8.9262*RICH(I)+1.)) ...
      /(9.4*RICH(I)-2.)

```

* SEE EQN 4.51


```

      IF (RICH1(I).GT.0.21) PHI=1.E10
123    K(I)=LMIX*WIND(1)/0.74
*     SEE EQN 4.52 (ADDITION OF IW IS LATER DEVELOPMENT)
      I =I+1
      IF (I.LE.NUMLL) GO TO 120
ENDPRO

```

```

*     CALCULATION OF RESISTANCES BETWEEN LAYERS, REFHT AND SOIL SURFACE
*     FROM TURBULENCE AND WINDSPEED
PROCED RSSH,RSSE=TURBR(NUML1,ALPHA,K,LMIX,ABTURR,PHI)
RINC(1)=0.5*DIK(1)/K(1)+ABTURR
TCOND(1)=RHOC/ RINC(1)
VCOND(1)=TCOND(1)/PSCH
      DO 149 I=2,NUMLL
        RINC(I)=(DIK(I-1)+DIK(I))*0.5/K(I)
        TCOND(I)=RHOC/ RINC(I)
149    VCOND(I)=TCOND(I)/PSCH
      WIND(NUML1)=WIND(1)*EXP(-ALPHA)
*     SEE EQN 4.48
      K(NUML1)=WIND(NUML1)*LMIX/0.74
      RINC(NUML1)=0.5*DIK(NUML1)/K(NUML1)
      RSS=185.*SQRT(HRES/WIND(NUML1))
*     SEE EQN 3.5
      RSSH=RSS+RINC(NUML1)
      RSSE=0.93*RSS+RINC(NUML1)
ENDPRO

```

***** SECTION 10 *****

```

*     RADIATION AND ENERGY BALANCE OF THE LEAVES
*     RADIATION CONDITIONS INSIDE CANOPY AND ON SOIL SURFACE.
*     ENERGY BALANCE OF INDIVIDUAL LEAVES, ACCUMULATION OF HEAT LOSS
*     AND TRANSPIRATION
      PROCED TNC02A,TNC02R,NRADS,ETS=...
      SOURCE(KDR,SRW,SUNDCL,SNHSS,TA,RSSH)

```

```

177 CONTINUE
      IF (X.NE.0) GO TO 4051
      DO 4050 I=1,NUML1
4050  DELTA(I)=6.11*EXP(17.4*TA/TA239)*4158.6/(TA239**2)
4051  X=1

```

***** ITERATION START *****

```

4000  XD =EXP(-DL*KDR)
      XNDR =EXP(-DL*KDRN)
      XVDR =EXP(-DL*KDRV)
      VISDF=(1.-RFOVV)*DIFCL *(1.-XVDF)/DL
      NIRDF=(1.-RFOVN)*DIFCL *(1.-XNDF)/DL
*     SEE EQN 3.35
      FVDR =(1.-RDRV) *SUNDCL*(1.-XVDR)/DL
      FNDR =(1.-RDNR) *SUNDCL*(1.-XNDR)/DL
*     SEE EQN 3.36
      NDIR =(1.-SCN) *SUNDCL*(1.-XD) /DL
      VDIR =(1.-SCV) *SUNDCL*(1.-XD) /DL

```

```

* SEE EQN 3.39
VISDFC=VISDF+FVDR-VDIR
NIRDFC=NIRDF+FNDR-NDIR
* SEE EQN 3.40
VSPEP=(1.-SCV)*SUNDC/SHHS
NSPEP=(1.-SCH)*SUNDC/SHHS
FSR=(1.-XD)/(DL*KDR)
* SEE EQN 3.38
VISDFD=(1.-RFUVV)*DIFOV*(1.-XVDF)/DL
NIRDFD=(1.-RFVDN)*0.7*DIFOV*(1.-XNDF)/DL
* SEE EQN 3.35
* DARK RESPIRATION OF LEAVES
DPL=6.*EXP(0.07*(TA-30.))

```

```

XLFD=1.-XL
XLFDU=XLFD/XL
TNCO2A=0.
ETS4=0.
NLWRS=0.

```

***** CALCULATION OF FICTIVE LEAF RESISTANCE PER LAYER *****

```

DO 154 I=1,NUMLL
* EXCHANGE OF THERMAL RADIATION BETWEEN SKY, LEAVES AND SOIL SURFACE
* SEE EQNS 2.68 AND 2.69
DELTAM=DELTA(I)
NLWR(I)=0.
DO 162 J=1,NUMLL
IF (I-J) 167,162,164
167 NLWRM(I,J)=(0.0281*(TL(I)+TL(J))+4.61)*(TL(I)-TL(J))*...
(1.-XL)*(1.-XL)*XL**((J-I)-1)
GO TO 165
164 NLWRM(I,J)=-NLWRM(J,I)
165 NLWR(I)=NLWR(I)-NLWRM(I,J)
162 CONTINUE
LWSOIL=(0.0281*(TL(I)+TS)+4.61)*(TS-TL(I))*XLN*XLFDU
NLWR(I)=NLWR(I)+LWSOIL
NLWRS=NLWRS-LWSOIL
TL4=(TL(I)+273.)*4
LWR=(NLWR(I)+SIGMA*(SKT4-TL4)*XLFD)/DL
DEWL(I)=DEW(I)
RA(I)=185.*SQRT(2*WIDTH/(WIND(I)+WIND(I+1)))*0.5
RAM=RA(I)
* SEE EQN 3.5
RR(I)=RA(I)/RHOCF
REM=RR(I)
TADT(I)=TA+DT(I)
TADTM=TADT(I)
DRYP=(VPD+DELTA(I)*DT(I)-DV(I))/RR(I)
* SEE EQN 3.13
LHLLQ,SHLLQ,TLQ,NCQ2AQ,LRESQ,ABSQ=...
TRPH(VISDFD,NIRDFD,LWR)
IF (HOUR.EQ.12.) WRITE (6,600) LRESQ
600 FORMAT (11H LRESQ IS E10.5/)
IF (SHHS.EQ.0.) GO TO 230
NCQ2AC=0.
SHLLCL=0.
LHLLCL=0.
TLCL=0.

```

```

ABSCL=0
IF (LFCL.EQ.0.) GO TO 261
DO 260 SN=1,10
  SNINC=-0.05+0.1*SN
  VIS =VISDFC+VSPER*SNINC
  NIR =NIRDFC+NSPER*SNINC
  LHLLC,SHLLC,TL SUN,NPHTSU,LRESSU,ABSC= ...
  TRPH(VIS,NIR,LWR)
  IF (HOUR.EQ.12.) WRITE (6,601) LRESSU
601  FORMAT (13H LRESSU IS E10.5/)
  IF (SUMF.NE.0.) ZISSN=Z(IS,SN)
  SHLLCL=SHLLCL+ZISSN*SHLLC
  LHLLCL=LHLLCL+ZISSN*LHLLC
  NCO2AC=NCO2AC+ZISSN*NPHTSU
*   SEE EQN 3.42
  TLCL =TLCL +ZISSN*TL SUN
260  ABSCL=ABSCL+ZISSN*ABSC
  SHLLCL=SHLLCL*FSR
  ABSCL=ABSCL*FSR
  LHLLCL=LHLLCL*FSR
  TLCL =TLCL *FSR
  NCO2AC=NCO2AC*FSR
  LHLLS,SHLLS,TL SHAD,NPHTSH,LRESSH,ABSS= ...
  TRPH(VISDFC,NIRDFC,LWR)
  IF (HOUR.EQ.12.) WRITE (6,602) LRESSH
602  FORMAT (13H LRESSH IS E10.5/)
  SHLLCL=SHLLCL+SHLLS *(1.-FSR)
  LHLLCL=LHLLCL+LHLLS *(1.-FSR)
  TLCL =TLCL +TL SHAD*(1.-FSR)
  NCO2AC=NCO2AC+NPHTSH*(1.-FSR)
  ABSCL=ABSCL+ABSS*(1.-FSR)
261  ABSRAD(I)=(LFOV*ABSO+LFCL*ABSCL)*DL
  SHLL(I)=(LFOV*SHLLO +LFCL*SHLLCL)*DL
  LHLL(I)=(LFOV*LHLLLO +LFCL*LHLLCL)*DL
  NCO2A(I)=(LFOV*NCO2AO+LFCL*NCO2AC)*DL
*   SEE EQN 3.43
  TL(I) = LFOV*TLO +LFCL*TLCL
  FSR =FSR *XD
  VISDFC=VISDFC*XVDF
  NIRDFC=NIRDFC*XNDF
  VISDF =VISDF *XVDF
  NIRDF =NIRDF *XNDF
  FVDR =FVDR *XVDR
  FNDR =FNDR *XNDR
  VDIR =VDIR *XD
  NDIR =NDIR *XD
  VISDFC=VISDF+FVDR-VDIR
  NIRDFC=NIRDF+FNDR-NDIR
*   SEE EQN 3.40
  GO TO 231
230  SHLL(I)=SHLLO *DL
  ABSRAD(I)=ABSO*DL
  LHLL(I)=LHLLLO *DL
  NCO2A(I)=NCO2AO*DL
  TL(I) =TLO
231  ETS4 =ETS4+XLFD*(TL(I)+273)**4
  XLFD =XLFD*XL
  XLFU =XLFU/XL
  TNCO2A=TNCO2A+NCO2A(I)
  TM(I)=(TL(I)+TA)/2

```

```
DELTA(I)=6.11*EXP(17.4*TM(I)/(TM(I)+239))*4158.6/ ...
      ((TM(I)+239)**2)
```

```
TSM=(TS+TA)/2
```

```
DELTA(NUML1)=6.11*EXP(17.4*TSM/(TSM+239))*4158.6/ ...
      ((TSM+239)**2)
```

```
EPS(I)=SVPA+DELTA(I)*(TL(I)-TA)
```

```
LBR(I)=(EPS(I)-VPA-DV(I))*RHOC/PSCN*DL/LHLL(I)
```

```
IF (LHLL(I).LE.0.AND.DEW(I).GT.0) GO TO 153
```

```
IF (LBR(I).GE.(RA(I)*0.93)) GO TO 154
```

```
153 LBR(I)=RA(I)*0.93
```

```
154 CONTINUE
```

```
RA(NUML1)=RSS
```

```
EPS(NUML1)=SVPA+DELTA(NUML1)*(TS-TA)
```

```
LBR(NUML1)=RA(NUML1)*0.93
```

```
ETS4 =ETS4+XLN*(TS+273)**4
```

```
ETS =ABS(ETS4)**0.25 - 273.
```

```
TNCO2R=TNCO2A/3600.
```

```
FVDR =FVDR *DL/(1.-XVDR)
```

```
FNDR =FNDR *DL/(1.-XNDR)
```

```
VISDFO=VISDFO*DL/(1.-XVDF)
```

```
NIRDFO=NIRDFO*DL/(1.-XNDF)
```

```
TS4 =(TS+273.)*4
```

```
NLWRS =NLWRS+SIGMA*XLN*(SKT4-TS4)
```

```
VISDF =VISDF *DL/(1.-XVDF)
```

```
NIRDF =NIRDF *DL/(1.-XNDF)
```

```
NRADS =LFOV *(VISDFO+NIRDFO) *NLWRS + ...
```

```
LFCL *(VISDF +NIRDF +FVDR+FNDR)
```

```
ABSNR =NRAD -NRADS
```

```
ABSRAD(NUML1)= NRADS
```

```
***** CALCULATION OF "MAT" AND "VEC" *****
```

```
DO 5100 I=1,NUML12
```

```
N=I-1
```

```
N=I+1
```

```
DO 5110 J=1,NUML12
```

```
IF (J.EQ.(I-1).AND.I.LT.NUML2) MAT(I,J)= ...
      -DELTA(I)*RA(I)/(RINC(I)*DL)
```

```
IF (J.EQ.I.AND.I.LT.NUML1) MAT(I,J)= ...
      DELTA(I)*RA(I)/(RINC(I)*DL)+DELTA(I)*RA(I)/ ...
      (RINC(N)*DL)+DELTA(I)
```

```
IF (I.EQ.NUML1.AND.J.EQ.I) MAT(I,J)= ...
      DELTA(I)*RA(I)/(RINC(I)*DS)+DELTA(I)
```

```
IF (J.EQ.(I+1).AND.I.LT.NUML1) MAT(I,J)= ...
      -DELTA(I)*RA(I)/(RINC(N)*DL)
```

```
IF (J.EQ.(I+NUMLL).AND.I.LT.NUML2) MAT(I,J)= ...
      LBR(I)/(RINC(I)*DL)
```

```
IF (J.EQ.(I+NUML1).AND.I.LT.NUML1) MAT(I,J)= ...
      -(LBR(I)/(RINC(N)*DL)+LBR(I)/(RINC(I)*DL)+1)
```

```
IF (J.EQ.(I+NUML2).AND.I.LT.NUML1) MAT(I,J)= ...
      LBR(I)/(RINC(N)*DL)
```

```
IF (I.GT.NUML1.AND.J.EQ.(I-1)) MAT(I,J)= ...
      -RHOC/PSCN*(RINC(I-1))
```

```
IF (J.GT.(I-1).AND.J.LT.(I-1)) MAT(I,J)=0
```

```
IF (J.GT.(I+1).AND.J.LT.(I+NUMLL)) MAT(I,J)=0
```

```
IF (J.GT.(I+NUML2)) MAT(I,J)=0
```

```
IF (J.LT.(I-1)) MAT(I,J)=0
```

```
IF (I.EQ.NUML12) GO TO 5110
```

```
IF (I.GT.NUML1.AND.J.EQ.(I-1)) MAT(I,J)= ...
```

```

      =RHOCF/RINC(I=NUML1)
IF (I.GT.NUML1.AND.J.EQ.(I=NUML1)) MAT(I,J)= ...
      RHOCF*(1/RINC(J)+1/RINC(J+1))
IF (I.GT.NUML1.AND.J.EQ.(I=NUMLL)) MAT(I,J)= ...
      =RHOCF/RINC(J)
IF (I.GT.NUML1.AND.J.EQ.I) MAT(I,J)= ...
      RHOCF/PSCH*(1/RINC(I=NUML1)+1/RINC(I=NUMLL))
IF (I.GT.NUML1.AND.J.EQ.(I+1)) MAT(I,J)= ...
      =RHOCF/(RINC(I=NUMLL)*PSCH)
5110  CONTINUE
      IF (I.LT.NUML2) VEC(I)=TA*DELTA(I)-SVPA
      IF (I.GT.NUML1) VEC(I)=ABSRAD(I=NUML1)-0.3*NC02A(I=NUML1)
5100  CONTINUE
      MAT(NUML1,1)=0
      MAT(NUML1,NUML2)=0
      MAT(NUML2,NUML1)=0
      MAT(NUML2,NUML12)=0
      MAT(1,NUML1)=0
      MAT(NUML12,NUML2)=0
      MAT(NUML1,NUMLL)=-DELTA(NUML1)*RA(NUML1)/(RINC(NUML1)*DS)
      MAT(NUML1,(NUML12=1))=LBR(NUML1)/(RINC(NUML1)*DS)
      MAT(NUML1,NUML12)=-MAT(NUML1,(NUML12=1))-1
      MAT(NUML12,NUMLL)=-RHOCF/RINC(NUML1)-RA(NUML1)/RINC(NUML1)* ...
      2*LAMBDA/TCOM1
      MAT(NUML12,NUML1)=RHOCF/RINC(NUML1)+(1+RA(NUML1)/RINC(NUML1))* ...
      2*LAMBDA/TCOM1
      MAT(NUML12,NUML12)=RHOCF/(RINC(NUML1)*PSCH)
      VEC(1)=TA*(DELTA(1)+DELTA(1)*RA(1)/(RINC(1)*DL))-SVPA=LBR(1) ...
      *VPA/(RINC(1)*DL)
      VEC(NUML2)=ABSRAD(1)-0.3*NC02A(1)+TA*RHOCF/RINC(1)+ ...
      RHOCF*VPA/(RSCH*RINC(1))
      VEC(NUML12)=ABSRAD(NUML1)+TEMP1*2*LAMBDA/TCOM1

```

***** SOLVATION OF THE MATRIX EQUATION *****

```
CALL LEQTF (MAT,1,NUML12,20,VEC,4,WKAREA,IER)
```

***** COMPUTATIONS FROM THE MATRIX SOLUTION *****

```

DO 5120 I=1,NUML1
  TADT(I)=VEC(I)
  VPAC(I)=VEC(I+NUML1)
5120 CONTINUE
DO 5130 I=1,NUML1
  DT(I)=TADT(I)-TA
  DV(I)=VPAC(I)-VPA
5130 CONTINUE
DO 5140 I=1,NUMLL
  M=NUML2-I
  SHLL(M)=(TADT(M)-TADT(M-1))*RHOCF/RINC(M)-SHFL(M+1)
  LHLL(M)=(VPAC(M)-VPAC(M-1))*RHOCF/(RINC(M)*PSCH)-LHFL(M+1)
  SHFL(M)=SHFL(M+1)+SHLL(M)
  LHFL(M)=LHFL(M+1)+LHLL(M)
5140 CONTINUE
  SHLL(1)=(TADT(1)-TA)*RHOCF/RINC(1)-SHFL(2)
  LHLL(1)=(VPAC(1)-VPA)*RHOCF/(RINC(1)*PSCH)-LHFL(2)
  SHFL(1)=SHFL(2)+SHLL(1)
  LHFL(1)=LHFL(2)+LHLL(1)

```

```

DO 5150 I=1,NUMLL
  LTL(I)=TL(I)
  TL(I)=RA(I)*SHLL(I)/(DL*RHOC)+TADT(I)
  EPS(I)=LBR(I)*PSCH*LHLL(I)/(DL*RHOC)+VPAC(I)
5150  CONTINUE
  LTS=TS
  TS=RA(NUML1)*SHLL(NUML1)/(DS*RHOC)+TADT(NUML1)
  EPS(NUML1)=LBR(NUML1)*PSCH*LHLL(NUML1)/(DS*RHOC)+VPAC(NUML1)
  SHLL(NUML2)=(TEMP1-TS)*2*LAMBDA/TCOM1
  X=X+1
  IF (X.EQ.15) TIME=86400
  IF (ABS((LTS-TS)/TS).GE.0.01) GO TO 4000
DO 5170 I=1,NUMLL
  IF (ABS((LTL(I)-TL(I))/TL(I)).GE.0.01) GO TO 4000
5170  CONTINUE

```

***** END OF ITERATION PROCESS *****

ENDPRO

***** SECTION 11 *****

* ENERGY BALANCE OF THE SOIL

* SOIL TEMPERATURES

PROCD TEMP1=STEMP(NRADS)

178 CONTINUE

DO 146 I=1,10

146 TEMP(I)=HC(I)/(TCOM(I)*VHCAP)

TEMP1=TEMP(1)

ENDPRO

* ENERGY BALANCE OF SOIL SURFACE (SECTION 3.3)

PROCD VPDB,LHFLB,SHFLB,TS,G=...

SOILS(NUML1,TA,SVPA,DELTAM,VRD,NRADS,RSSH,RSSE,DIST1,TEMP1)

HFL(1)=SHLL(NUML2)

G=-SHLL(NUML2)

DO 147 I=2,10

147 HFL(I)=(TEMP(I)-TEMP(I-1))*LAMBDA/DIST(I)

DO 148 I=1,10

148 NHFL(I)=HFL(I+1)-HFL(I)

ENDPRO

***** SECTION 12 *****

* ENERGY AND MASS BALANCE OF THE AIR LAYERS IN THE CANOPY

* FLUXES BETWEEN LAYERS

* RATES OF THE INTEGRALS THAT DESCRIBE THE STATE OF THE LAYERS

```

PROCED TELFO=RATE(G)
  DO 158 I=1,NUMLL
    IF (DEW(I).GT.0.) GO TO 157
    DDEWT(I)=0.
    DDEW(I) =AMAX1(0.,-LHLL(I) )
    GO TO 158
  157  DDEWT(I)=1.
    DDEW(I) =-LHLL(I)
  158  CONTINUE
  TELFO=TELF0+1
ENDPRO

```

```

PROCED INVL=PROCO2(NUML1,PRT,TELF0)
  CO2FL(NUML1)=SRESP
  DO 210 I=1,NUMLL
    INVL=NUML1-I
  210  CO2FL(INVL)=CO2FL(INVL+1)-NCO2A(INVL)
    CO2(I)=ECO2C+(SRESP-TNCO2A)*RINC(I)/68.4
    IF (NUMLL.EQ.1) GO TO 212
    DO 211 I=2,NUMLL
  211  CO2(I)=CO2(I-1)+CO2FL(I)*RINC(I)/68.4
  212 CONTINUE
ENDPRO

```

NOSORT

***** SECTION 13 *****

* INTEGRALS AND OUTPUT

** LAST ARGUMENT OF THESE STATEMENTS MUST EQUAL THE NUMBER OF LAYERS

DEW=INTGRL(IDEW,DDEW,3)

DEWT=INTGRL(IDEWT,DDEWT,3)

**

HC=INTGRL(HCI,NHFL,10)

DNCO2A=INTGRL(0.,TNCO2R)

DLHFL=INTGRL(0.,LHFL(1))

DSHFL=INTGRL(0.,SHFL(1))

DNRAD=INTGRL(0.,NRAD)

DLHFLB=INTGRL(0.,LHFLB)

DSHFLB=INTGRL(0.,SHFLB)

DSOILF=INTGRL(0.,G)

IF (PRT.LT.0.5) GO TO 851

WRITE (6,900) TIME, HOUR

WRITE (6,901) (DT(I),I=1,NUMLL)

WRITE (6,903) (DV(I),I=1,NUMLL)

WRITE (6,902) (NCO2A(I),I=1,NUMLL)

WRITE (6,904) (SHLL(I),I=1,NUMLL)

WRITE (6,905) (LHLL(I),I=1,NUMLL)

WRITE (6,907) (DEW(I),I=1,NUMLL)

WRITE (6,908) (DEWT(I),I=1,NUMLL)

WRITE (6,909) (TL(I),I=1,NUMLL)

WRITE (6,1003) (SHFL(I),I=1,NUML1)

WRITE (6,1004) (LHFL(I),I=1,NUML1)

```

WRITE (6,1007) ECU2C,(CU2(I),I=1,NUMLL)
WRITE (6,1008) ( (WIND(I),K(I) ),I=1,NUMLL)
SIX=IMPULS(0.,21600.)
IF (SIX.LT.0.5) GO TO 851
WRITE (6,1005) (TEMP(I),I=1,10)
TRCOEF=0.006*DLHFL/(NOT(DNCO2A)+DNCO2A)
DBOWR =DSHFL/(NOT(DLHFL)+DLHFL)
WRITE (6,1006) DNCO2A,DLHFL,DSHFL,DNRAD,DLHFLB,DSHFLB,DSOILF,...
TRCOEF,DBOWR
1006 FORMAT (61H DNCO2A,DLHFL,DSHFL,DNRAD,DLHFLB,DSHFLB,DSOILF,TRCOEF,
      8DBOWR /9E12,3//)
1005 FORMAT (12H TEMP(I=10) 10F11,3//)
900 FORMAT (21H TIME AND HOUR 2F15,2//)
901 FORMAT (11H DT 20F15,3//)
902 FORMAT (11H NCO2A 20E15,5//)
903 FORMAT (11H DV 20F15,3//)
904 FORMAT (11H SHLL 20E15,5//)
905 FORMAT (11H LHLL 20E15,5//)
907 FORMAT (11H DEW 20E15,5//)
908 FORMAT (11H DEWT 20E15,5//)
909 FORMAT (11H TL 20F15,3//)
1001 FORMAT (11H DDTDT 20E15,5//)
1002 FORMAT (11H DDVDT 20E15,5//)
1003 FORMAT (11H SHFL 21E15,5//)
1004 FORMAT (11H LHFL 21E15,5//)
1007 FORMAT (11H CO2 21F15,1//)
1008 FORMAT (25H WIND K (INSIDE)/20(2E12,3//) )
851 CONTINUE
CALL DEBUG(10,0.)
CALL DEBUG(10,3600.)
CALL DEBUG(1,86400.)
PRINT TS,G,INTN,INTN,SHFL(1),LHFL(1),NRAD,NRADS,ENLOSS,...
TELF0,TELN,ABTERR,RWCP,SRW,USTAR,MONOBL,DZETA,TA,...
RSSE,VPD,TNCO2A,SRAD,FOV,RICHN,...
SKT,LWRI,ETS
1000 CONTINUE
END
STOP
ENDJOB

```


W. Bouter

F

		Q_2	$\frac{2 * \text{Cond} * \gamma}{d * \rho * C_p * \Delta t} (Q_1 + Q_2)$
			$\frac{-\gamma}{R_{4\Delta 4}} (Q_1 + Q_2 + Q_3)$ $+\frac{\gamma}{R_{4\Delta 3}} (Q_1 + Q_2 + Q_3 + Q_4)$ +
$R_{4\Delta 3}$	$-\frac{\gamma}{R_{3\Delta 3}} Q_1$ $+\frac{\gamma}{R_{3\Delta 2}} Q_1$	$-\frac{\gamma}{R_{3\Delta 3}} (Q_1 + Q_2)$ $+\frac{\gamma}{R_{3\Delta 2}} (Q_1 + Q_2 + Q_3)$ $+\frac{r_2}{R_3}$	$-\frac{\gamma}{R_{3\Delta 3}} (Q_1 + Q_2 + Q_3 + Q_4)$ $+\frac{\gamma}{R_{3\Delta 2}} (Q_1 + Q_2)$ $-\frac{r_3}{R_3}$
	$-\frac{\gamma}{R_{2\Delta 2}} Q_1$ $+\frac{\gamma}{R_{2\Delta 1}} (Q_1 + Q_2) + \frac{r_1}{R_2}$	$-\frac{\gamma}{R_{2\Delta 2}} (Q_1 + Q_2 + Q_3)$ $+\frac{\gamma}{R_{2\Delta 1}} (Q_1)$ $-\frac{r_2}{R_2} - 1$	$-\frac{\gamma}{R_{2\Delta 2}} (Q_1 + Q_2)$ $+\frac{\gamma}{R_{2\Delta 1}} Q_1$ -1
	$-\frac{\gamma}{R_{1\Delta 1}} (Q_1 + Q_2)$ $-\frac{r_1}{R_1} - 1$	$-\frac{\gamma}{R_{1\Delta 1}} (Q_1)$ -1	$-\frac{\gamma}{R_{1\Delta 1}} Q_1$ -1

Figuur 2b Matrix B. De open plaatsen bevatten de zelfde e
