Ontwikkeling van een simulatiemodel voor transpiratie en wateropname en van een integraal gewasmodel

Verslag van een onderzoek gedaan in opdracht van en gedeeltelijk gefinancierd door het NOVEM

H. Gijzen

ab-dlo

Contract 35116/0070
NOVEM Programma Agrarische Sector

Rapport 18, Wageningen
december 1994

Het DLO-Instituut voor Agrobiologisch en Bodemvruchtbaarheidsonderzoek (AB-DLO) is onderdeel van de Dienst Landbouwkundig Onderzoek (DLO) van het Ministerie van Landbouw, Natuurbeheer en Visserij.
Het instituut is opgericht op 1 november 1993 en is ontstaan door de samenvoeging van het Wageningse Centrum voor Agrobiologisch Onderzoek (CABO-DLO) en het in Haren gevestigde Instituut voor Bodemvruchtbaarheid (IB-DLO).

DLO heeft tot taak het genereren van kennis en het ontwikkelen van expertise ten behoeve van de beleidsvoorbereiding en -uitvoering van het Ministerie van Landbouw, Natuurbeheer en Visserij, het bevorderen van de primaire landbouw en de agrarische industrie, het inrichten en beheren van het landelijk gebied, en het beschermen van natuur en milieu.

AB-DLO heeft tot taak het verrichten van zowel fundamenteel-strategisch als toepassingsgericht onderzoek en is gepositioneerd tussen het fundamentele basisonderzoek van de universiteiten en het praktijkgerichte onderzoek op proefstations. De verkregen onderzoeksresultaten dragen bij aan de bevordering van:

- de bodemkwaliteit;
- duurzame plantaardige produktiesystemen;
- de kwaliteit van landbouwprodukten.

Kernexpertises van het AB-DLO zijn: plantenfysiologie, bodembiologie, bodemchemie en -fysica, nutriëntenbeheer, gewas- en onkruidecologie, graslandkunde en agrosysteemkunde.

Adres

Vestiging Wageningen:
Postbus 14, 6700 AA Wageningen
tel. 08370-75700
fax 08370-23110
e-mail postkamer@ab.agro.nl

Vestiging Haren:
Postbus 129, 9750 AC Haren
tel. 050-337777
fax 050-337291
e-mail postkamer@ab.agro.nl

Dankbetuiging

Hierbij dank ik Dr. S.C. van de Geijn, hoofd van de afdeling Plantenfysiologie, voor zijn grote steun bij de werkzaamheden aan dit project, en de overige leden van de begeleidingscomissie: Prof. dr. ir. R. Rabbinge (LUW-Vakgroep Theoretische Productie-Ecologie), Prof. dr. ir. H. Challa (LUW-vakgroep Tuinbouwplantenteelt), Ir. C.H.M.G. Custers (NOVEM) en Dr. ir. J.C. Bakker (PTG, Naaldwijk) voor hun begeleiding en inzet. Ik ben Dr. C. Stanghellini (IMAG-DLO) en Ir. R. de Graaf (PTG, Naaldwijk) zeer erkentelijk voor het beschikbaar stellen van verdampingsgevens.

Inhoudsopgave

Inleiding

De transpiratie en wateropname van kasgewassen hebben een grote invloed op de gewasgroei, de produktie en de produktkwaliteit. Daarom wordt door de tuinder veel aandacht gegeven aan de beïnvloeding van het kasklimaat, om door sturing van transpiratie en wateropname de groei en kwaliteit te optimaliseren. Veelal is dit een sterk door persoonlijke deskundigheid en ervaring bepaalde activiteit, waarbij moeilijk te objectiveren waarnemingen aan het gewas de richting en grootte van ingrepen bepalen. Met het gebruik van extra verwarming en ventilatie (vaak een combinatie van stoken en ventileren), wordt het gewenste effect nagestreefd. Dit leidt tot een ongewenste verhoging van het energiegebruik, waarvan de doelmatigheid bovendien niet altijd duidelijk is. Op grond van het belang dat gehecht wordt aan een voldoende nauwkeurige kwantitatieve beschrijving van wateropname en verdamping, is het hier beschreven project gestart.

Het hoofddoel van het project is geweest een of meerdere bruikbare modellen voor transpiratie en wateropname te ontwikkelen, te kalibreren en te testen. Daarbij zouden de modellen gekoppeld moeten worden aan een model voor fotosynthese en drogestofproduktie. Het integrale model en de onderdelen zijn zo ontworpen dat deze in principe geschikt zijn voor toepassing in een later/elders te ontwikkelen verbeterde kasklimaatregeling. Met name de effectieve regeling van de gewenste gewasverdamping (via stoken en ventileren) is van groot belang voor de beperking van de energiekosten van kasteelten. Hoewel het eindprodukt primair van belang is voor de doelgroep in onderzoek en bedrijfsleven die over deskundigheid op het gebied van toepassing van modellen beschikt, kan een vereenvoudigd model ook gebruikt worden voor elementaire verkenningen van diverse kasklimaat/gewassituaties, en daarmee van belang zijn voor voorlichting en IKC.

In het hier gerapporteerde onderzoek is uit literatuur en experimentele gegevens informatie verzameld over verschillende componenten van kortgolvige straling buiten de kas, en over samenstelling en energiekosten van de biosynthese van plantedelen. Stralingstransmissie van het kasdek, het stralingsklimaat en overig kasklimaat bepalen samen met gewaseigenschappen de gewasverdamping en gewasfotosynthese. Uit fotosyntheseprodukten wordt, onder aftrek van ademhalingskosten en kosten van biosynthese, drogestof gevormd. In het rapport worden een aantal alternatieve manieren voor het beschrijven van gewasverdamping uitgewerkt en vergeleken. Aangegeven wordt wat de mogelijkheden en beperkingen zijn, en welke factoren, zoals gewasstructuur en huidmondjesgeleidbaarheid, van belang zijn voor een adequate beschrijving. In de conclusies wordt aangegeven waar de toepasbaarheid ligt van verschillende modules.

Samenvatting

Stralingsklimaat in de kas en het effect van de gewasstructuur

De fractie fotosynthetisch actieve straling (PAR) in globale straling is een belangrijke parameter in gewasgroeimodellen omdat groei vrijwel evenredig is met onderschepte PAR. Tot dusverre werd deze fractie als constant beschouwd hoewel het kan variëren. Een regressievergelijking is opgesteld voor de schatting van de fractie PAR in globale straling op basis van gemeten globale straling. De fractie PAR blijkt bij helder weer rond de 45 % te liggen, en toe te nemen tot ongeveer 50 % bij zwaar bewolkt weer. Er is een klein effect van de zonshoogte. Ook is een model opgesteld, aan de hand van literatuurgegevens, om op basis van gemeten globale straling de fractie diffuus in PAR te schatten en tevens de grootte van de fluxen diffuse en directe Nabij InfraRode straling (NIR) straling . Deze laatste stralingscomponenten zijn van belang voor de warmtebelans van gewas en kas. Op basis van het model kunnen deze nu beter geschat worden.

Er is een begin gemaakt met de ontwikkeling van het model voor de absorptie en verdeling van Nabij InfraRode straling in een rijgewas. Vanwege de relatief kleinere relevantie vergeleken met andere te modelleren aspecten en vanwege tijdgebrek is het niet afgerond. In de paragraaf volgend op de conclusies is een korte beschouwing gewijd aan het belang van de absorptie van NIR op gewasverdamping.

Testen van het drogestofproduktiemodel

Optimalisatie van de produktie (ook in economische zin) betekent een zo goed mogelijk afstemming van inputfactoren voor het bereiken van de gewenste gewasgroei, produktie en kwaliteit. Drogestofproduktie legt hiervoor de basis, en vereist daarom een nauwkeurige schatting en afweging met de inputs. Bij de vorming van drogestof worden voor de biosynthese van bijvoorbeeld celwandmateriaal en eiwitten energie (suikers) verbruikt. De samenstelling van het gevormde materiaal bepaalt voor een belangrijk deel de kosten van de biosynthese en onderhoud van de weefsels. Daarom zijn chemische analyses zijn uitgevoerd van plantmateriaal van komkommer, paprika, tomaat en aubergine. Op basis hiervan zijn nauwkeuriger schattingen verkregen van de assimilatenbehoefte voor de vorming van drogestof dan tot dusver bestonden. Gesimuleerde drogestofprodukties zijn vergeleken met gemeten produktie voor komkommer, paprika en tomaat. Ondanks enige overschatting werd een bevredigend resultaat verkregen. Afwijkingen houden, naar verwachting, verband met het feit dat een aantal factoren, waarmee geen rekening werd gehouden, mogelijk een rol hebben gespeeld.

Toetsing van het transpiratie- en wateropnamemodel

Bij metingen die voor validatie en calibratie van verdampingsmodellen worden gedaan kan niet altijd worden voorkomen dat beschaduwing een rol speelt. In een opstelling met lysimeters is beschaduwing door kasconstructiedelen en door aanliggende rijen er vaak de oorzaak van dat de meting niet representatief is voor het gewas als geheel. Dit speelt uiteraard ook een rol als de resultaten van een meting met lysimeters voor regel-doeleinden wordt gebruikt. In het project is een model is ontwikkeld om beschaduwingseffecten van kas-

constructiedelen (2- en 3-dimensionaal) en buur-rijen op een gewasrij, plantrij of groepje planten uit te rekenen.
Een aantal submodellen voor huidmondjesgeleidbaarheid zijn getest op hun vermogen om, als onderdeel van een gewasmodel, gemeten gewastranspiratie te benaderen en te verklaren. Twee huidmondjesmodellen zijn gebaseerd op bladfotosynthese, en twee andere zijn beschrijvende modellen. Bij tuning van de parameters van deze 4 modellen met metingen aan gewasverdamping van tomaat, paprika en komkommer werd een goede fit verkregen. De op fotosynthese gebaseerde huidmondjesmodellen bleken bij paprika vaak te hoge stomataire geleidbaarheden in de top van het gewas te voorspellen.
Een belangrijke verbetering t.o.v. het bestaande gewasverdampingsmodel (zoals die o.a. in het ECP-model wordt gebruikt) werd verkregen door de introductie van het effect van de luchtvochtigheid op de huidmondjesgeleidbaarheid. Listings van alle huidmondjesmodellen zijn in appendices opgenomen. In het interactieve programma is een beschrijvende model voor huidmondjesopening opgenomen.

Gevoeligheidsanalyses met transpiratiemodel

De ontwikkelde modellen hebben het mogelijk gemaakt te onderzoeken welke factoren belangrijk zijn aangaande de gewasverdamping. Op grond hiervan wordt geconcludeerd dat de belangrijkste factoren die invloed hebben zijn: de bladindex, de intensiteit van globale straling en de luchtvochtigheid. Andere belangrijke factoren zijn de responsen van huidmondjesgeleidbaarheid op licht en luchtvochtigheid, en de temperaturen van kasdek en grondoppervlak. Minder belangrijke factoren zijn bladhoekverdeling en gewasgeometrie.

Koppeling verdampingsmodel met drogestofproduktiemodel

Deze koppeling is tot stand gebracht via de introductie van huidmondjesmodellen op basis van fotosynthese. Met deze modellen kon de gemeten gewasverdamping goed benaderd worden. Ook voor de beschrijvende huidmondjesmodellen is een koppeling met de berekening van gewasfotosynthese gedaan. Echter, in dit geval is de koppeling minder strikt, omdat de huidmondjesopening wel de bladfotosynthese beïnvloedt, maar de bladfotosynthese geen effect heeft op de huidmondjesopening. Dit laatste is in werkelijkheid wel het geval. Deze koppeling zal dus een minder betrouwbare inschatting geven van het effect van huidmondjesopening op fotosynthese.

Ontwikkeling vereenvoudigde modellen

Een bladfotosynthesemodel wat op dit moment in een aantal van de huidige drogestofproduktie-modellen is ingebouwd, is verbeterd wat betreft de responsen op CO_2 en temperatuur. Het is nu tevens beter te parameteriseren. Hiermee bestaat nu een goed alternatief voor het meer ingewikkelde biochemische bladfotosynthesemodel van Farquhar et al. Van het bestaande 'multilayer'-verdampingsmodel is een vereenvoudigd 'big-leaf'-model gemaakt. Listings van deze modellen zijn in de appendix opgenomen.

Computerprogramma's

De programma's van de ontwikkelde modellen en submodellen zijn gedocumenteerd en worden op floppy-disks bij het rapport bijgeleverd. Programma's zijn modulair opgebouwd om modelonderhoud, en -verandering makkelijk te houden. Bij de ontwikkeling van het hoofdprogramma is ernaar gestreefd om data in- en uitvoer gebruikersvriendelijk te houden.

Conclusies

Het stralingsklimaat in de kas kan nu beter berekend worden op basis van een nauwkeuriger inschatting van de componenten fotosynthetisch actieve straling en nabij infrarode straling in de totale globale straling.
In bestaande modellen voor drogestofproduktie kunnen nauwkeuriger waarden voor de parameters voor de assimilatenbehoefte voor drogestofproduktie van de gewassen tomaat, paprika, komkommer en aubergine gebruikt worden.
Gewastranspiratie kon goed geschat worden, bij gebruikmaking van verschillende modules voor de huidmondjesrespons. Voor berekeningen aan gewastranspiratie en wateropname kan het beste gebruikt worden gemaakt van de beschrijvende module van de huidmondjesrespons met negatief-exponentiële respons op licht. Parameters van deze module kunnen worden vergeleken met, of geschat worden uit literatuurgegevens. De eenvoud van deze module maakt dat het totale gewastranspiratiemodel makkelijk ingebouwd kan worden in modellen op hogere integratieniveau's en dat het totale gewastranspiratiemodel weinig rekentijd behoeft.

Een aantal verschillen in parameterwaarden voor de verschillende gewassen zijn gevonden. Het is nog weinig duidelijk hoezeer de parameters van deze module afhankelijk zijn van gewas danwel klimaatsomstandigheden. De beschikbare datasets (van PTG-Naaldwijk en AB-DLO) vertegenwoordigen slechts een deel van het groeiseizoen, en niet voor alle gewassen dezelfde periode. Voorzichtigheid is daarom geboden bij toepassing van het model voor andere gewasomstandigheden dan welke in de huidige experimenten geheerst hebben. De module zal bij inbouw in het ECP-model naar verwachting leiden tot een betere berekening van de vochtbalans in de kas, met name vanwege de terugkoppeling met luchtvochtigheid die door dit huidmondjesmodel wordt beschreven.

De modules voor huidmondjesrespons, welke bladverdamping en bladfotosynthese koppelen, kunnen gebruikt worden in meer gedetailleerde en verklarende modellen, bijvoorbeeld om experimenten te analyseren. Op grond van de huidige gegevens is het niet duidelijk hoe groot hun voorspellende waarde is wat betreft de mate waarin verdamping de fotosynthese kan beïnvloeden. Meer experimentele gegevens zijn nodig omtrent deze interactie.

De doelstelling van het project een model voor de verdamping en wateropname van kasgewassen te ontwikkelen, te kalibreren en te testen, en hiervan afgeleide vereenvoudigde modellen beschikbaar te krijgen is bereikt. Ook de integratie in een groter model, waarin ook fotosynthese en gewasproduktie worden beschreven is gerealiseerd. Hoe groot de energiebesparing is die bij inbouw van deze modellen in verbeterde klasklimaatregelaars kan worden behaald is hier niet onderzocht.

Een belangrijke bron van onzekerheid blijft de kwaliteit van de parameterisatie voor verschillende gewassen en voor het hele seizoen. De beperkte beschikbaarheid van datasets is hieraan debet.

Het effect van de rij-structuur op absorptie van NIR

Veel tuinbouwgewassen hebben een duidelijke rij-structuur gedurende een kortere of langere periode in het groeiseizoen. Deze rij-structuur beïnvloedt de absorptie van straling, en daarmee invloed op processen als gewasfotosynthese en verdamping. Door Gijzen & Goudriaan (1989) is een model ontwikkeld om het effect van de rij-structuur op de absorptie van fotosynthetisch actieve straling (PAR, 400-700 nm) te berekenen. In dit model is ervan uitgegaan dat er geen interactie optreedt tussen rijen onderling wat betreft het weer uitzenden van geabsorbeerde straling naar een buurrij ('multiple scattering'). Deze aanname kon worden gedaan omdat PAR relatief weinig verstrooid in het gewas (ongeveer 85 % van de PAR dat op een enkel blad valt wordt geabsorbeerd, en 15 % wordt weer uitgezonden).

Voor Nabij-InfraRode straling kan deze aanname niet worden gedaan omdat het gewas voor deze straling veel transparanter is en hier de verstrooiing veel sterker is (ongeveer 20 % van de NIR die op een enkel blad valt wordt absorbeert). Daardoor 'ziet' een blad in een rij ook de NIR die door een blad in een buurrij wordt verstrooid.

Voor aanvang van het onderzoek was gepland om het rij-effect te kwantificeren voor het NIR. Na de ontwikkeling van enige basis-onderdelen van het model is van verdere model-ontwikkeling afgezien. Dit vanwege tijdgebrek en omdat de relevantie van het rij-effect voor de berekening van de verdamping relatief kleiner was dan van andere, ook minder goed beschreven processen als b.v. de huidmondjesrespons. Hieronder wordt kort beargumenteerd waarom het rij-effect voor NIR-absorptie niet zo groot is.

Voor een gemiddeld gewas met bladindex gelijk aan 3, wordt bijna 40 % van de inkomende NIR gereflecteerd, en 43 % geabsorbeerd. Inkomende PAR wordt voor 4 % gereflecteerd en voor 87 % geabsorbeerd. PAR en NIR komen in ongeveer gelijke hoeveelheden in globale straling voor, wat dus betekent dat de hoeveelheid geabsorbeerde NIR meestal ongeveer de helft zal zijn van de hoeveelheid geabsorbeerde PAR. Het rij-effect heeft tot gevolg dat de hoeveelheid geabsorbeerde straling lager zal zijn. Voor diffuse PAR ligt dit vaak in de orde van 5-10 %. Hoe dit voor NIR zal zijn is niet bekend. Mogelijk is dit percentage voor NIR hoger, maar omdat NIR absorptie op zich veel lager is dan die van PAR, zal het verlies van NIR door het rij-effect waarschijnlijk niet groot zijn.

1. General introduction

The following 4 sections, written in English, contain the scientific part of this report. The sections are followed by appendices, some of which give further explanation on topics treated in the 4 sections. Most of the appendices contain the listing of programs used in the simulations.

Ratio of PAR to global radiation

Crop growth is very much dependent on the amount of PAR (Photosynthetically Active Radiaton) intercepted by the canopy. In crop growth models the flux PAR is commonly estimated by assuming that PAR is 45 % of global radiation, although the fraction is known to vary. Here, a regression model is developed of the ratio of PAR to global radiation, based on measuments of the PAR flux and global radiation. In the model also a dependency on the amount of clouds and the solar elevation is incorporated.
In addition, relations are developed for estimation of the fraction diffuse in the PAR flux and the fluxes diffuse and direct NIR (Near Infraread Radiation) from measured global radiation. The developed relations enable one to estimate more accurately the radiation climate inside the greenhouse and the amount of radiation absorbed by the canopy.

Simulation of dry matter production

In a model of the production of greenhouse crops it is necessary to know how much dry matter (i.e. biomass minus the water) is formed from the photosynthetic assimilates (the sugars) for each of the plant parts. For greenhouse crops very little is known about this conversion, although it is a very important parameter in each crop growth model. Therefore, the assimilate requirements (g dry matter per g sugars) were determined of leaves, stems and fruits of cucumber, tomato, sweet pepper and eggplant, based on chemical analysis of these plant parts.
Using the calculated assimilated requirements, the dry matter production was simulated in several experiments on cucumber, tomato and sweet pepper, and compared with measured productions.

Simulation of transpiration

An accurate prediction of transpiration of greenhouse crops is important for the control of the humidity in the greenhouse, which has a large influence on, among others, the quality of the harvested product, many aspects of crop growth, and disease development. Here, several model versions of a canopy transpiration model were developed, using several models of stomatal response. The stomatal models were taken from literature or developed based on some literature data; in two of the stomatal models stomatal conductance is calculated based on the rate of leaf photosynthesis, and in the other models conductance is related directly to environmental conditions. The simulated canopy transpiration is compared with measured canopy transpiration of tomato, cucumber and sweet pepper. A sensitivity analysis is done to investigate the effect of several climate variables and greenhouse and crop parameters on canopy transpiration.
Two special models are developed that account for the varying shade that a group of plants placed on a lysimeter is receiving during sunny days from neighbouring plants and from

construction elements. The models calculate the 2- and 3-dimensional position of crop rows and construction elements (gutters, ridges and beams) and the amount of shade they cast on a plant stand. In this way, it can be assessed how much measured transpiration, necessarily measured on only a few plants, could differ from transpiration of the crop as a whole. This information will be useful in a humidity control system that is based on measured canopy transpiration.

Model simplifications

Models are often needed in different levels of detaildness. I.e. in various diverse applications often some parameters or data are lacking, thereby necessitating simplification. Also, when the model is used as part of (super)model at a higher integration level, a certain amount of accuracy contained in the submodel is not needed as other parts of the supermodel lack precision. Another reason for model simplification could be increase of execution speed, e.g. in optimization algorithm's.
Here some simplified models were developed of leaf photosynthesis, canopy photosynthesis, and dry matter production. The photosynthesis models have sufficient accurary to be used in many crop growth models. The model of dry matter production is actually a set of simple conversion factors for relating incident radiation to dry matter production.

2. Estimation of PAR in global radiation

Summary

The modelling of the partitioning of global radiation in photosynthetically active radiation PAR) and near infrared radiation (NIR), and of the separation of these fluxes into diffuse and direct components is important in models aimed at predicting photosynthesis and transpiration of greenhouse crops. In present research the fraction diffuse in global radiation was related to the ratio between measured global radiation and extra-terrestrial radiation (K_g) for 10 minute intervals; in addition the ratio PAR to global radiation was related to K_g.
The ratio of PAR photon flux to global radiation was at intermediate and high radiation levels 2.03 µmol J^{-1} with standard deviation 0.1. At cloudy weather this ratio increased to values above 2.2 µmol J^{-1}. At low solar elevations (< 20°), the ratio was decreased by 5-10 %. The ratio of NIR to global radiation and the fractions diffuse in PAR and NIR were related to K_g based on literature data.

2.1. Introduction

In models predicting the rate of photosynthesis of greenhouse crops, the flux of Photosynthetically Active Radiation (PAR, 400-700 nm) must be known. (The energy flux is denoted here as Q_{pe}, in units J m^{-2} s^{-1}, and the photon flux as Q_{pp}, in units µmol m^{-2} s^{-1}). PAR can either be measured or can be estimated from measured global radiation (Q_g, 300-3000 nm, J m^{-2} s^{-1}) (Monteith & Unsworth, 1990). If the estimation of PAR from global radiation is sufficiently accurate, then no PAR-measurements would be necessary. This would be of advantage for future practical climate control based on crop models. It would also strengthen the validation of models with growth experiments in which no PAR has been recorded.

In crop growth models the PAR-energy flux Q_{pe} is often taken to be 45 % of global radiation (Jones, 1983; Monteith & Unsworth, 1990), based on the work of Moon (1940). However, several reports in the literature indicate that the ratio of PAR to global radiation, Q_{pe}/Q_g, could depend on, among others, the climate and the length of the measurement interval. Some of the variation reported appears also to be due to the fact that several authors measured 'PAR' in wave bands slightly different from the range 400-700 nm.

Significant variation in Q_{pe}/Q_g, associated with variation in cloudiness has been reported. From spectral data of Anonymous (1981a,b) at Ukkel, Belgium (51° N), it was calculated that the daily average Q_{pe}/Q_g at clear days varied from 0.40 in November-December to 0.48 in July. Daily average Q_{pe}/Q_g at cloudy days varied from 0.48 to 0.55. Britton & Dodd (1976) found daily Q_{pe}/Q_g to decrease with decreasing daily Q_g from 0.50 to 0.45 in the period October-February, and from 0.58 to 0.47 in the period April-August, at College Station, Texas (30° N). Howell *et al.* (1983) reported an average Q_{pe}/Q_g of 0.45, with small effects of clouds or day-length, at Fresno, California (36° N). The daily ratio of PAR-photon flux to global radiation was 2.04 ± 0.04 µmol J^{-1}. At a slightly different waveband Szeicz (1974) found the daily ratio $Q_{pe,300-700}/Q_g$ to increase from 0.48 to 0.51 when the daily fraction diffuse increased from 0.25 to 0.9, at Cambridge, UK (52° N). Stigter & Musahilba (1982) found daily $Q_{pe,300-700}/Q_g$ to be 0.51 at clear days and 0.63 at cloudy days, at Dar es Salaam, Tanzania (7° S). Instantaneous (i.e. half-hourly) $Q_{pe,300-700}/Q_g$ increased from 0.51 to 0.60 when the fraction diffuse increased from 0.1 to 1.

In some reports a slight dependency of Q_{pe}/Q_g, or of the ratio of 'PAR' to Q_g, on solar elevation (β) is apparent. From data of Anonymous (1981a,b) it appeared that at clear days the daily ratio Q_{pe}/Q_g was lower in winter time than in summer. At a slightly different waveband Velds et al. (1992) reported daily $Q_{pe,380\text{-}700}/Q_g$ to vary at clear days between 0.41 in winter to 0.46 in summer, at Cabauw (the Netherlands, 52° N). Szeicz (1974) found that $Q_{pe,300\text{-}700}/Q_g$ increased from 0.48 to 0.51 when solar elevation decreased from 60 to 10°. Other authors found little or no effect. Stanhill & Fuchs (1977) found in an arid climate half-hourly $Q_{pe,220\text{-}680}/Q_g$ for clear days to be about constant at 0.49 for solar elevation between 80 and 10°. Stigter & Musahilba (1982) found half-hourly $Q_{pe,300\text{-}700}/Q_g$ to be constant at about 0.51 at clear skies for β between 0 and 80°.

The fraction diffuse in PAR, $f_{dif,pe}$, can be important, as it affects both the total PAR transmittance of the greenhouse and crop photosynthesis. Theoretical considerations indicate that scattering of radiation by atmospheric gasses (Rayleigh-scattering) is larger in the shorter wavelengths, which tends, for clear skies, to increase the fraction diffuse in the PAR waveband compared to global radiation. Spitters *et al.* (1986) assumed $f_{dif,pe}$ for clear skies to be 40 % higher than the fraction diffuse in global radiation, $f_{dif,g}$, based on measurements by Anonymous (1981a,b). Weiss & Norman (1985) related the fraction diffuse in both PAR and NIR to locally potentially available PAR and NIR.

The modelling of the spectral distribution of solar radiation has become very sophisticated, and quite accurate predictions of the spectrum can be obtained for either overcast or completely clear skies (c.f. Bird & Hulstrom, 1983 and Justus & Paris, 1985). From these models the fraction PAR in global radiation could be calculated. However, these models are quite computation-intensive and need more parameters than are commonly available. Therefore a simple equation was developed to predict the flux PAR from global radiation, based on measurements of PAR and global radiation. The fraction diffuse in PAR and NIR were estimated based on literature data.

Definitions of symbols

	Description	Unit
$f_{dif,g}$	fraction diffuse in global radiation	-
$f_{dif,pe}$	fraction diffuse in PAR energy flux	-
$f_{dif,pp}$	fraction diffuse in PAR photon flux	-
f_c	apparent fraction clear sky	-
K_g	atmospheric transmission	-
Q_g	global radiation	$J\ m^{-2}\ s^{-1}$
Q_n	Near Infrared Radiation	$J\ m^{-2}\ s^{-1}$
Q_{pe}	PAR energy flux	$J\ m^{-2}\ s^{-1}$
Q_{pp}	PAR photon flux	$\mu mol\ m^{-2}\ s^{-1}$
Q_{uv}	Ultra-Violet radiation	$J\ m^{-2}\ s^{-1}$
β	solar elevation	degrees
subscripts		
dif	diffuse	
dir	direct	
ex	extra-terrestrial	

2.2. The data

2.2.1. Measurements

Measurements were done both on the PAR photon flux and global radiation. Part of the measurements were performed at a mobile weather station, in use adjacent to an experimental setup for crop photosynthesis measurements; locations were at Assen, and at Randwijk, and another part was done on the top of a roof of a root research facility, at the AB-DLO, Wageningen, all in the Netherlands (latitudes 51.5 - 52.5° N). Global radiation was measured with a solarimeter (Kipp & Zonen), the PAR photon flux was measured with a quantum sensor (Bottemanne Weather Instruments). Measurements were recorded over intervals of 288 seconds and averaged over 9.5 minute intervals. Measurements were performed at selected days in the period May 1992 until January 1994. Measurements at solar elevation below 10°, or at intensities below 5 J m^{-2} s^{-1} were not used. After exclusion of these data the set consisted of 2187 records.

2.2.2. Some additional data

Part of the model is based on data of Anonymous (1981a,b). The dataset consists of measurements on the instantaneous spectral distribution for both diffuse and direct radiation, at overcast and at clear skies, about 50 measurements in total. From this dataset the diffuse and direct energy flux in the UV, PAR and NIR wavebands, and the diffuse and direct photon flux in the PAR waveband could be calculated.

The calculations performed with the model of solar spectral irradiance of Justus & Paris (1985) were used to support some of the models assumptions. In a report of the International Commision on Illumination results are tabulated of calculations of this model on the spectral distributions of solar radiation of completely clear and overcast skies (CIE, 1989). From these spectral distributions the energy or photon flux in the UV, PAR and NIR wavebands were calculated.

2.3. The model

2.3.1. Ratio of the photon flux of PAR to global radiation

The simplest form of the prediction of Q_{pp}/Q_g was by assuming it to be constant

$$Q_{pp}/Q_g = a \qquad (2.1)$$

In the more detailed equations the atmospheric transmission, K_g, was chosen as the main predictor variable as the ratio of the PAR energy flux to global radiation appears to depend to some extent on the fraction diffuse in global radiation. K_g was calculated as the ratio of measured global radiation to global radiation outside the atmosphere, $Q_{g,ex}$,

$$K_g = \frac{Q_g}{Q_{g,ex}} \qquad (2.2)$$

where $Q_{g,ex}$ was calculated as the solar constant times the sine of the solar elevation (Spitters *et al.*, 1986). K_g is commonly used as the main predictor variable in the so-called 'Liu & Jordan'-type models for estimation of the fraction diffuse in global radiation.

As a second predictior was chosen solar elevation β. Data from Anonymous indicate that the daily Q_{pe}/Q_g at clear skies is decreasing with shorter daylengths and, consequently, with lower average solar elevations (cf. Fig. 2.1).

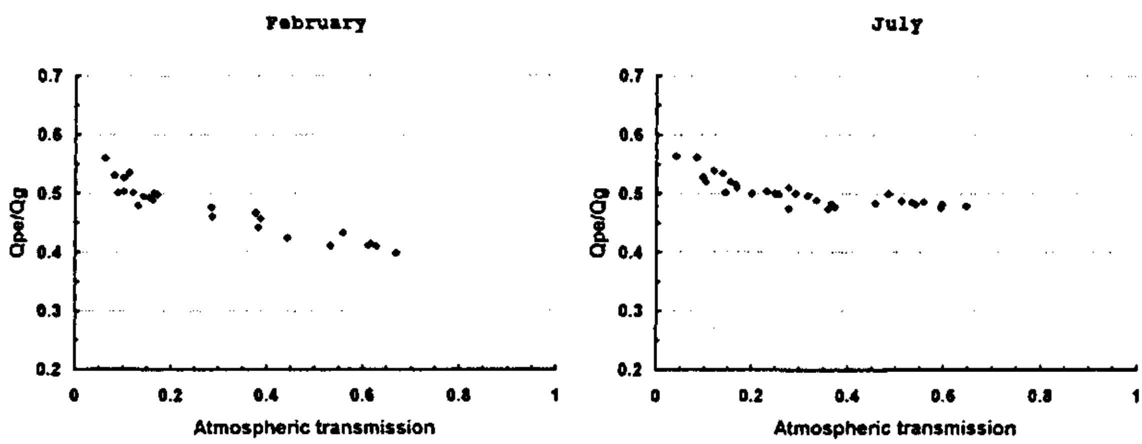


Figure 2.1 The relation between measured daily fraction PAR in global radiation to daily atmospheric transmission, at Ukkel, for February 1980 and July 1980

A negative-exponential function was used to relate Q_{pp}/Q_g to Kg. The form without β was

$$Q_{pp}/Q_g = a - (1 - exp(-b\, K_g^{\,c}) \qquad (2.3)$$

where a, b and c are parameters, and with use of the solar elevation

$$Q_{pp}/Q_g = a - f_m\,(1 - exp(-b\, K_g^{\,c}) \qquad (2.4a)$$

where f_m is an intermediate variable·f_m was modelled to depend on β using an exponential function

$$f_m = d\, exp(\, e \,/\, sin\, \beta\,) \qquad (2.4b)$$

where d and e are parameters.

The equations were fitted to the data by minimizing the sum of squares of the predicted and measured flux Q_{pp}.

Results

The general trend in the ratio Q_{pp}/Q_g was to decrease from a maximal value of about 2.6, at lowest K_g and highest cloud amounts, to about 2.10-2.05 for K_g at 0.2-0.3, and to remain approximately constant at 2.03 ± standard deviation (SD) 0.1 at higher K_g (Figs. 2.2, 2.3 and 2.4). Solar elevations below 20° appeared to slightly decrease Q_{pp}/Q_g. Only for β lower than 10° would Q_{pp}/Q_g significantly be decreased. Eqns 2.3 and 2.4 were only little better in predicting Q_{pp}/Q_g than the assumption of a constant ratio Q_{pp}/Q_g, as indicated by the small decrease in the standard error of estimate (Table 2.1). Most of the differences with the constant value occurred at low radiation intensities (caused by high cloud amounts or low solar elevations) that have little weight in the fit.

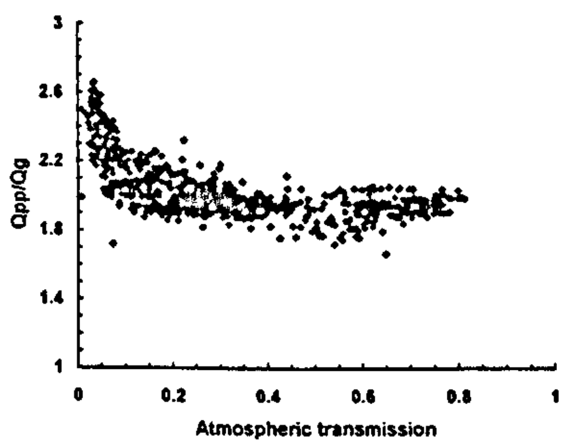


Figure 2.2
The relation between measured ratio of PAR photon flux to global radiation (Q_{pp}/Q_g, *μmol J⁻¹*) and atmospheric transmission, in the period May to November 1993

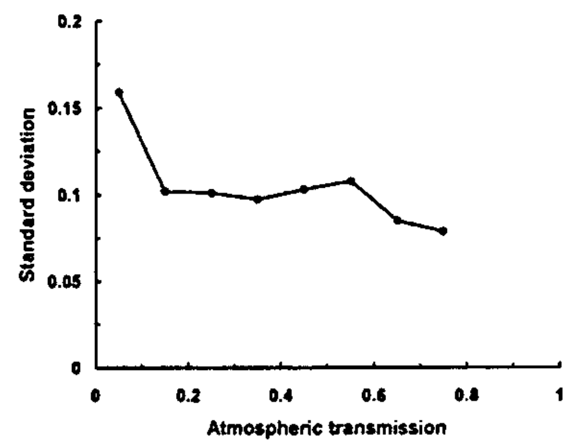


Figure 2.3
Standard deviation of measured ratio of PAR photon flux to global radiation (Q_{pp}/Q_g, μmol J^{-1}) as dependent on atmospheric transmission. Atmospheric transmission values were divided into classes of 0.1.

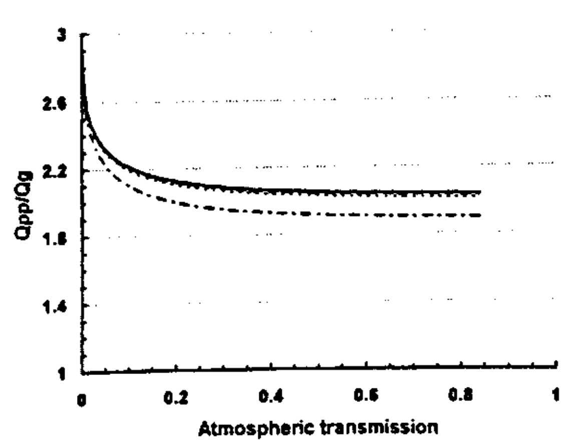


Figure 2.4 Modelled ratio of PAR photon flux to global radiation (Q_{pp}/Q_g, μmol J^{-1}) as a function of atmospheric transmission, according to Eqn 2.4

Table 2.1 Parameters, coefficients of determination (r^2) and standard errors of estimate (SEE) for the fit of regression relation for the ratio Q_{pp}/Q_g.

Eqn	Parameters					r^2	SEE
	a	*b*	*c*	*d*	*e*		%
2.1	2.03	-	-	-	-	0.9960	5.4
2.3	3.02	7.36	0.65	-	-	0.9962	5.2
2.4	2.89	4.48	0.51	0.84	0.033	0.9963	5.1

2.3.2. Estimation of the fraction diffuse in the PAR photon flux

2.3.2.1 PAR photon flux in diffuse and direct global radiation

From measurements of Anonymous (1981a,b) it was calculated that the ratio of the diffuse PAR photon flux to diffuse global radiation, $Q_{pp,dif}/Q_{g,dif}$, at clear skies varied between 2.6 and 3.4, with average 2.95 (Fig. 2.5). The ratio calculated by the model of Justus & Paris (CIE, 1989) lies in the lower part of this measured range (Table 2.2). Their model also indicates that the dependency of $Q_{pp,dif}/Q_{g,dif}$ on β should be small.

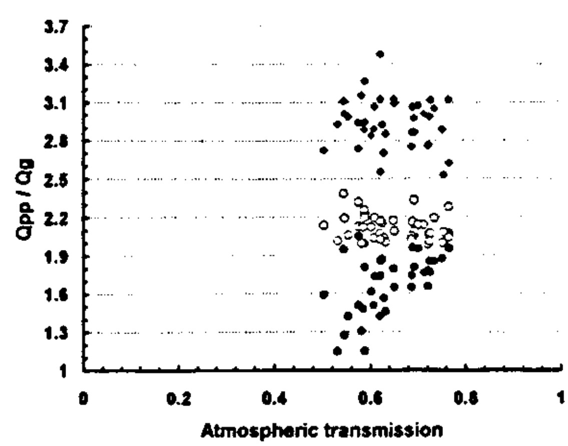


Figure 2.5 The relation between the ratio of the PAR photon flux to global radiation and atmospheric transmission, for clear skies, calculated from spectral measurements by Anonymous (1981a,b), for the total fluxes (open circles, Q_{pp}/Q_g), the diffuse fluxes (diamonds, $Q_{pp,dif}/Q_{g,dif}$) and the direct fluxes (closed circles, $Q_{pp,dir}/Q_{g,dir}$)

The range of the ratio of the direct PAR photon flux to direct global radiation, $Q_{pp,dir}/Q_{g,dir}$, at clear skies, as measured by Anonymous was somewhat larger, i.e. 1.6 ± 0.5 (Fig. 2.5). According to the calculations by CIE (1989), $Q_{pp,dir}/Q_{g,dir}$ is affected by the solar elevation, and is very low at solar elevation 10°.

Table 2.2 Characteristics of some clear skies, calculated from the solar spectral irradiance calculated with the model of Justus & Paris (in CIE, 1989).
Model parameter sets:
A: aerosol optical depth 0.2, ozone = 0.3 cm, precipitable water = 2.0 cm;
B: aerosol optical depth 0.4, ozone = 0.3 cm, precipitable water = 2.0 cm;
C: aerosol optical depth 0.0, ozone = 0.6 cm, precipitable water = 4.0 cm.
Aerosol optical depth τ is extinction by aerosols at λ = 500 nm, according to exp(-τ m) where m is relative airmass.

			Solar elevation					
			10° A	30° A	42° A	90° A	90° B	90° C
K_g		-	0.51	0.71	0.75	0.8	0.78	0.77
Q_g	Total	$J\ m^{-2}\ s^{-1}$	124.	486.	683.	1091.	1075.	1055.
$Q_{g,dir}$	Direct		63.	357.	538.	908.	797.	1003.
$Q_{g,dif}$	Diffuse		60.	128.	146.	183.	278.	52.
Q_{pp}/Q_g	Total	$\mu mol\ J^{-1}$	1.96	2.05	2.06	2.06	2.05	2.09
$Q_{pp,dir}/Q_{g,dir}$	Direct		1.30	1.83	1.88	1.95	1.85	2.07
$Q_{pp,dif}/Q_{g,dif}$	Diffuse		2.67	2.68	2.74	2.63	2.63	2.48
Q_{pe}/Q_g	Total	-	0.43	0.45	0.45	0.45	0.45	0.46
$Q_{pe,dir}/Q_{g,dir}$	Direct		0.27	0.39	0.40	0.42	0.40	0.45
$Q_{pe,dif}/Q_{g,dif}$	Diffuse		0.60	0.61	0.63	0.60	0.59	0.60
Q_{pp}/Q_{pe}	Total	$\mu mol\ J^{-1}$	4.59	4.58	4.58	4.57	4.57	4.57
$Q_{pp,dir}/Q_{pe,dir}$	Direct		4.90	4.87	4.65	4.62	4.64	4.59
$Q_{pp,dif}, Q_{pe,dif}$	Diffuse		4.45	4.40	4.39	4.37	4.42	4.15
Q_n/Q_g	Total	-	0.52	0.50	0.49	0.49	0.49	0.48
$Q_{n,dir}/Q_{g,dir}$	Direct		0.73	0.59	0.57	0.54	0.57	0.50
$Q_{n,dif}/Q_{g,dif}$	Diffuse		0.29	0.24	0.21	0.22	0.26	0.06
Q_{uv}/Q_g	Total	-	0.049	0.051	0.054	0.058	0.058	0.055
$Q_{uv,dir}/Q_{g,dir}$	Direct		0.002	0.020	0.026	0.036	0.031	0.042
$Q_{uv,dif}/Q_{g,dif}$	Diffuse		0.099	0.139	0.157	0.168	0.134	0.320

The fraction diffuse in the PAR photon flux appeared to depend somewhat on the solar elevation (Fig. 2.6), but a significant scatter was present.

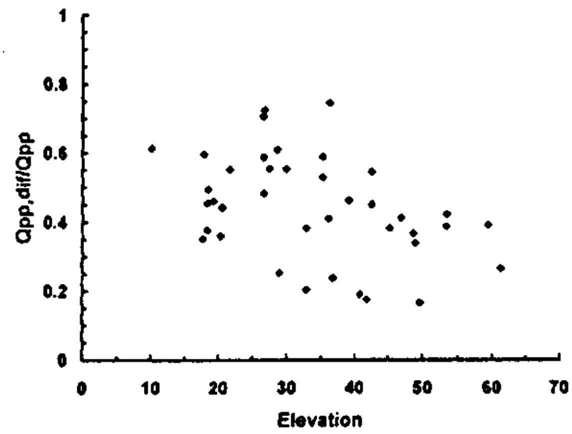


Figure 2.6 The relation between the fraction diffuse in the PAR photon flux and solar elevation, for clear skies, calculated from spectral measurements by Anonymous (1981a,b)

2.3.2.2 The fraction diffuse in PAR in relation to fraction diffuse in global radiation

In the measurements of Anonymous (1981a,b) the average ratio $Q_{pp,dif}/Q_{g,dif}$ was 1.38 times the average ratio Q_{pp}/Q_g. From $Q_{pp,dif}/Q_{g,dif} = 1.38 * Q_{pp}/Q_g$ it follows that $f_{dif,pp} = 1.38 * f_{dif,g}$, i.e the fraction diffuse in the PAR photon flux was 1.38 times the fraction diffuse in global radiation. From these measurements it appeared further that, for clear skies, the fraction diffuse in the PAR photon and energy flux was quite correlated with the fraction diffuse in global radiation (Fig. 2.7). Also from the data of McCartney (1978) it appeared that the ratio of fraction diffuse in the PAR energy flux ($f_{dif,pe}$), for clear skies was strongly correlated with the fraction diffuse in global radiation. Both $f_{dif,pe}$ and $f_{dif,g}$ were linearly related with turbidity. Turbidity is normally defined as the atmospheric attenuation at some specified wavelength; in case of McCartney (1978) this was at 500 nm. From his data the ratio $f_{dif,pe} / f_{dif,g}$ was calculated to be 1.31 ± 0.03. The ratio increased somewhat with turbidity for higher solar elevations. The ratio $f_{dif,pp} / f_{dif,g}$ calculated by CIE (1989) (Table 2.2) varied between 1.28 and 1.33. Both the calculations by CIE (1989) and the measurements by McCartney (1978) point to the occurrence of the highest ratio's at intermediate solar elevations.

The reasons for the discrepancy between the measurements of Anonymous (1981a,b) and McCartney (1978) are not clear.

For clear skies the ratio $f_{dif,pp}/f_{dif,g}$ is somewhat lower than the ratio $f_{dif,pe}/f_{dif,g}$ (see below), and is, based on the measurements from McCartney (1978) and the calculations by CIE (1989), here taken to be 1.3.

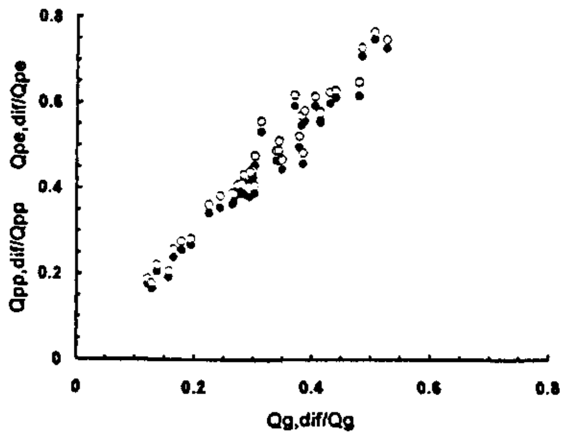


Figure 2.7 The relation between the fraction diffuse in the PAR flux to the fraction diffuse in global radiation, for clear skies, as calculated from the spectral measurements of Anonymous (1980a, b).

Closed circles: ratio $f_{dif,pp}$ / $f_{dif,g}$. A line fitted through the data points and forced through the origin would have slope 1.38.

Open circles: ratio $f_{dif,pe}$ / $f_{dif,g}$. A line fitted through the data points and forced through the origin would have slope 1.45

No data are available on the fraction diffuse in the PAR photon flux for partly cloudy skies. Therefore, it was assumed that the ratio $f_{dif,pp}/f_{dif,g}$ decreases linearly with a decreasing K_g. The average K_g for clear skies was estimated at 0.8. The highest K_g at overcast skies at which all global radiation as measured for 10 minute intervals is still diffuse, was estimated at 0.3 (Gijzen, in prep.). Based on this, an apparent sky clearness, f_c, was introduced, decreasing from 1 to 0 for K_g decreasing from 0.8 to 0.3

$$f_c = \frac{K_g - 0.3}{0.8 - 0.3} \qquad (2.5)$$

Thus, it was assumed that with K_g decreasing from 0.8 to 0.3 the difference between $f_{dif,pp}$ and $f_{dif,g}$ decreases until both are 1.

$$f_{dif,pp} = min\ \{1,\ f_{dif,g} * (1. + f_c * 0.3)\} \qquad (2.6)$$

The fraction diffuse in global radiation in 10-minute intervals can be calculated from a regression relation between measured fraction diffuse in global radiation and K_g, based on diffuse radiation measurements at Naaldwijk (Gijzen, in prep.). This regression relation has parameters slightly different from the relation given for hourly intervals by Spitters *et al.* (1986).

Table 2.3 Characteristics of cloudy skies of various cloud optical depths, calculated from the solar spectral irradiance calculated with the model of Justus & Paris (in CIE, 1989). Cloud optical depth is the 'atmospheric extinction coefficient' at $\lambda = 500$ nm.

		Cloud optical depth			
		3	10	30	100
K_g	-	0.65	0.43	0.22	0.08
Q_g	J m^{-2} s^{-1}	597.	388.	200.	69.
Q_{pp}/Q_g	µmol J^{-1}	2.16	2.35	2.47	2.71
Q_{pe}/Q_g	-	0.47	0.51	0.54	0.60
Q_{pp}/Q_{pe}	µmol J^{-1}	4.57	4.56	4.54	4.52
Q_n/Q_g	-	0.47	0.41	0.37	0.30
Q_{uv}/Q	-	0.06	0.07	0.08	0.09
Q_{uv}/Q_{pe}	-	0.12	0.13	0.14	0.15

2.3.3. Estimation of the total, diffuse and direct PAR energy flux

2.3.3.1. Total PAR

The factor for converting total PAR photon flux to the total PAR energy flux appears to be rather constant. For clear skies the ratio Q_{pp}/Q_{pe} (µmol J^{-1}) was found by McCree (1972) to be 4.57. McCartney (1978) found it to range from 4.51 to 4.62, with average 4.54. It increased somewhat with decreasing solar elevation and with increasing turbidity. From the spectral measurements by Anonymous (1981a,b) it was calculated to range from 4.56 to 4.66, with average 4.59 (Fig. 2.8). It was calculated by CIE (1989) to be 4.58 (Table 2.2). Note that if the energy would evenly be distributed over all wavelength's from 400 to 700 nm the ratio Q_{pp}/Q_{pe} would be equal to 4.597. For overcast skies Q_{pp}/Q_{pe} measured by Anonymous (1981a,b) varied from 4.48 to 4.59. CIE (1989) calculated it to decrease from 4.56 to 4.53 when the thickness of the cloud cover changed from small to very large (Table 2.3). Here, the ratio Q_{pp}/Q_{pe} was taken to be 4.57.

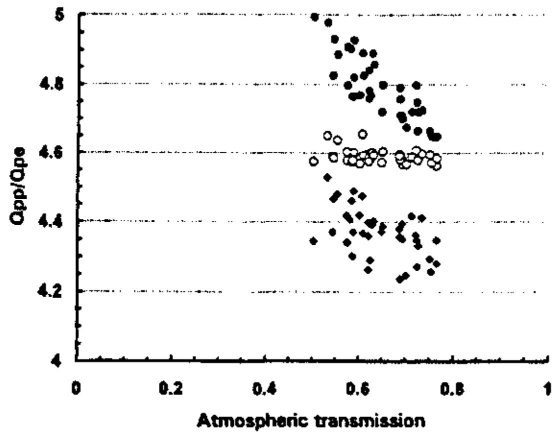


Figure 2.8 The relation between the ratio of PAR photon flux to PAR energy flux (in μmol J^{-1}) and atmospheric transmission, calculated from spectral measurements by Anonymous (1981a,b), for clear skies, for the total fluxes (open circles, Q_{pp}/Q_{pe}), the diffuse fluxes (diamonds, $Q_{pp,dif}/Q_{pe,dif}$) and the direct fluxes (closed circles, $Q_{pp,dir}/Q_{pe,dir}$)

By dividing Q_{pp}/Q_g, as calculated by equations 2.1, 2.3 and 2.4, by 4.57, the ratio Q_{pe}/Q_g was found. See Fig. 2.9 for the fit for the dependance on K_g alone. Assuming a constant fraction would give $Q_{pe}/Q_g = 0.445$. Q_{pe}/Q_g at a dense cloud cover would be about 2.6 / 4.55 = 0. 57.

2.3.3.2. Diffuse and direct PAR

From measurements of Anonymous (1981a,b) it was calculated that the fraction PAR energy in diffuse global radiation, $Q_{pe,dif}/Q_{g,dif}$, at clear skies varied between 0.58 and 0.79 and was on average 0.67 (Fig. 2.10). With $Q_{pp,dif}/Q_{g,dif} = 2.95$ (see above), the average ratio $Q_{pp,dif}/Q_{pe,dif}$ was calculated to be 4.39 μmol J^{-1}. It varied between 4.24 and 4.53 (Fig. 2.8).

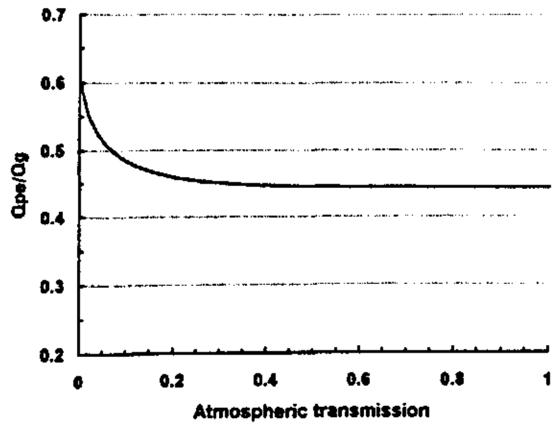


Figure 2.9 The relation between the ratio of PAR energy flux to global radiation (Q_{pe}/Q_g) and atmospheric transmission, calculated using Eqn 2.4 and using the conversion $Q_{pp}/Q_{pe} = 4.57$

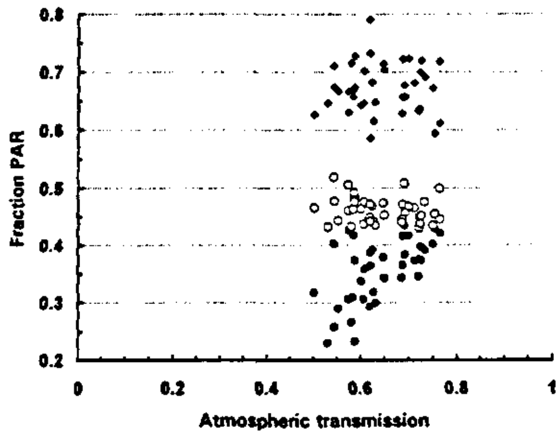


Figure 2.10 The relation between the fraction PAR energy in global radiation and atmospheric transmission, for clear skies, calculated from spectral measurements by Anonymous (1981a,b), for the total fluxes (open circles, Q_{pe}/Q_g), the diffuse fluxes (diamonds, $Q_{pe,dif}/Q_{g,dif}$) and the direct fluxes (closed circles, $Q_{pe,dir}/Q_{g,dir}$)

The fraction PAR in direct global radiation, $Q_{pe,dir}/Q_{g,dir}$, at clear skies varied between 0.23 and 0.45, and was on average 0.37 (Fig. 2.10). The ratio $Q_{pp,dir}/Q_{pe,dir}$ was calculated to be on average 4.74 µmol J^{-1} (Fig. 2.8).

Similarly as with the PAR photon flux, the fraction diffuse in the PAR energy flux was correlated with fraction diffuse in global radiation. The slope of the fit of $f_{dif,pe}$ on $f_{dif,g}$ was 1.45 (Fig. 2.7). The weighted average ratio $f_{dif,pe}/f_{dif,g}$ was 1.46. This ratio is somewhat higher than the ratio for the PAR photon flux (i.e. 1.38) as photons in diffuse PAR contain on average more energy than the average photon in the whole PAR spectrum (i.e. 4.57 and 4.39 µmol J^{-1} for total and diffuse PAR, respectively). The ratio $f_{dif,pe} / f_{dif,g}$ calculated by CIE (1989) (Table 2.2) varied between 1.31 and 1.40.

In the model, the ratio $f_{dif,pe}/f_{dif,g}$ is taken to be somewhat higher than for the PAR photon flux, and is set at 1.35. As with the PAR photon flux (Eqn 2.6), interpolation to partly cloudy skies is done based on the apparent fraction clear sky

$$f_{dif,pe} = min\{\, 1,\ f_{dif,g} * (1.0 + f_c * 0.35) \,\} \qquad (2.7)$$

2.3.4. Estimation of the total, diffuse and direct NIR and UV fluxes

Diffuse and direct NIR were calculated by subtracting both the energy flux of PAR and the flux UV (300-400 nm, J m^{-2} s^{-1}) from global radiation. The flux UV was estimated as a fraction of global radiation, both for the diffuse and the direct flux.

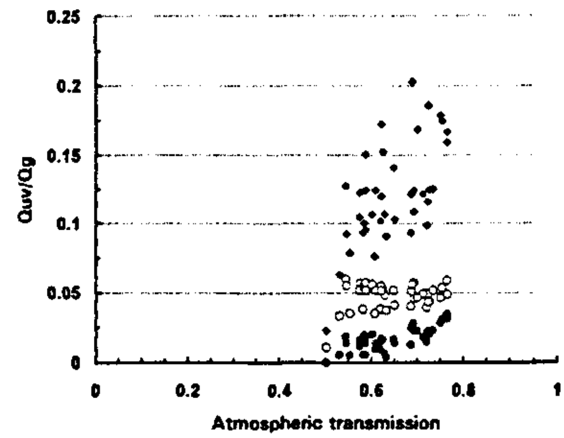


Figure 2.11 The relation between the fraction UV in global radiation and atmospheric transmission, for clear skies, calculated from spectral measurements by Anonymous (1981a,b), for the total fluxes (open circles, Q_{uv}/Q_g), the diffuse fluxes (diamonds, $Q_{uv,dif}/Q_{g,dif}$) and the direct fluxes (closed circles, $Q_{uv,dir}/Q_{g,dir}$)

The fraction UV in the diffuse global radiation, as measured by Anonymous (1981a,b), was for clear skies on average 0.12 (cf. Fig. 2.11). As the fraction NIR in diffuse global radiation was about 0.21, this means that the fraction UV in global radiation becomes relatively important for estimation of the flux diffuse NIR. For cloudy skies the ratio Q_{uv}/Q_g was 0.065. Here it is assumed that the fraction UV in diffuse global radiation decreases with decreasing K_g, until global radiation is completely diffuse.

$$Q_{uv,dif}/Q_{g,dif} = 0.05 + f_c * 0.07$$

The fraction UV to Q_g was set at 0.05, so that the direct flux UV could be found from the difference between total UV and diffuse UV.

2.3.5. Discussion

The measured ratio Q_{pp}/Q_g was on average 2.03, and was similar to the 2.04 measured by Howell *et al.* (1983) and in the lower end of the range reported by Britton & Dodd (1976). The calculated ratio Q_{pe}/Q_g was on average 0.45, i.e. similar to values found Weiss & Norman (1984). When assuming that the ratio UV to global radiation is about 5-6 % (Tables 2.2 & 2.3), then this value of Q_{pe}/Q_g is also similar to the average ratio's found by Szeicz (1976), Stanhill & Fuchs (1979) and Stigter & Musahilba (1982). Thus its value appears to be rather stable among diverse climates. It was measured that the ratio Q_{pe}/Q_g was higher at cloudy skies, i.e. about 0.55 for heavy overcast skies. This is about equal to the findings of Britton and Dodd (1976), and is comparable to the value of 0.63 for $Q_{pe,300\text{-}700}/Q_g$ of Stigter & Musahilba (1982). Increase of the PAR fraction in global radiation by clouds is expected because water is mainly absorbing in the NIR waveband (Iqbal, 1983).
The ratio PAR to global radiation was measured to be decreased significantly only at low solar elevations, i.e. lower than 20°. The measurements are on this point not very reliable as radiation intensities are low, and the low angles of incidence could cause significant measurement errors. However, this result is supported by Anonymous (1981a,b) and Velds *et al.* (1992), who measured a daily Q_{pe}/Q_g of about 0.40 at clear days during winter months, at latitudes 51° -

52°. The decrease at low β is somewhat different from the absence of any effect of β (for angles above 10°) found by Stanhill & Fuchs (1977) and Stigter & Musahilba (1982), or the increase in ratio PAR to global radiation with decreasing β as found by Szeicz (1974). Calculations by CIE (1989) (Table 2.2), calculations presented by Szeicz (1974) and measurements in Switzerland referred to by Szeicz (1974) all indicate decreases in the ratio for β below 30°, with the ratio Q_{pe}/Q_g being decreased for β at 10° by about 10 %. Thus, it appears that at low β the ratio is quite sensitive to atmospheric conditions. Molecular scattering of radiation will tend to deplete the PAR waveband as molecular scattering is larger in this waveband and scattering at low β will increase the apparent reflection of the atmosphere. On the other hand could the increased pathlength at low solar angles of radiation cause significant absorption of NIR by water vapour. E.g. the lower PAR content in global radiation at clear days in winter at Cabauw (the Netherlands) was attributed by Velds *et al.* (1992) to the dry easterly winds occuring specifically at sunny weather. This could also have contributed to the lower ratio Q_{pp}/Q_g measured here.

2.4. References

Anonymous, 1981a.
Distribution spectrale du rayonnement solaire a Uccle. Miscellanea Serie B No. 52. 1er semestre 1980. Section de Radiometrie. Institut Royal Meteorologique de Belgique.

Anonymous, 1981b.
Distribution spectrale du rayonnement solaire a Uccle. Miscellanea Serie B No. 53. 2e semestre 1980. Section de Radiometrie. Institut Royal Meteorologique de Belgique.

Bird, R.E. & R.L. Hulstrom, 1983.
Availability of SOLTRAN 5 solar spectral model. Solar Energy 30: 379.

Bot, G.P.A., 1983.
Greenhouse climate: from physical processes to a dynamic model. Diss., Agric. Univ., Wageningen, 240 pp.

Britton, C.M. & J.D. Dodd, 1976.
Relationships of photosynthetically active radiation and shortwave radiation. Agric. Meteorol. 17: 1-7.

CIE, 1989.
Commission Internationale de L' Eclairage. Technical Report. Solar Spectral Irradiance. Publ. No CIE 85.

Gijzen, H., 1992.
Simulation of photosynthesis and dry matter production of greenhouse crops. CABO-DLO/TPE Simulation Report nr. 28. 69+49 pp.

Howell, T.A., D.W. Meek & J.L Hatfield, 1983.
Relationship of photosynthetically active radiation to shortwave radiation in the San Joaquin Valley. Agric. Meteor. 28: 157-175.

Iqbal, M., 1983.
An introduction to solar radiation. Academic Press, Toronto, 390 pp.

Jones, H.G., 1983.
Plants and microclimate. Cambridge University Press, Cambridge. 323 pp.

Jong, J.B.R.M. de, 1980.
Een karakterisering van de zonnestraling in Nederland. Doctoraalverslag Vakgroep Fysische Aspecten van de Gebouwde Omgeving, afd. Bouwkunde, en Vakgroep Warmte- en Stromingstechnieken afd. Werktuigbouwkunde, Technische Hogeschool, Eindhoven, the Netherlands, 97+67 pp.
Justus, C.G & M.V. Paris, 1985.
A model for solar spectral irradiance at the top and the bottom of a cloudless atmosphere. J. Climate Appl. Meteor. 24: 193-205.
McCartney, H.A., 1978.
Spectral distribution of solar radiation. II: global and diffuse. Quart. J. R. Met. Soc. 104: 911-926.
McCree, K.J., 1972.
Test of current definitions of photosynthetically active radiation against leaf photosynthesis data. Agric. Meteorol. 10: 443-453.
Moon, P., 1940.
Proposed standard solar radiation curves for engineering use. J. Franklin Inst., 230: 583-618.
Monteith, J.L. & M.H. Unsworth, 1990.
Principles of environmental physics. Second Edition. Edward Arnold, London. 291 pp.
Spitters, C.J.T., 1986.
Seperating the diffuse and direct component of global radiation and its implications for modeling canopy photosynthesis. II. Calculation of canopy photosynthesis. Agric. For. Meteorol. 38: 231-242.
Spitters, C.J.T., H.A.M. Toussaint & J. Goudriaan, 1986.
Separating the diffuse and direct component of global radiation and its implication for modeling canopy photosynthesis. Part I. Components of incoming radiation. Agric. For. Meteorol. 28: 217-229.
Stanhill, G. & M. Fuchs, 1977.
The relative flux density of photosynthetically active radiation. J. Appl. Ecol. 14: 317-322.
Stigter, C.J. & V.M.M. Musabilha, 1982.
The conservative ratio of photosynthetically active to total radiation in the tropics. J. Appl. Ecol. 11: 617-636.
Szeicz, G., 1974.
Solar radiation for plant growth. J. Appl. Ecol. 11: 617-636.
Temps, R.C. & Coulson, K.L., 1977.
Solar radiation incident upon slopes of different orientations. Solar Energy 19: 179-184.
Velds, C.A., P.C.T van der Hoeven, J.M. Koopstra, W.R Raaff & W.H. Slob, 1992.
Zonnestraling in Nederland. Thieme-Baarn/KNMI. 170 pp.
Weiss, A. & J.M. Norman, 1985.
Partitioning solar radiation into direct and diffuse, visible and near-infrared components. Agric. For. Meteorol. 34: 205-213.

3. Simulation of dry matter production

Summary

The assimilate requirements (g glucose per g dry matter) were determined of leaves, stems and fruits of cucumber, tomato, sweet pepper and eggplant. The requirements were calculated from chemical composition according to the method of Penning de Vries *et al.* (1974). Calculations based on only the carbon and ash content (following Vertregt & Penning de Vries, 1987) gave deviating results.
Dry matter production was simulated in 3 experiments on cucumber, 2 on sweet pepper, and 1 on tomato. In general simulations somewhat overestimated measured dry matter productions.

3.1. Introduction

Crop dry matter is produced from the assimilates formed by photosynthetic CO_2 assimilation. The calculation of the rate of greenhouse crop CO_2 assimilation is described elsewhere (Gijzen, in prep.). Here the assimilate requirement (g assimilates (CH_2O) needed per 1 g of dry matter formed) for the formation of dry matter (DM) of leaves, stems and fruits in cucumber, tomato, sweet pepper and eggplant are estimated from chemical analysis of these plant parts. Two calculation methods were compared: the first one was the method according to Penning de Vries et al. (1974), in which calculation is based on the chemical composition of plant material of carbohydrates, proteins, lignin, fats, organic acids and minerals (this method is denoted as 'method PdV'); the second method was the method of Vertregt & Penning de Vries (1987), in which calculation is based on the carbon and ash content of the plant material (this method is denoted as 'method V&PdV').

The calculated values of assimilate requirements of cucumber, tomato and sweet pepper were used in simulations of the dry matter production. The assimilate requirement of eggplant plant parts were calculated for use in the ECP-model of the Horticultural Crops Research Station, at Naaldwijk.

In the second part of this chapter, simulated dry matter production is compared with measured dry matter production.

3.2. Estimation of the assimilate requirements

3.2.1. Material

Plant material from cucumber, tomato, sweet pepper and eggplant was collected at commercial farms, each crop at two farms. Some details on the crops are given in Table 3.1. Crops were grown on rockwool. It was recommended to give nitrogen in the nutrient solutions in the form of 14-16 mmol l^{-1} NO_3 and 1 - 1.25 mmol l^{-1} NH_4 (Sonneveld & van der Wees, 1988). Leaf and stem material was sampled from the older, middle and younger parts of the plants. Leaf petioles were, except for tomato leaves, considered part of the stems. Fruit material was sampled from harvestable fruits. Further details are described by Rijsdijk (1993).

Table 3.1 Some characteristics of the cucumber, tomato, sweet pepper and eggplant crops, each grown at 2 commercial farms, and dates at which plant material was sampled for chemical analysis.

Crop	Farmer	Cultivar	Planting date	Sampling dates
Cucumber	1	Ventura	15-01-1992	2 April, 29 April, 21 May
	2	Ventura	27-12-1991	2 April, 29 April, 27 May
Tomato	1	Pronto	25-11-1991	26 March, 16 July
	2	Pronto	03-12-1991	26 March, 16 July
Sweet pepper	1	Mazurka	20-11-1991	19 April, 9 July
	2	Eagle	20-11-1991	19 April, 9 July
Eggplant	1	Lunor	11-12-1991	12 March
	2	Lunor	19-11-1991	12 March

3.2.2. Chemical analysis

Plant material was oven dried at 80 °C during 24 hours and ground. Chemical analysis of the content was performed in leaves of all four species and in fruits of tomato of: carbon (C), total nitrogen (N), NO_3, crude fibre, fats, K, Ca, Mg and crude ash. In addition, alkalinity of the ash was determined. Fruits of cucumber, sweet pepper and eggplant, and some of the material of the stems were analysed for part of these chemical constituents. Contents were expressed on the basis of the dry weight determined after overnight drying of ground material at 105 °C. Carbon and N-content were determined by an automatic C-H-N analyser (Hereaus) according to the Dumas method. NO_3 was measured with a TRAACS (Bran & Lubbe) autoanalyser. Crude fibre was determined according to the Weende method, and fat content by extraction with petroleumbenzine 40-60 °C (Soxlet System HT). K, Ca and Mg were analysed by atomic absorption, using a Varian Techtron (AAS). Ash content was measured after combustion of the sample in a muffle furnace at 550 °C for minimal 1 hour. Alkalinity of the ash was determined by addition of excess HCl and back titration with NaOH to pH 5 (Dijkshoorn, 1973).

3.2.3. Calculations on the chemical composition

Protein content was calculated from 6.25 times the difference of total N and nitrate-N content. The lipid content was assumed to be equal to the fat content. Lignin was assumed to constitute 10 % of the crude fibre content. This figure was based on the data of Poorter (1991) who estimated, based on chemical analysis of 24 wild annual species, the percentage lignin of the fraction lignin+(hemi)cellulose in leaves and stems of fast growing species at 11-12 %.

The organic acid content was estimated from the ash alkalinity (eq kg^{-1}) and NO_3-content. The carbonate ions in the ash (CO_3^{2-}, eq. w. 30) originated from organic acids and nitrate. It was assumed that the average equivalent weight of the organic acids was 60, following Vertregt & Penning de Vries (1987). This is about equal to the equivalent weight of a 1:1 mixture of malate and citrate, or a 1:1 mixture of malate and oxalate. However, in cucumber leaves significant amounts of carbonate have been found (A. Schapendonk, AB-DLO, N. Vertregt, AB-DLO, pers. comm.); for leaves in this crop a 1:1 mixture of malate and carbonate was assumed, with equivalent weight of about 50. Consequently, the weight of organic acids could be estimated as 60 or 50 times the ash alkalinity-corrected for the NO^-_3-charge concentration (Dijkshoorn, 1973). The carbohydrate fraction was used to arrive at 100 % material, i.e. its fraction was taken as 1 minus the fractions of the other components.

Ash-alkalinity was not determined in some of the stem and fruit material. In those cases, it was estimated from the ash content and the ratio of ash-alkalinity to ash content as found in the other samples.

Estimation of mineral content

The estimation of the weight of the minerals was done in three ways. In the first one it was calculated by

$$m1 = ash - 30 * ash\ alkalinity + NO_3 \quad (3.1)$$

In the second way it was estimated as the sum of the weights of K, Ca, Mg and NO_3:

$$m2 = K + Ca + Mg + NO_3 \quad (3.2)$$

In the third way it was estimated following the approximation given by Vertregt & Penning de Vries (1987), which estimates the weight of the minerals equal to 67 % of the weight of the ash. This follows 1) from the rule of thumb that the weight of the inorganic ions equals the weight of the organic anions, and 2) from the fact that during ashing organic acids and NO_3, both with equivalent weight of about 60 are converted to carbonate with equivalent weight of 30. Thus

$$m3 = 0.67 * ash \quad (3.3)$$

The authors stated that their method was only applicable to leaf material with a salt content less than 130 g kg^{-1}, and to storage material with a salt content less than 60 g kg^{-1}. However, for comparison with the other calculation methods, *m3* was calculated for all the plant material.

Calculation of C-content

The C-content was measured directly by the C-H-N-analyser, but was also calculated from the C-content of the chemical constituents. C-content of organic matter was calculated, following Vertregt & Penning de Vries (1987), by

$$C_{om} = 0.535 * proteins + 0.444 * carbohydrates + 0.774 * lipids + 0.667 * lignin + 0.370 * organic\ acids \quad (3.4)$$

Organic matter is dry matter minus mineral content.

3.2.4. Calculation procedures of the assimilate requirement

Two calculation methods were followed to estimate the assimilate requirement from the chemical composition of plant material.

The first one was the method following Penning de Vries *et al.* (1974), in which chemical constituents are divided into 6 categories, i.e. proteins, carbohydrates, lipids, lignin, organic acids and minerals (method PdV). The assimilate requirement of a plant part is calculated from the assimilate requirement of each category and the fraction its constitutes in the total dry matter. The assimilate requirement of dry matter was calculated by

$$ASRQ_{dm} = 1.887 * proteins + 1.275*carbohydrates + 3.189*lipids + 2.231* lignin + 0.954*organic\ acids + 0.12*minerals \quad (3.5)$$

The coefficients were taken from Spitters *et al.* (1989). In the value of the assimilate requirement of the protein fraction it is implicitly assumed that energy for NO_3-reduction is supplied by the photosynthesis process.

The second calculation method was according to Vertregt & Penning de Vries (1987) (method V&PdV). In this procedure the assimilate requirement is estimated from the carbon content of the organic matter, C_{om}.

$$ASRQ_{om} = 5.39*C_{om} - 1.191 \quad (3.6)$$

By estimating minerals as 0.67 times ash content, $ASRQ_{dm}$ is calculated from

$$ASRQ_{dm} = 5.39*C_{dm} + (1.191 * 0.67) * ash - 1.191 \quad (3.7)$$

To account for translocation costs the value calculated by Eqn 3.7 must be multiplied by 1.053. These additional costs assume that 2 ATP is needed per glucose molecule for active passage of two membranes (Vertregt & Penning de Vries, 1987).

Table 2.2 . The chemical composition of leaves of cucumber, tomato, sweet pepper and eggplant. Each figure is mean of two samples that were taken at approximately the same date, from the same part of the plant, and were taken from each of the two growers. Leaves were sampled from different parts of the stems.
Contents are expressed in g kg-1; ash alkalinity in eq kg-1; assimilate requirements in g CH2O g-1 DM.

							Org		Ash								ASRQ	
		C	Protein	Lignine	Lipids	Carbo	acids	Ash	alkal	NO3	K	Ca	Mg	*m1*	*m2*	*m3*	Pen	V&PdV
Cucumber	branch, yo	421	314	11	29	372	148	207	3.23	16	37	35	8	126	95	138	1.341	1.306
	branch	382	255	9	19	373	172	262	3.86	26	32	56	9	172	123	176	1.222	1.135
	branch	363	240	10	27	277	254	334	5.44	22	34	82	12	193	150	223	1.178	1.087
	branch	357	228	8	14	320	197	342	4.24	19	25	102	13	233	159	229	1.117	1.060
	branch, old	345	215	10	34	223	307	387	6.37	14	35	100	14	210	163	259	1.141	1.027
	main stem	307	154	8	11	292	195	443	4.33	27	33	145	16	340	221	297	0.943	0.859
Tomato	top	432	235	13	20	557	58	136	1.33	22	49	11	3	118	86	91	1.314	1.309
	top	453	208	14	11	628	62	107	1.07	2	33	13	4	76	51	71	1.328	1.407
	middle	384	213	15	22	442	97	233	2.48	53	62	27	6	211	147	156	1.186	1.121
	middle	419	203	15	30	482	137	198	2.38	6	29	31	6	133	72	133	1.274	1.288
	bottom	385	184	15	39	423	154	251	2.92	22	36	47	7	186	112	168	1.213	1.140
	bottom	369	175	16	15	430	100	275	2.90	77	62	37	7	264	182	184	1.089	1.071
Sw pepper	top	430	239	7	24	506	132	151	2.43	14	43	25	6	91	88	101	1.326	1.313
	top	452	311	7	34	427	120	156	2.17	11	49	26	6	101	92	104	1.382	1.442
	middle	387	273	8	44	293	225	255	4.18	27	61	52	11	156	150	171	1.282	1.157
	middle	389	255	9	62	280	252	261	4.43	14	59	55	11	142	139	175	1.314	1.171
	bottom	345	201	8	32	321	204	309	4.26	53	66	66	16	234	202	207	1.131	0.961
	bottom	337	175	11	32	286	289	326	5.62	50	65	73	16	207	204	218	1.123	0.933
Eggplant	top	430	305	12	22	419	120	171	2.36	22	53	26	4	122	105	114	1.336	1.330
	middle	378	203	15	31	395	155	243	3.72	71	57	52	5	202	185	162	1.190	1.093

Table 2.3. The chemical composition of stems of cucumber, tomato, sweet pepper and eggplant. Each figure is mean of two samples that were taken at approximately the same date, from the same part of the stem, and were taken from each of the two growers.
Contents are expressed in g kg-1; ash alkalinity in eq kg-1; assimilate requirements in g CH2O g-1 DM.

		C	Protein	Lignine	Lipids	Carbo	Org acids	Ash	Ash alkal	NO3	K	Ca	Mg	*m1*	*m2*	*m3*	ASRQ Pen	V&PdV
Cucumber	branch	323	88	30	9	548	39	222	2.91*	151				286		149	1.032	0.763
	branch	347	106	31	8	554	38	214	2.77	132				263		143	1.069	0.892
	main stem	329	90	32	8	539	35	234	2.97*	154				295		156	1.024	0.807
	main stem	371	93	37	6	601	48	192	2.27	91	74	20	2	215	184	129	1.116	1.010
Tomato	top	437	92	31	10	716	59	119	1.07	5	45	6	2	91	58	79	1.256	1.323
	middle	401	69	44	6	688	41	149	1.42*	46				152		100	1.181	1.144
	bottom	394	86	42	6	654	52	163	1.61	46	51	16	5	161	117	109	1.176	1.116
	bottom	379	73	42	3	673	24	161	1.53*	70				185		108	1.146	1.032
Sw pepper	top	407	102	33	8	642	71	152	2.05	53	56	15	6	144	130	102	1.195	1.181
	middle	399	97	40	7	613	79	174	2.28*	60				165		116	1.170	1.156
	bottom	467	70	56	3	786	23	62	0.75	23	19	7	4	62	53	41	1.298	1.448
Eggplant	middle	427	101	46	5	706	51	104	1.24*	25				91		69	1.269	1.256

* Ash alkalinity was estimated; consequently mineral contents and organic acids contents were also estimated based on the estimated ash alkalinity.

Table 2.4 . The chemical composition of fruits of cucumber, tomato, sweet pepper and eggplant. Each figure is mean of two samples that were taken at approximately the same date, and were taken from each of the two growers.
Contents are expressed in g kg-1; ash alkalinity in eq kg-1; assimilate requirements in g CH2O g-1 DM.

	C	Protein	Lignine	Lipids	Carbo	Org acids	Ash	Ash alkal	NO3	K	Ca	Mg	*m1*	*m2*	*m3*	ASRQ Pen	ASRQ V&PdV
Cucumber	453	230	9	5	602	41	128	0.82 *	9				112		86	1.293	1.422
	454	254	11	7	542	60	150	1.22	14	63	4	4	127	84	101	1.288	1.446
Tomato	471	127	13	27	550	60	253	1.00	1	52	1	2	223	56	169	1.141	1.631
	488	113	13	30	684	39	195	0.66	0	39	1	2	121	42	94	1.261	1.631
Sw pepper	500	157	13	45	692	28	77	0.50 *	2				64		52	1.388	1.648
	494	144	15	41	680	37	100	0.66	2	34	1	9	83	46	67	1.348	1.634
Eggplant	451	125	16	3	773	20	70	0.45 *	7				63		47	1.292	1.362

* Ash alkalinity was estimated; consequently mineral contents and organic acids contents were also estimated based on the estimated ash alkalinity.

3.2.5. Results

Results of the chemical analyses and of the calculations based thereupon are given in Tables 3.2, 3.3, and 3.4, for leaves, stems and fruits, respectively. These figures are averages of two samples, each taken at two different growers, from the same plant part, and taken about the same date. Chemical compositions differed little between samples within a pair.

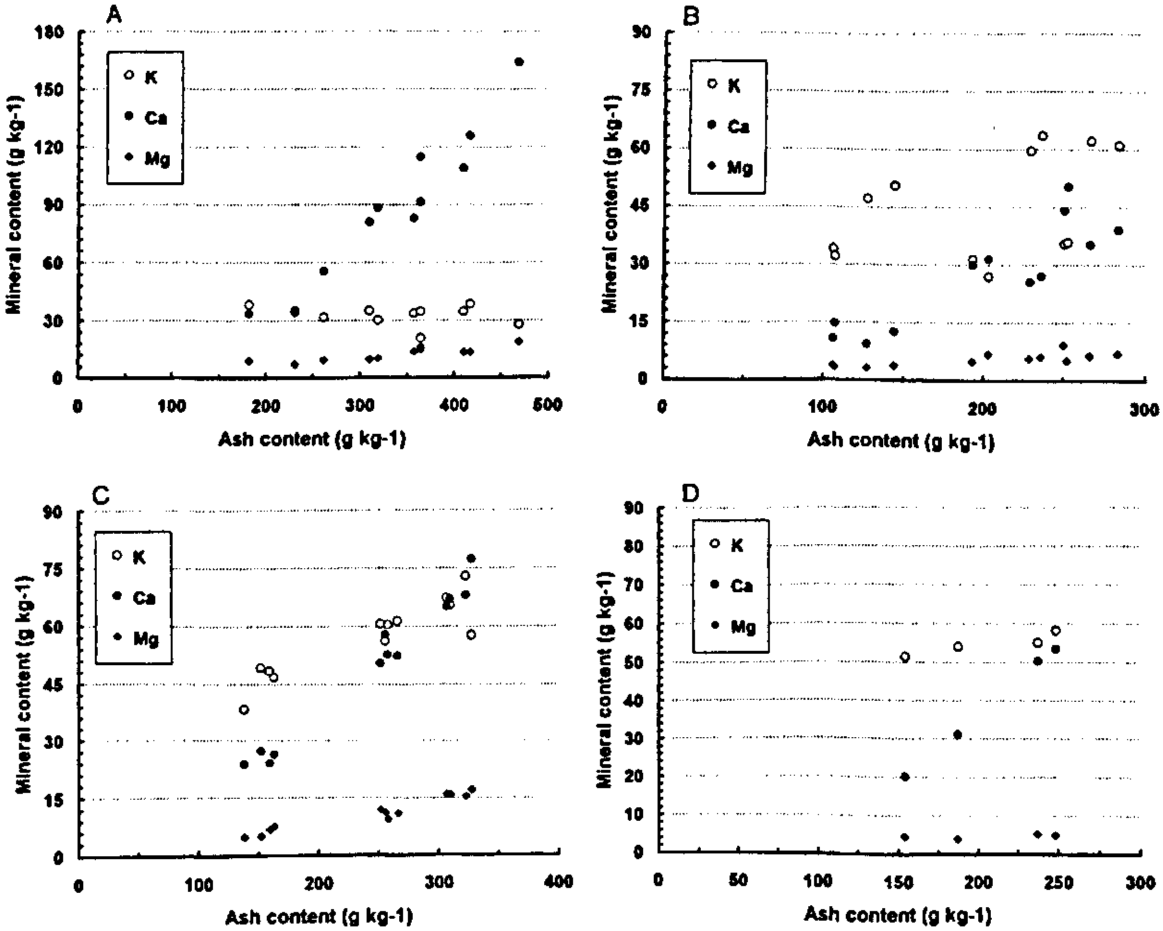


Figure 3.1 The relation between contents (g kg^{-1} dry weight) of K, Ca and Mg and the ash content (g kg^{-1} dry weight) of leaves of cucumber (A), tomato (B), sweet pepper (C) and eggplant (D). Each data point was from one sample.

Leaves

The carbon contents of leaves in all three species decreased markedly with age (Table 3.2). It was quite low in the older leaves. The most important cause for this decrease in C-content appeared to be the increase in mineral content. In leaves of cucumber, sweet pepper and egg-plant the increase in ash content with leaf age was largely due to the increase in Ca-content (Fig. 3.1). In tomato leaves increases in Ca and K-content were less closely correlated with the increase in ash content.

The mineral content as estimated from the ash content, *m1*, differed in a number of cases significantly from the mineral content estimated from the total content of K, Ca, Mg and NO_3 (*m2*). In middle and old aged leaves of cucumber and tomato the difference between *m1* and

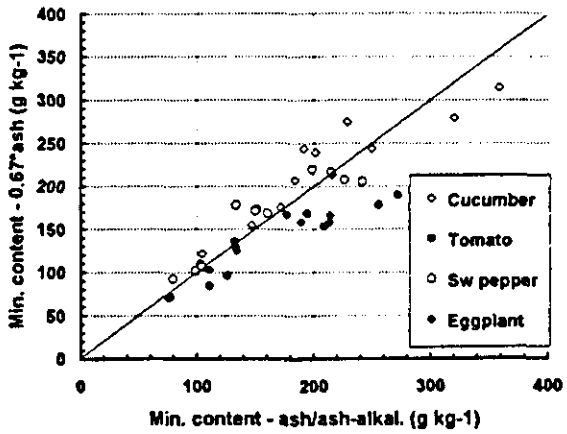


Figure 3.2
The relation between mineral content (g kg^{-1}) calculated from ash content, ash alkalinity and NO_3-content (*m1*) and mineral content estimated as 0.67*ash (*m3*), in leaves of cucumber, tomato, sweet pepper and eggplant. Each data point was from one sample

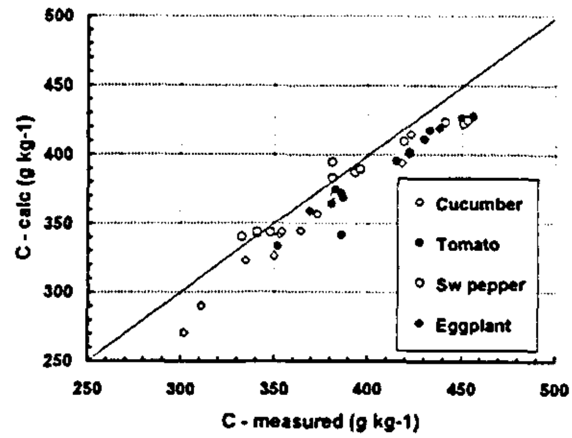


Figure 3.3
The relation between C-content of dry matter (g kg^{-1}) calculated from the chemical composition of leaves of proteins, carbohydrates, lipids, lignin, organic acids and the measured C-content of dry matter (g kg^{-1}). Each data point was from one sample.

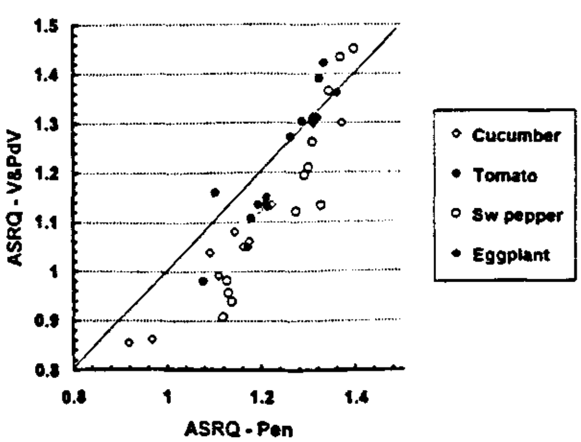


Figure 3.4
The relation between the assimilate requirement of dry matter of leaves (g CH_2O g^{-1} DM) calculated according to the method of Penning de Vries *et al.* (1974) ('ASRQ-Pen') and the assimilate requirement calculated according to Vertregt & Penning de Vries (1987) ('ASRQ-V&PdV'). Each data point was from one sample.

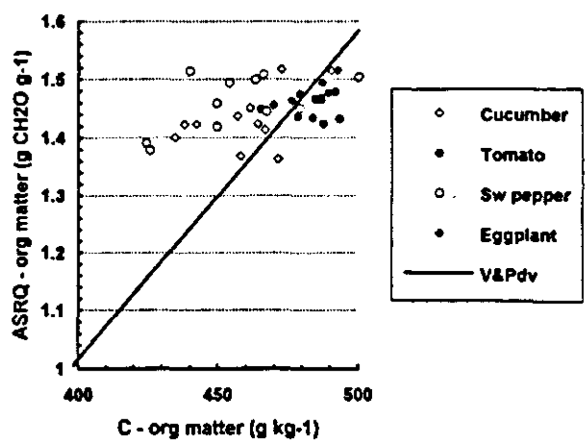


Figure 3.5
The relation between $ASRQ_{om}$ (g CH_2O g^{-1}) and C-content of organic matter of leaves (g kg^{-1}) as calculated according to method PdV and compared with the predicted $ASRQ_{om}$ based on method V&PdV (solid line). Each data point was from one sample.

m2 was 6 to 8 % of the dry weight. Mineral content estimated as 0.67 * ash, *m3*, was on average about equal to *m1*, but could differ siginificantly for the higher ash contents (Fig. 3.2).

The C-content of dry matter as calculated from the C-content of the proteins, carbohydrate, etc., was about 20 g kg^{-1} dry matter less than the measured C-content (Fig. 3.3), i.e a small difference.

The ASRQ-values calculated by method PdV were about 1.35-1.40 for young leaves (Table 3.2). The oldest leaves had ASRQ-values of about 0.95 in cucumber, 1.05 in tomato, and 1.10 in sweet pepper. Proteins and carbohydrates had the largest contributions in the costs. Costs of organic acids were important in older leaves of cucumber and sweet pepper, i.e. 0.2 - 0.3 g glucose per g DM (not shown). Note that in tomato older leaves are pruned regularly. Although minerals accumulated to considerable amounts, their costs were small, i.e. less than 0.03 g glucose per g DM.

The assimilate requirements as calculated by method V&PdV differed for many of the leaf samples more than 5 % from the ASRQ-values calculated by method PdV (Fig. 3.4). Notably at high mineral contents the difference between the two methods became large. This seemed to be largely caused by the fact that $ASRQ_{om}$ calculated by Eqn 3.5 was for low C_{om} values higher than the value predicted by V&PdV (Eqn 3.7) (Fig. 3.5).

Stems

Most of stem material contained significant amounts of ash, i.e. more than 10 % of the dry weight (Table 3.3). Cucumber stems contained very high amounts of NO_3^- (about 13 % of the dry weight), this was associated with low C-contents. In cucumber stems *m1* was calculated to be higher than the ash content, which is obviously not possible.
m3 was significantly smaller than *m1* (Fig. 3.6). The difference increased with increase in ash content and NO_3-content.

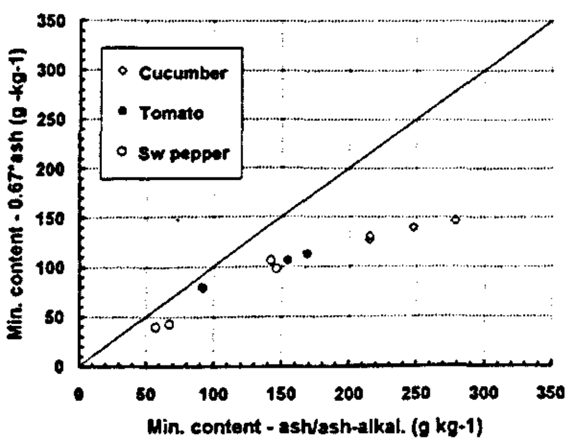


Figure 3.6
The relation between mineral content (g kg^{-1}) calculated from ash content, ash alkalinity and NO_3- content (*m1*) and mineral content estimated as 0.67*ash (*m3*), in stems of cucumber, tomato and sweet pepper. Each data point was from one sample.

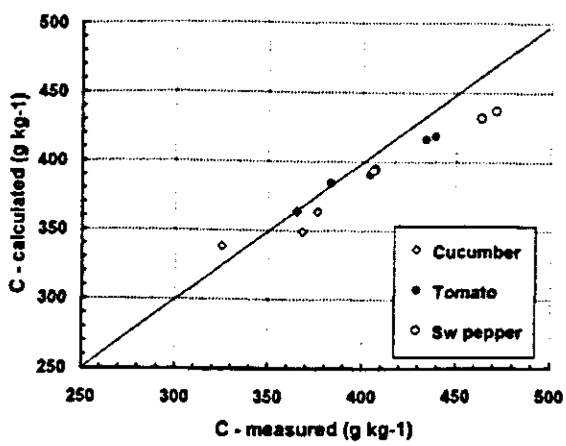


Figure 3.7
The relation between C-content of dry matter (g kg^{-1}) calculated from the chemical composition of stems of proteins, carbohydrates, lipids, lignin, organic acids and the measured C-content of dry matter (g kg^{-1}). Each data point was from one sample.

The C-content calculated from the protein content, etc, differed little from the measured C-content (Fig. 3.7).

The ASRQ-values calculated according to Penning de Vries *et al.* (1974) were rather low, i.e. mostly in the range 1.05 - 1.20 (Table 3.3). The young parts of tomato stems, the oldest parts of sweet pepper stems and eggplant stems had ASRQ-values above 1.20, mainly because of low mineral content and/or high lignin content.

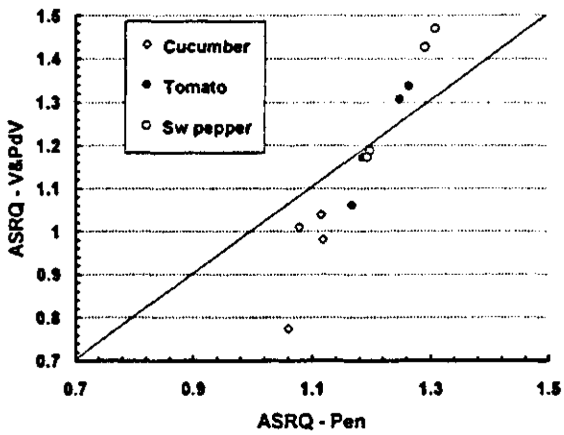


Figure 3.8
The relation between the assimilate requirement of dry matter of stems (g CH_2O g^{-1} DM) calculated according to the method of Penning de Vries et *al.* (1974) ('ASRQ-Pen') and the assimilate requirement calculated according to Vertregt & Penning de Vries (1987) ('ASRQ-V&PdV'). Each data point was from one sample.

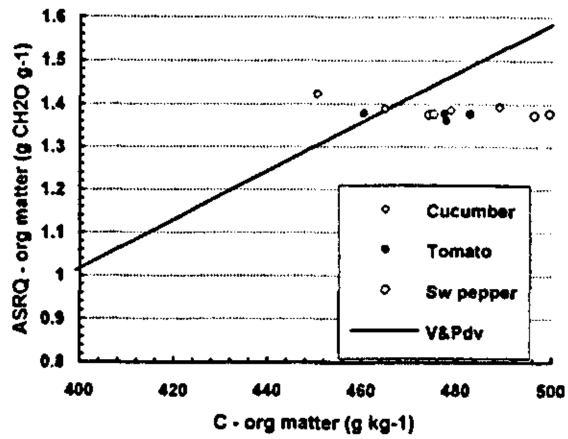


Figure 3.9
The relation between $ASRQ_{om}$ (g CH_2O g^{-1}) and C-content of organic matter of stems (g kg^{-1}) as calculated according to method PdV and compared with the predicted $ASRQ_{om}$ based on method V&PdV (solid line). Each data point was from one sample.

The assimilate requirements as calculated by method V&PdV differed for most of the samples significantly from the values calculated according to method PdV (Fig. 3.8). The lowest ASRQ-values estimated by method V&PdV for cucumber stems seem unlikely.
The $ASRQ_{om}$ values based on method PdV were about constant at 1.35 - 1.40 (Fig. 3.9).

Fruits

Fruits of cucumber, tomato and sweet pepper had a typical chemical composition, which made it characteristic for that species. Cucumber fruits had relatively higher protein and lower lipid and lignin content than the other species, tomato fruits had relative high ash and low protein and NO_3-content, and sweet pepper fruits had a high lipid content and a low ash content (Table 3.4).

The difference between *m1* and *m2* in tomato fruits was 7 to 17 %. *m3* underestimated mineral content *m1* with 19-24 % (Fig. 3.10).
The calculated C-content was significantly lower than the measured C-content (Fig. 3.11).

The ASRQ-values calculated according to method PdV were rather low for the tomato fruits sampled in spring, due to their high mineral content (first of the two entries in Table 3.4). The ASRQ-value of the sweet pepper fruit sample that was completely analysed was 1.35 g CH_2O g^{-1} DM.

The ASRQ-values calculated according to method V&PdV were considerably larger than the values calculated according to method PdV (Fig. 3.12), as $ASRQ_{om}$ based on method PdV was on average about 1.45 and was quite lower than $ASRQ_{om}$ based on method V&PdV (Fig. 3.13).

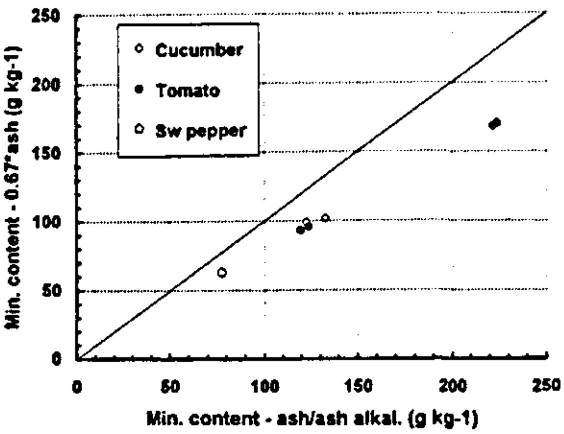


Figure 3.10
The relation between mineral content (g kg^{-1}) calculated from ash content, ash alkalinity and NO_3-content (*m1*) and mineral content estimated as 0.67*ash (*m3*), in fruits of cucumber, tomato and sweet pepper. Each data point was from one sample.

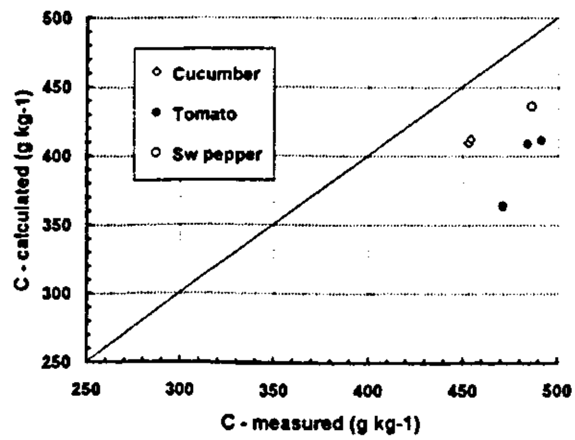


Figure 3.11
The relation between C-content of dry matter (g kg^{-1}) calculated from the chemical composition of fruits of proteins, carbohydrates, lipids, lignin, organic acids and the measured C-content of dry matter (g kg^{-1}). Each data point was from one sample.

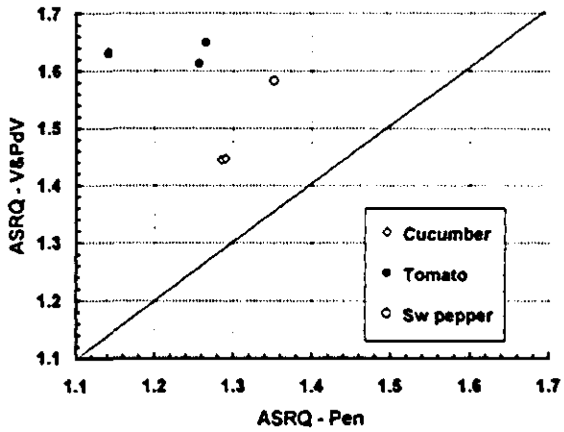


Figure 3.12
The relation between the assimilate requirement of dry matter of fruits (g CH_2O g^{-1} DM) calculated according to the method of Penning de Vries *et al.* (1974) ('ASRQ-Pen') and the assimilate requirement calculated according to Vertregt & Penning de Vries (1987) ('ASRQ-V&PdV'). Each data point was from one sample.

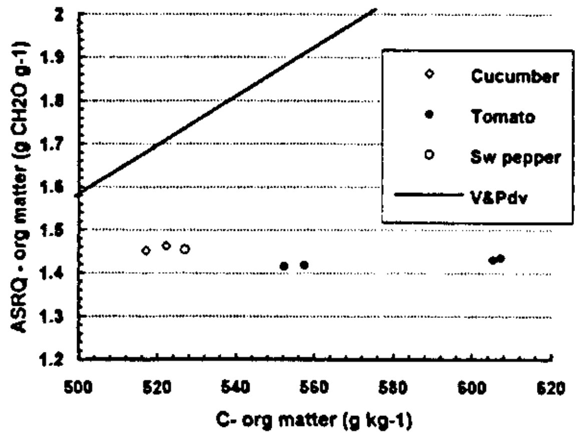


Figure 3.13
The relation between $ASRQ_{om}$ (g CH_2O g^{-1}) and C-content of organic matter of fruits (g kg^{-1}) as calculated according to method PdV and compared with the predicted $ASRQ_{om}$ based on method V&PdV (solid line). Each data point was from one sample.

3.2.6. Discussion

All crops appeared to accumulate siginificant amounts of minerals in the leaves. This phenomenon is commonly found in species that reduce NO_3^- in the shoot and compensate the charge of OH^- that is liberated in the reduction process by uptake of cations (e.g. as found in tomato by Kirkby and Mengel (1967)). The gradual decline with leaf age of the assimilate requirement makes that a single ASRQ-value for leaves will be somewhat of an approximation.

The two methods for estimation of the mineral content *m1* and *m2* (sum of Ca, K, Mg and NO_3) appeared to yield rather significant different values for a number of samples. Perhaps, some of the discrepancy could be due to one or several of the other minerals not analysed being present in significant amounts in the dry matter, e.g. Na^+, Cl^-, SO_4^{2-}, $H_2PO_4^-$, SiO_2. Concentrations of SO_4^{2-} and $H_2PO_4^-$ in tomato leaves have been reported to be in the range 1-2 % (Kirkby & Mengel, 1967; Kirkby & Knight, 1977). Cucumber and tomato are known to be able to accumulate SiO_2 (Marschner, 1986). Miyake & Takahashi (1983) found that cucumber supplied with 0.83 mmol SiO_2 contained about 2 % SiO_2 in the leaf dry weight. Here, the Si-concentration in cucumber nutrient solutions was about 0.5 mmol l^{-1}; possibly it could reach a significant fraction of the dry weight in the older leaves of cucumber. No Si was added to the nutrient solutions of the tomato crops, so these will presumably contain only the trace amounts that are normally present.

The value of *m1* being larger than the ash content in cucumber stems could perhaps be caused by measurement errors. The high NO_3 content makes the dried and ground stem material somewhat 'explosive', so that loss of material from the ashing scales could have occurred when heating to 550 °C for determination of the ash content.

Differences between method PdV and V&PdV were often more than 5 % in older leaves and more than 10 % in fruits. The exact causes for the discrepancies were not sure. Applying the method V&PdV and using *m1* (based on ash and ash alkalinity) instead of *m3* (0.67 * ash) changed little the differences between the two methods. Thus the differences seem to mainly ly in the estimation of $ASRQ_{om}$.

In case of the fruits the discrepancy could be partly explained when it is assumed that a major part of the 'carbohydrates' estimated to arrive at 100 % material is not purely carbohydrate but are compounds that contain more C than carbohydrate itself. Notably volatile aromatic compounds have a high C-content. The 'missing' C was 10-15 % as indicated from comparison of measured and calculated C-content.

The tomato fruit used by V&PdV in determining the regression line had an ASRQ-value of 1.424, i.e. significantly higher than the values of 1.15-1.25 as found here. The fruit of V&PdV had a higher protein content (17 % versus 12 % here) and an exceptionally high lignin content, i.e. 9 %. This latter value is higher than the lignin contents estimated for the stems of the crops considered here, and contributed significantly to total costs. From data on the composition of a tomato fruit from Davies & Hobson (1981 in Grierson & Kader, 1986) an ASRQ-value of 1.25 was calculated.

Vertregt & Penning de Vries (1987) considered their method at least as accurate as the method of Penning de Vries *et al.* (1974) as the elaborate chemical analysis of the latter would be more liable to errors. Perhaps the fact that C-content calculated from chemical composition differed significantly from C-content determined by gaschromatography could be an indication of this. However, the restriction of method V&PdV to leaf material with less than 13 % minerals and to storage material with less than 6 % minerals makes this method less applicable to the majority of plant material of the crops considered here.

It is concluded that the ASRQ-values of the greenhouse crops investigated here must be based on more elaborate chemical analysis than on only C- and ash content.

3.3. Validation

3.3.1. Model description

Crop dry matter production was simulated with a model for photosynthesis and dry matter production in Venlo-type glasshouses described elsewhere (Gijzen, in prep.). In this model crop photosynthesis was calculated taking account of the row structure of the crops. Leaf photosynthesis was calculated based on the model of Farquhar *et al.* (1980). Dry matter production was calculated from the daily rate of canopy gross photosynthesis, following the model SUCROS87 (Spitters *et al.*, 1989).

3.3.2. Experiments

The simulation model was validated with 3 experiments on cucumber, 1 on tomato, and 2 on sweet pepper. Experiments were performed in glasshouses at the Glasshouse Crops Research Station (Vegter, 1989 and Rijsdijk et al, 1989) (Table 3.5). Crops grown in different compartments had the same treatments, except for the cucumber 1988 autumn experiment. In this experiment different CO_2 treatments were applied; in the simulation runs were used the treatments in which the CO_2 concentration setpoints were at 350 or at 700 µmol mol^{-1} (6 compartments in total).

Table 3.5 Some characteristics of the experiments used for validation of the simulation model of dry matter production.

No.	Experiment	Start	End	No. of comp.	Reference
1	Cucumber - autumn	20-08-1987	22-10-1987	4	Vegter (1989)
2	Cucumber - spring	21-12-1987	16-05-1988	2	"
3	Cucumber - autumn	03-08-1988	27-10-1988	6	Rijsdijk *et al.* (1989)
4	Sweet pepper - year round	18-12-1987	02-11-1988	1	Vegter (1989)
5	Sweet pepper - autumn	13-07-1988	28-11-1988	2	"
6	Tomato - spring	30-12-1988	08-08-1989	3	"

In the growth experiments plants were harvested at specific times, and dry weights determined of leaves, stems and fruits on the plants. Dry weights were generally determined by oven drying at 70-80 °C. However, when drying was done at 105 °C dry matter contents were found to be lower in fruits and stems (Marcelis, CABO-DLO, pers. comm., 1990, Rijsdijk *et al.*, 1992). As ASRQ-values were determined on material that had been dried at 105 °C, measured dry weights of stems and fruits on the plants were corrected by multiplication by 0.98 and 0.91, respectively. Dry weights of harvested fruits were, except for one time, not measured. From data of Houter (1991), Rijsdijk *et al.* (1992), De Koning (1993) and Rijsdijk *et al.* (1993) it appeared that dry matter content of harvested fruits of cucumber and tomato can vary by about 10-15 % depending on, among others, time of season or grower.

Based on these data the dry matter contents of fruits of cucumber and tomato were assumed to follow a sinusoidal course during the year. I.e. for cucumber

$$\%\text{-}age\ DM = 2.7 + 0.4 * (sin(day - 80)/180) \tag{3.8}$$

and for tomato

$$\%\text{-}age\ DM = 5.4 + 0.6 * (sin(day - 80)/180) \tag{3.9}$$

where day is day number of the year. In sweet pepper a seasonal pattern was less discernible; here the dry matter content of harvested fruits (in the red stage) was taken to be 8 %.

3.3.3. Model input of climate variables

Half hour averages of measured global radiation outside the greenhouse, and CO_2 concentration and temperature inside the greenhouse were input to the model.

3.3.4. Model parameterization

The values of the parameters for leaf photosynthesis were estimated from validation of the crop photosynthesis model with measured canopy photosynthesis (Gijzen, Nederhoff and Vegter, in prep.). I.e. the maximal rate of carboxylation ($V_{cmax,}$ at 25 °C) was set at 150 µmol m^{-2} s^{-1}, the maximal rate of electron transport ($J_{max,}$ at 25 °C) was set at 300 µmol m^{-2} s^{-1}. Leaf Area Indices, row widths and heights as measured in the experiment were input to the model. Maintenance respiration was calculated based on measured crop dry weights, following SUCROS87.

The assimilate requirements of plant parts were calculated based on chemical analysis of material sampled in other experiments (Table 3.6). The ASRQ-value of the roots was taken to be 1.45, based on Spitters *et al.* (1989).

Table 3.6 Assimilate requirements (g CH_2O per g dry matter) of plants parts, as calculated from chemical analysis (see Chapter 3.2).

	Cucumber	Tomato	Sweet pepper
Leaves	1.20	1.25	1.30
Stems	1.10	1.20	1.25
Fruits	1.30	1.20	1.35

Dry matter partitioning to plant parts (leaves, stems, roots and fruits) as measured in the experiments was input to the model.

3.3.5. Simulation results

Cucumber

Dry matter production of cucumber was simulated reasonably well (Figs. 3.14 A, B, C, D). The rate of dry matter production was generally overestimated by about 5 - 15 %, but in some cases this was siginificantly higher. Notably in the autumn '88 experiment this was the case both in the beginning of the growth period and at the end.

Sweet pepper

Dry matter production of the year round sweet pepper crop was simulated quite well (Fig. 3.14E). However, in the last 40 days of this experiment no increase of dry matter production was simulated, as opposed to the measurements. According to the simulations all assimilates were in this period consumed by maintenance respiration. The fruit dry matter production of the autumn crop was underestimated significantly (Fig. 3.14F).

Tomato

Dry matter production of tomato was overestimated by 10 % (Fig. 3.14G). It was mostly overestimated in the beginning of this experiment.

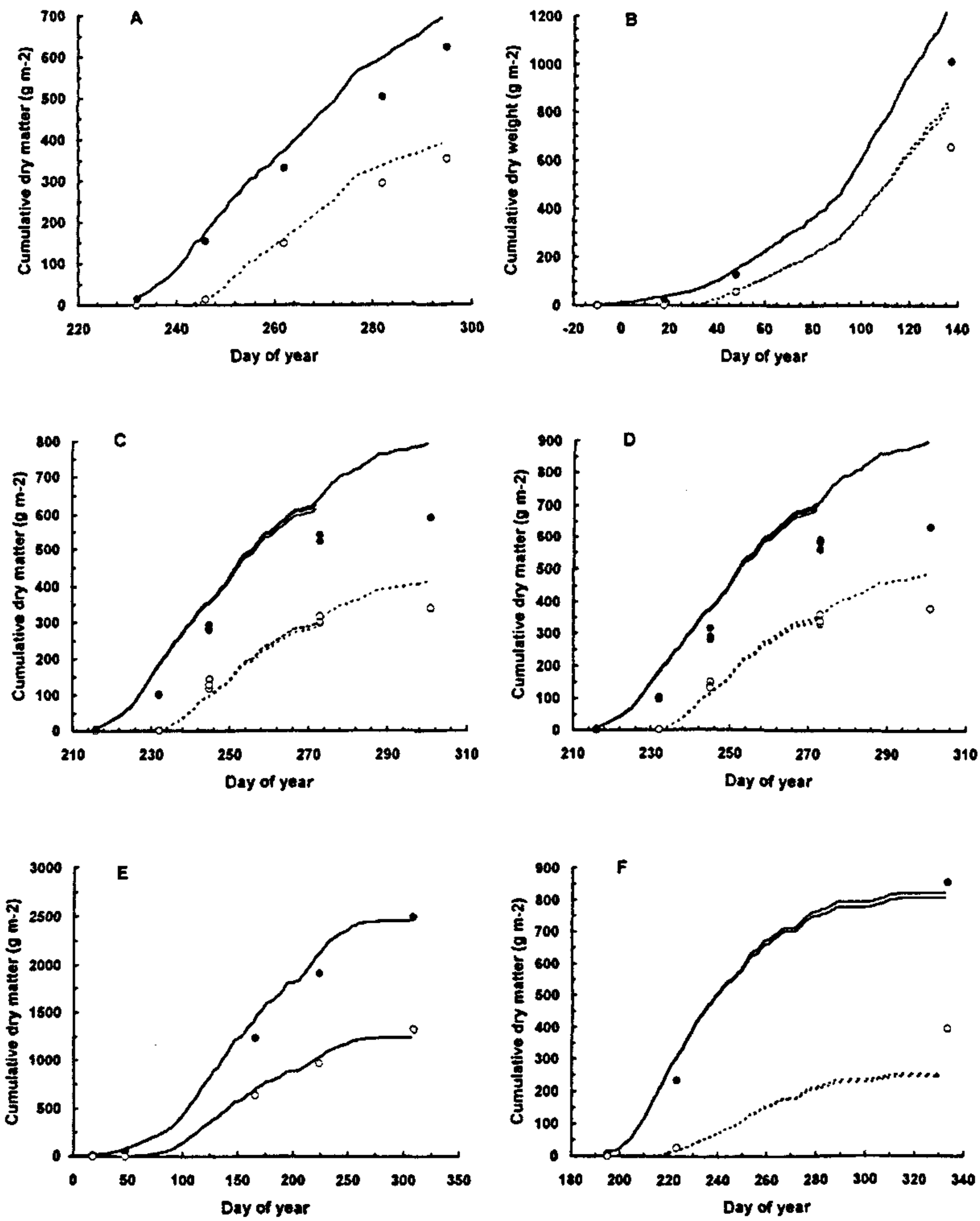


Figure 3.14 Measured and simulated dry matter production of cucumber, sweet pepper and tomato. Measurements: black dots: cumulative total dry matter production, open dots: cumulative fruit dry matter production. Simulations: solid line: cumulative total dry matter production; dashed line: cumulative fruit dry matter production. Multiple lines and dots indicate productions in two or more compartments.

A. Cucumber autumn 1987. B. Cucumber spring 1988. C. Cucumber autumn 1988 340 ppm treatment. D. Cucumber autumn 1988 - 700 ppm treatment. E. Sweet pepper 1988. F. Sweet pepper autumn 1988.

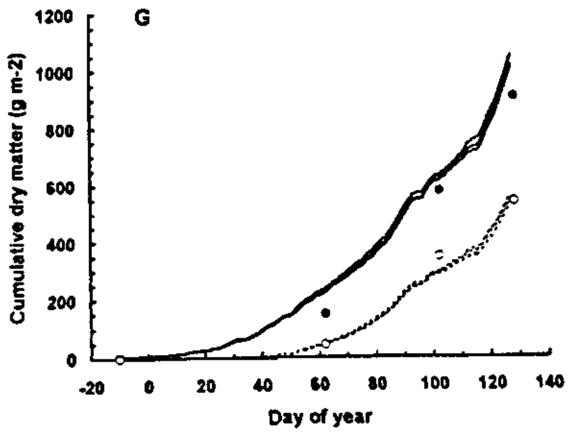


Figure 3.14 Continued. G. Tomato spring 1989.

3.3.6. Discussion

The simulation results were on average fairly good. Generally dry matter production was overestimated. Perhaps, increased stomatal limitation of assimilation could have occurred in periods of high transpiration rates. In simulation this was not taken into account. A fixed high stomatal conductance of 0.02 m s^{-1} was assumed, not affected by environmental or plant conditions.

The cumulative production of the cucumber autumn '88 experiment was overestimated significantly. One possible cause for this could be the exceptionally low humidities occurring in the first month of this experiment (R.H's as low as 50 % were measured, Rijsdijk, 1989), from which the young crop could have suffered much. Another cause could be the high incidence of fungal diseases in the last month of this experiment.

The simulated fruit dry matter production depends partly on the course in time of dry matter partitioning that was presumed. A linear increase in the fraction of DM partitioned to the fruits was assumed with the onset of fruit production, until the 'steady state' value of the production stage was reached. This transition period was taken somewhat less than the length of period from first flowering to first harvest. In the case of the sweet pepper autumn '88 experiment any assumption had significant effect on simulated fruit production at the end of the experiment, and thus could have contributed to the underestimation.

3.4. References

Dijkshoorn, W., 1973.
Organic acids, and their role in ion uptake. In: G.W. Butler & R.W Bailey. (eds). Chemistry and biochemistry of herbage. 163-188.

Grierson, D. & A.A. Kader, 1986.
Fruit ripening and quality. In: J.G. Atherton & J. Rudich (eds). The tomato crop. Chapman and Hall, London. 241-280.

Houter, B., 1991.
ECP-model: simulation model for energy consumption, CO_2 consumption and production in glasshouses. Final Report. Glasshouse Crops Research Station, Naaldwijk, Dec. 1991. (in Dutch).
Kirkby, E.A. & A.H. Knight, 1977.
Influence of the level of nitrate nutrition on ion uptake and assimilation, organic acid accumulation, and cation-anion balance in whole tomato plants. Plant Physiol. 60: 349-353.
Kirkby, E.A. & K. Mengel, 1967.
Ionic balance in different tissues of the tomato plant in relation to nitrate, urea or ammonium nutrition. Plant Physiol. 42: 6-14.
Koning, A. de, 1993.
Growth of a tomato crop. Measurements for model validation. Acta Hort. 328: 141-146.
Marschner, H., 1986.
Mineral nutrition of higher plants. Academic press Ltd, London. 674 pp.
Miyake, Y & E. Takahashi, 1983.
Effect of silicon on the growth of solution-cultured cucumber plant. Soil Sci. Plant Nutr. 29: 71-83.
Penning de Vries, F.W.T, A.H.M. Brunsting & H.H. van Laar, 1974.
Products, requirements and efficiency of biosynthesis: a quantitative approach. J. theor. Biol. 45: 339-377.
Poorter, H., 1991.
Interspecific variation in the relative growth rate of plants. Ph.D. Diss., Univ. Utrecht, 102 pp.
Rijsdijk, T., E. Nederhoff & F. Stelder, 1989.
Optimization of CO_2 enrichment. Comparison optimization program with various strategies used in practice. Internal Report nr. 4, Jan. 1989. Glasshouse Crops Research Station, Naaldwijk. 49 pp. (in Dutch).
Rijsdijk, T., R. Verbeek & E. de Ruiter, 1992.
Comparison optimization program with various practical measures. Internal Report nr. 55, April 1992. Glasshouse Crops Research Station, Naaldwijk, 52 pp. (in Dutch).
Rijsdijk, T., H.W. de Ruiter & G. Bergman, 1993.
Crop data collected at commercial growers. Season 1991-1992. Internal Report nr. 5, Feb. 1993. Glasshouse Crops Research Station, Naaldwijk, 71 pp. (in Dutch).
Sonneveld, C. & A. van der Wees, 1988.
Nutrient solutions for growth on rockwool in the Westland en De Kring. Tomato, cucumber, sweet pepper and eggplant. Series: Nutrient solutions glasshouse horticulture, nr. 1. Glasshouse Crops Research Station, Naaldwijk. (in Dutch).
Spitters, C.J.T., H. van Keulen & D.W.G. van Kraalingen, 1989.
A simple and universal crop growth simulator: SUCROS87. In: R. Rabbinge, S.A. Ward and H.H. van Laar (eds). Simulation and system management in crop production. PUDOC, Wageningen. 147-181.
Vegter, J., 1989.
Measurements of crop photosynthesis 1987-1989. Report of growth experiments and of experimental set-up. Internal Report nrs. 26 & 27. Glasshouse Crops Research Station, Naaldwijk. (In Dutch).
Vertregt, N. & F.W.T. Penning de Vries, 1987.
A rapid method for determining the efficiency of biosynthesis of plant biomass. J. Theor. Biol. 128: 109-119.

4. Simulation of transpiration

Summary

A multilayer model and a big-leaf model for canopy transpiration are described. The multilayer model incorporated several submodels for stomatal response. A canopy transpiration module has been built to take account of the varying shade that transpiring plants could receive from neighbouring plants or from the greenhouse construction elements. It was simulated that measured transpiration rate could be significantly affected by the shading effects of neighbouring plants and by the greenhouse construction, especially when plants cover half the width of the row.

A sensitivity analysis was performed with the multilayer canopy transpiration model. The multilayer transpiration model was tested with parameters obtained from literature against measured canopy transpiration rates of tomato, sweet pepper and cucumber. Simulated transpiration did not agree well with measured transpiration.

The submodels for stomatal conductance have been parameterized by fitting simulated crop transpiration to measured crop transpiration. In general a good fit has been obtained. Their predictive power needs to be further tested. Recommendations are given about the use of the models.

4.1. Introduction

Many models have been developed of greenhouse crop transpiration. Most of them are simple linear regression models using global radiation and VPD of the greenhouse air as driving variables. In more elaborate models stomatal conductance is introduced as an additional variable (e.g. Stanghellini, 1987; Jolliet, 1993). Most of these models are so-called 'big-leaf' models.
A multilayer model of transpiration was developed by Marcelis (1989), which took into account the gradient of absorbed PAR and NIR within the canopy. In this model stomatal conductance, g_s, was dependent on absorbed PAR and the water content of the plant.

In present research three types of models were used to calculate canopy transpiration (E_{cr}):
- simple equations in which E_{cr} is related directly to environmental conditions,;
- a multilayer model: the canopy was modelled to consist of various leaf layers, and E_{cr} was obtained by summing transpiration of individual layers, and;
- a big-leaf model: the canopy is assumed to consist of a single leaf layer.

The multilayer model of Marcelis (1989) was adapted by leaving out the effect of the plant water content, and by replacing the submodel for g_s by other variants. These were: *a)* 2 submodels in which g_s was made dependent on the rate of leaf photosynthesis, and *b)* 2 so-called descriptive submodels in which g_s was not dependent on leaf photosynthesis but solely a function of absorbed PAR and VPD.

No modelling was done on the waterstatus of the plant and its effects on stomatal conductance, as too little data on these aspects were available.

Measured transpiration can have some variation caused by temporal variation in the amount of shading of the plants on the weighing scale by neighbouring plants or by glasshouse construction elements. An extension to the transpiration model was build to account for this variation, so that physiological responses could be discerned better.

Some stomatal responses published in the literature were used in the multilayer model and their performance was tested with datasets on measured canopy transpiration of tomato, sweet pepper and cucumber.

Finally, simulated crop transpiration was fitted to the measurements by tuning parameters of the stomatal conductance models.

4.2. Model description

4.2.1. The radiation climate inside the greenhouse

For all models and all but one experiment the shortwave radiation climate outside the greenhouse was calculated based on measured global radiation. From measured global radiation the atmospheric transmission was calculated. Then the fraction PAR in global radiation, the fraction diffuse in PAR, the diffuse and direct fluxes of PAR and UV were calculated as described in Chapter 2.

The short wave radiation climate inside was calculated based on the diffuse and direct radiation transmissivities of the glasshouse cover, using the model of Bot (1983).
Also the transmission of UV-radiation was taken account of. The transmission of glass for UV is lower than of PAR and NIR, and depends on the glass intrinsic properties. Based on some spectral transmission measurements (J.A. Stoffers, IMAG-DLO; F. Maas, AB-DLO) the glass transmissivity for UV was set at 67 % of that of global radiation as a whole. However, as the fraction UV in global radiation is only about 0.05, its contribution to the global radiation intensity in the greenhouse is low.

4.2.2. Simple equations for canopy transpiration

Relation based on global radiation and VPD

In this relation measured E_{cr} was described by

$$E_{cr} = a\, Q_g + b\, D_a \qquad (4.1)$$

where

Q_g	= global radiation inside the greenhouse	(J m^{-2} s^{-1})
D_a	= vapour pressure deficit of the air	(kPa)

Makkink-formula

In this formula E_{cr} is simply related to global radiation and indirectly to air temperature via the slope of the water vapour saturation curve

$$\lambda E_{cr} = c\frac{s}{s+\gamma}Q_g \qquad (4.2)$$

where

λ	= heat of vaporization of water	($J\ g^{-1}\ H_2O$)
s	= slope of water vapour saturation curve	($Pa\ °K^{-1}$)
γ	= adiabatic psychrometric constant	($Pa\ °K^{-1}$)
c	= is a constant	(-)

This formula has been succesfully applied in field crops.

4.2.3. Multilayer model

Crop transpiration was modelled based on the model as described by Marcelis (1989). In the model, the canopy was divided into several leaf layers, and transpiration of each layer was calculated from its energy balance. Crop transpiration was computed by summing transpiration of all this layers. Because the gradient in the canopy of absorbed PAR is calculated, this type of model enables stomatal conductance to be calculated based on leaf photosynthesis.

In the multilayer model the only vertical gradient of climatic factors in the canopy was that of absorbed radiation. No gradients of air velocity, air humidity, air temperature or CO_2 concentration were assumed. By default a horizontal homogeneous (closed) canopy was assumed. In certain cases account was taken of the effect of row structure and shading by the greenhouse cover.

Radiation penetration in canopy

Penetration and absorption of shortwave radiation were calculated as described by Spitters (1986). Both for PAR and NIR the same set of equations was used to calculate extinction and reflection. The difference between the extinction profiles of PAR and NIR arose from the different value of the scattering coefficient, σ. σ was for PAR assumed to be 0.15 (unpubl. results) and for NIR 0.8 (Monteith & Unsworth, 1990). Extinction of UV was treated the same as that of PAR.

The extinction of long wave radiation was calculated in the same manner. Here, the leaves were assumed to be black (σ at 0).

For the calculation of the extinction coefficients the so-called near-planophile leaf angle distribution was assumed (Gijzen, 1992). This leaf angle distribution is somewhat more horizontal than the spherical leaf angle distribution, and was considered characteristic for the crop species cucumber, tomato and sweet pepper.
Reflection by the ground surface was taken account of.

Leaf transpiration

Leaf transpiration was calculated with the Penman-Monteith equation.

$$\lambda\ E_l = \frac{sQ_n + D_a \rho c_p g_{b,v}}{s + \frac{g_{b,ht}}{g_{b,v} + g_l}\gamma} \qquad (4.3)$$

where

E_l	= leaf transpiration	(mg H_2O m^{-2} s^{-1})
Q_n	= absorbed net radiation	(J m^{-2} s^{-1})
g_l	= leaf conductance	(m s^{-1})
$g_{b,v}$	= boundary layer conductance for water vapour	(m s^{-1})
$g_{b,ht}$	= bound. layer cond. for heat (incl. thermal rad.)	(m s^{-1})
ρc_p	= volumetric heat capacity of air	(J m^{-3} °K^{-1})

The conductance for sensible heat ($g_{b,h}$, m s^{-1}) in the original PM-equation was replaced by a combined conductance for sensible heat and thermal radiation coming from above the canopy and from below the canopy. Conductances for heat and thermal radiation can be placed in parallel (Jones, 1983):

$$g_{b,ht} = g_{b,h} + g_{rad,top} + g_{rad,bot} \qquad (4.4)$$

where

$g_{b,h}$	= leaf conductance for heat (= $g_{b,v}$ / 0.93)	(m s^{-1})
$g_{rad,top}$	= leaf cond. for thermal radiation from top of canopy	(m s^{-1})
$g_{rad,bot}$	= leaf cond. for thermal radiation from bottom of canopy	(m s^{-1})

$g_{rad,bot}$ is used for calculating the conductance for thermal radiation coming from the ground surface and from heating pipes below the canopy.

$g_{rad,top}$ and $g_{rad,bot}$ are calculated from

$$g_{rad,top},g_{rad,bot} = \frac{K_{dif,bl}}{exp(-K_{dif,bl}\ L)} \qquad (4.5)$$

where

$K_{dif,bl}$	= the extinction coefficient of the canopy for black leaves	(-)
L	= the partial Leaf Area Index	(-)

The partial Leaf Area Index must be reckoned from the top and from the bottom of the canopy for $g_{rad,top}$ and $g_{rad,bot}$, respectively.

Absorbed net radiation consisted of short wave radiation and long wave thermal radiation

$$Q_n = Q_{p,abs} + Q_{p,abs} + Q_{t,cc} + Q_{t,pc} + Q_{t,gc} \quad (4.6)$$

where

$Q_{p,abs}$	= absorbed PAR	($J\ m^{-2}\ s^{-1}$)
$Q_{n,abs}$	= absorbed NIR	($J\ m^{-2}\ s^{-1}$)
$Q_{t,cc}$	= thermal rad. exchange between leaf and greenhouse cover	($J\ m^{-2}\ s^{-1}$)
$Q_{t,pc}$	= thermal rad. exchange between leaf and pipes	($J\ m^{-2}\ s^{-1}$)
$Q_{t,gc}$	= thermal rad. exchange between leaf and ground	($J\ m^{-2}\ s^{-1}$)

Also some UV-radiation is received by leaves. Here, the contribution of UV to transpiration was neglected as also no account was taken of the fact that part of the energy of absorbed PAR is required for metabolic processes. Both this energy and the energy contained in UV were considered to cancel each other out.

Boundary layer conductance

In greenhouses, air velocities are low, so that relatively large boundary layers develop around leaves. Little data are available on the magnitude of the boundary layer conductance. $g_{b,v}$ was measured to be about 0.01 m s^{-1} by Stanghellini (1985), inside a tomato canopy using replica leaves of 5 cm width, and it was estimated to be 0.005-0.01 m s^{-1} for *Ficus benjamina* having a leaf width of 5 cm (Zhang & Lemeur, 1992). Here, $g_{b,v}$ was set to 0.01 m s^{-1}.

Leaf conductance

Leaf conductance was calculated as the sum of parallel conductances of stomata and cuticula

$$g_l = g_s + g_{cut} \quad (4.7)$$

where

g_s	= stomatal conductance	($m\ s^{-1}$)
g_{cut}	= cuticular conductance	($m\ s^{-1}$)

In several greenhouse crops it was found that leaf conductance in the dark responded to leaf-air vapour pressure deficit (leaf-air VPD, D_l, kPa) (Bakker, 1991). Leaf conductance in the dark was calculated using a negative-exponential function, as described by Bakker (1991).

$$g_{ld} = g_{md} \exp(-a_d D_l) \quad (4.8)$$

where

g_{ld}	= leaf conductance in the dark	($m\ s^{-1}$)
g_{md}	= maximal leaf conductance at night	($m\ s^{-1}$)
a_d	= parameter	(kPa^{-1})

The average value of parameter a_d found for cucumber, tomato, sweet pepper and eggplant was about 1.2 (Bakker, 1991). This value was used here in the simulations. The value of parameter g_{md} was estimated from fitting (by eye) simulated crop transpiration to measured crop night transpiration.

g_{cut} was set at 0.0002 m s^{-1}. At low light intensities g_s could decrease to values lower than the difference of $g_{ld} - g_{cut}$. In those cases the lower limit of g_s was set by $g_{ld} - g_{cut}$.

$$g_s = \max\{g_s, g_{ld} - g_{cut}\} \qquad (4.9)$$

Stomatal conductance

With several models of stomatal conductance it was tested whether measured crop transpiration could satisfactorily be approached after parameter tuning. Two types of models were tested: 1) photosynthesis based models, in which stomatal conductance is related to the rate of photosynthesis, and 2) models in which stomatal conductance is calculated from ambient conditions.

Photosynthesis based models

1) Model of Ball et al. (1987)

Stomatal conductance as based on the model of Ball *et al.* (1987), was calculated to be dependent on leaf net photosynthesis,

$$g_s = 0.025m\frac{P_n h_s}{C_s - \Gamma} - b \qquad (4.10)$$

where

P_n	= leaf net photosynthesis		(μmol CO_2 m^{-2} s^{-1})
h_s	= relative humidity at the leaf surface		(as a ratio)
C_s	= CO_2 concentration at the leaf surface		(μmol mol^{-1})
Γ	= CO_2 compensation point		(μmol mol^{-1})
m	= parameter		(-)
b	= parameter		(m s^{-1})
0.025	converts mol m^{-2} s^{-1} to m s^{-1}		

In the model of Ball *et al.* the denomitor consisted only of the term C_s; here the CO_2 compensation point was introduced, following Leuning (1990). This author found a slightly better fit to data of Eucalyptus-leaves when this was done.

Several values of the parameters m and b have been published. m was found to be about 6 in a number of C3 species (Ball, 1988, cited by Collatz *et al.*, 1992), to vary from 7 to 11 in leaves of Eucalyptus grandis (Leuning, 1990), to be 9 in cotton leaves (Harley *et al.*, 1992) and to be about 10-11 in soybean (Harley & Tenhunen, 1991). b varied from 0.01 (Leuning, 1990) to 0.08 mol m^{-2} s^{-1} (Harley *et al.*, 1992).
The occurrence of relative humidity in the numerator has been questioned by Aphalo & Jarvis (1991). They found that g_s in ivy did not respond to relative humidity.

Iteration was needed to find P_n, g_s and the conditions at the leaf surface. Iteration was started with a certain value of C_i. Then P_n could be calculated, and from this C_s and g_s. The next value of C_i was calculated from

$$C_i = C_a - P_n \;(1/g_{s'} + 1/g_{b'}) \tag{4.11}$$

where the prime indicates unit mol m^{-2} s^{-1}. The Wegstein-S iteration method was used to find the equilibrium value of C_i.

2) Ratio C_i/C_s

Stomatal conductance often varies in parallel with leaf photosynthesis. As a consequence of this, the ratio of CO_2 concentration in the substomatal spaces, C_i, to the CO_2 concentration in the ambient air, C_a, appears to be rather constant at intermediate and high light intensities. Therefore, g_s was made a function of this ratio. Following Goudriaan (1989) this ratio was corrected for the CO_2 compensation point Γ. When g_s is based on C_a this would give

$$\frac{C_i - \Gamma}{C_a - \Gamma} = F_{cica} \tag{4.12}$$

where F_{cica} is the 'setpoint' at intermediate and high light intensities. As the boundary layer has been found to affect this ratio, the ratio of C_i to CO_2 concentration at the leaf surface, C_s, would for greenhouse crops probably be a better base for the calculation of g_s

$$\frac{C_i - \Gamma}{C_s - \Gamma} = F_{cics} \tag{4.13}$$

This latter ratio was used as input to the second stomatal conductance submodel. The stomatal conductance was found by an iterative procedure. In each round P_n was calculated for a given value of g_s. Then C_s was calculated by

$$C_s = C_a - \frac{P_n}{g_b / 1.37} \tag{4.14}$$

where g_b is the leaf boundary layer conductance in unit mol m^{-2} s^{-1}, and 1.37 converts the conductance for H_2O to conducance for CO_2. Substituting the new value of C_s in Eqn 4.13 yielded a new value of C_i. A new value of g_s was then calculated from the drop in CO_2 concentration from leaf surface to intercellular spaces

$$g_s = \frac{1.6\,P_n}{C_s - C_i} \tag{4.15}$$

where 1.6 converts the stomatal conductance to H_2O to conductance to CO_2. This equation is equal to

$$g_l = \frac{1.6\,P_n}{C_s - C_i} + g_{cut} \tag{4.16a}$$

(see Fig. 4.1). From measurements it has appeared that the relation between g_l and P_n is somewhat shifted (Goudriaan, 1989); in Fig. 4.1 to the left by a value equal to R_d.
Thus Eqn 4.16a becomes

$$g_l = \frac{1.6(P_n + R_d)}{C_s - C_i} + g_{cut} \qquad (4.16b)$$

From measurements on leaf conductance of cucumber, tomato and sweet pepper it was found that leaf conductance in the dark could be much higher that g_{cut} (Bakker, 1991). Therefore, conductance g_{cut} in Eqn 4.16b was replaced by a residual conductance g_{res}

$$g_l = \frac{1.6(P_n + R_d)}{C_s - C_i} + g_{res} \qquad (4.16c)$$

By subtracting g_{res} from g_l a new value of g_s was found. The Wegstein-S iteration method was used to find the equilibrium value of gs.

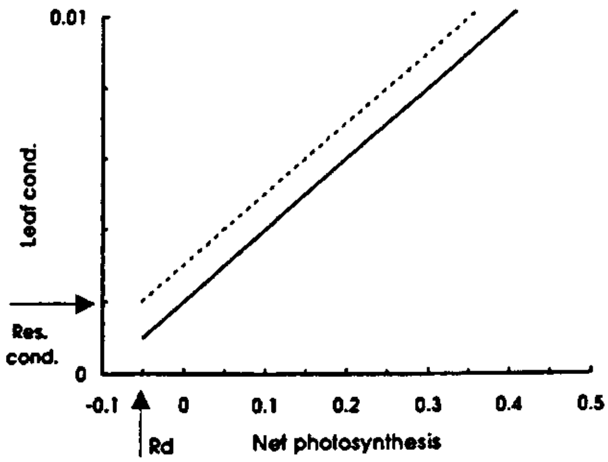


Figure 4.1 Schematized relation between leaf conductance and leaf net photosynthesis, when PAR is the varying factor. The dashed line indicates the relation shifted to the left by a value of P_n equal to R_d. Note that the slope of the relation will decrease for higher CO_2 concentrations (adapted from Goudriaan, 1989).

Morison & Gifford (1983) found the ratio of C_i to C_a to decrease approximately linearly with leaf-air VPD. Here a sensitivity to air humidity was introduced by making the internal setpoint dependent on the VPD at the leaf surface (D_s, kPa)

$$\frac{C_i - \Gamma}{C_s - \Gamma} = F_{cics} \exp(-fc1\, D_s) \qquad (4.17)$$

where $fc1$ is a parameter.

<u>Descriptive models</u>

These models were differentiated to the type of light response curve.

3) Negative-exponential model

Here, the response of stomatal conductance to absorbed PAR and leaf-air VPD was modelled following Bakker (1991) and Nederhoff & De Graaf (1993), whereas the response to CO_2 concentration of the greenhouse air was described according to Nederhoff & De Graaf (1993)

$$g_s = g_{smax}\ (1 - cd1\ \exp(-\ cd2\ PAR_{abs}))\ \exp(-cd3\ D_l)\ \exp(-cd4\ CO_2) \qquad (4.18)$$

where

g_{smax}	= maximal stomatal conductance	(m s^{-1})
$cd1$	= parameter	(-)
$cd2$	= parameter	(s m^2 J^{-1})
$cd3$	= parameter	(kPa^{-1})
$cd4$	= parameter	(μmol^{-1} mol)

Leaf-air VPD was found by iteration.

4) Linear model

Here, the response of stomatal conductance to light was assumed to be a Blackman-curve, i.e., to increase linearly with PAR, up to a ceiling, and to decrease exponentially with leaf-air VPD

$$g_s = \min\{g_{s1}, g_{smax}\}\ \exp(-\ cf2\ \ D_l) \qquad (4.19)$$

where

$$g_{s1} = cf1\ *\ PAR_{abs} \qquad (4.20)$$

and where g_{s2} is the ceiling, the conductance at saturating light intensity, and $cf1$ and $cf2$ are parameters. This model is very similar to the stomatal conductance model as described by Marcelis (1989).

Leaf photosynthesis

Leaf photosynthesis was modelled as described by Farquhar *et al.* (1980) and Kirschbaum & Farquhar (1984). With respect to modelling stomatal conductance, this means that a number of additional parameters were introduced, of which the most important were:
1) V_{cmax}, the maximal rate of carboxylation (µmol CO_2 m^{-2} s^{-1}), and
2) J_{max}, the maximal rate of electron transport (µmol e^- m^{-2} s^{-1}).

Parameters V_{cmax} and J_{max} were estimated from model tuning of a model of greenhouse crop photosynthesis with experimental data on net photosynthesis of whole crops of cucumber, tomato and sweet pepper (Gijzen *et al.*, in prep.). A value of V_{cmax} of 150 µmol CO_2 m^{-2} s^{-1} and a value of J_{max} of 300 µEq m^{-2} s^{-1} were found to give a good approximation of canopy photosynthesis of all three crops. These values are somewhat high in the range reported for several species.

4.2.4. Big-leaf model

In the big-leaf model the canopy is considered to consist of a single leaf layer. Transpiration of the canopy was calculated using the Penman-Monteith equation

$$\lambda\, E_{cr} = \frac{sQ_n + D_a \rho c_p\, g_{b,v,cr}}{s + \dfrac{g_{b,ht,cr}}{g_{b,v,cr} + g_{cr}}\gamma} \qquad (4.22)$$

where

Q_n	= absorbed net radiation	($J\ m^{-2}\ s^{-1}$)
g_{cr}	= canopy conductance	($m\ s^{-1}$)
$g_{b,v,cr}$	= aerodynamic + boundary layer conductance for water vapour	($m\ s^{-1}$)
$g_{b,ht,cr}$	= bound. layer cond. for heat (incl. thermal radiation)	($m\ s^{-1}$)

The big-leaf aerodynamic + boundary layer conductance for water vapour ($g_{b,v,cr}$) consists of the sum all leaf boundary layer conductances plus the aerodynamic conductance (i.e. the conductance of the air within the canopy outside the leaf boundary layers), and is difficult to estimate. It was, following Stanghellini (1987) and many other authors, calculated by placing in parallel the boundary layer conductances of all leaves. Thus

$$g_{b,v,cr} = g_{b,v}\ LAI \qquad (4.23)$$

$g_{b,ht,cr}$ was calculated as the sum of the conductances for sensible heat ($g_{b,h,cr}$) and the conductance for thermal radiation ($g_{rad,cr}$)

$$g_{b,h,cr} = g_{b,h}\, LAI \qquad (4.24)$$

$$g_{rad,cr} = \frac{200}{1 - \exp(-K_{dif,bl}\, LAI)} \qquad (4.25)$$

Canopy conductance (g_{cr}) was calculated using the descriptive negative-exponential model for stomatal conductance (see Eqn 4.18)

$$g_{cr} = g_{lm,av}\, LAI\ (1 - cn1\ \exp(-\ cn2\ PAR_{abs}))\ \exp(-cn3\ D_a)) \qquad (4.26)$$

where

$g_{lm,av}$	= maximal leaf conductance averaged over all leaf layers	($m\ s^{-1}$)
PAR_{abs}	= total PAR absorbed by the canopy	($J\ m^{-2}\ s^{-1}$)

4.2.5. Row and greenhouse cover effects

Row effect

The average absorption of direct radiation by plants in a small area of the total crop can be greatly different from that of the whole crop if it is positioned at one side of the row (Fig. 4.2A). Average absorption of plants in one half of the row will be larger than of the whole crop when their side of the row is directly exposed to the sun, and absorption is less when at another time of the day the plants are shaded by the other half of the row.

In two of the experiments reported here, plants were placed on lysimeters covering half of the row. As the effect of this particular placement in the row on measured diurnal transpiration seemed to be significant (R. de Graaf, PTG, pers. comm.), a special simulation routine was developed accounting for the differential distribution of absorbed direct PAR in plants on the lysimeter. In Appendix I the procedure followed is described.

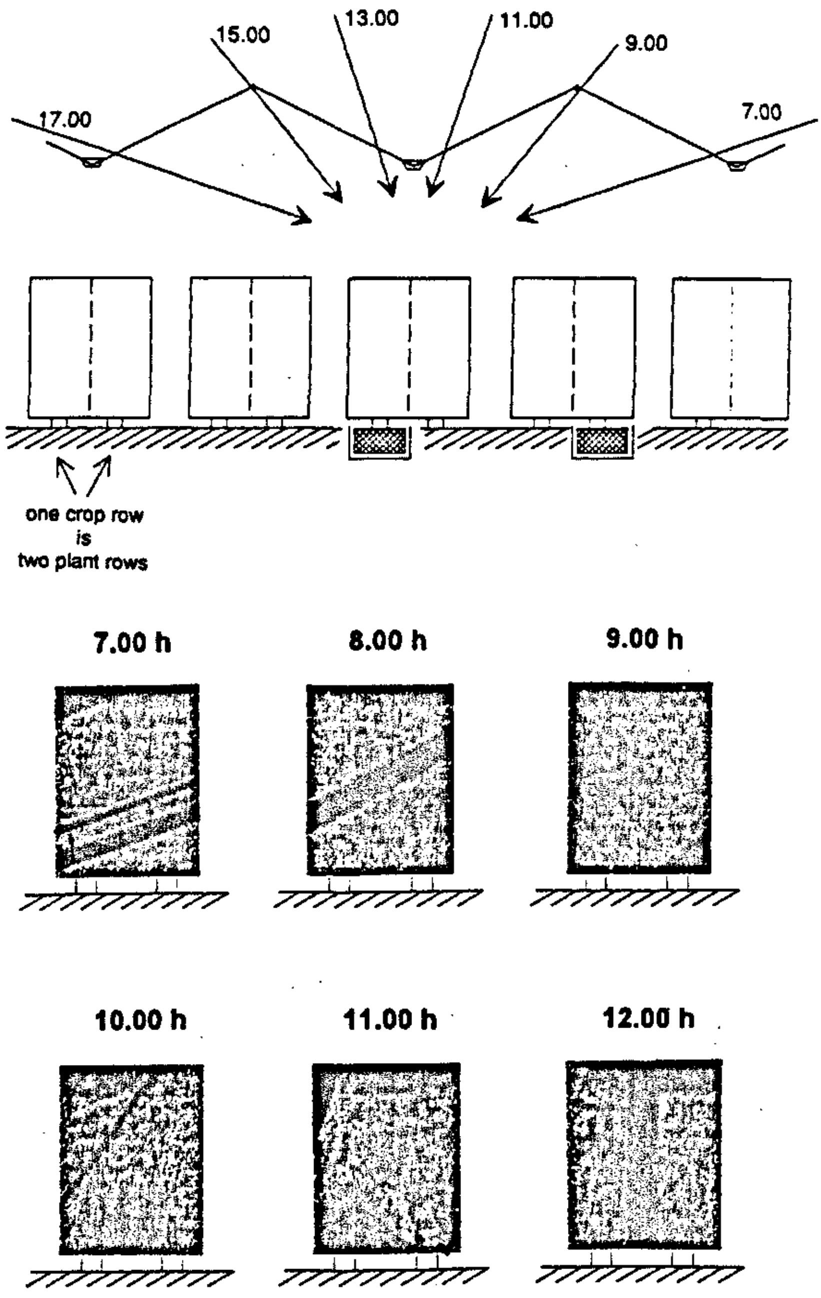


Figure 4.2 Schematization of the shading caused by the greenhouse cover or neighbouring plants. Both the row and the glasshouse are oriented north-south. Day number 115 (25 April). Row height 1.60 m, row width 1.15 m.

A. Depiction of a situation where transpiration is measured from plants at one half of the row. Arrow indicate directions of sun beam at various times (solar time) during the day). In the afternoon increased or decreased transpiration will be measured depending on whether their side is shaded by the neighbouring plants or directly exposed by the sun.

B. *Presentation of the shifting shades thrown by the gutter and ridge on a row crop.* Each block represents a row with rectangular transsection, with positions of shades of gutter (large bars) and gutters (thin bars) inside the row.

Effect of the greenhouse cover

Variability in transpiration rate between plants is also caused by the greenhouse cover. At sunny weather conditions the construction casts shadows on the canopy. In sunlit patches plant transpiration and photosynthesis will be higher than in shaded patches.
At the scale of a whole crop, photosynthesis was simulated to be significantly decreased when, at the same average light level, the crop was divided in a sunlit and a shaded part (with consequently different incident PAR intensities) (Gijzen ,1992). This decrease was due to the strongly non-linear photosynthesis-light response curve of leaves. It is expected that transpiration of a crop covering a relatively large area will be little decreased when taking account of unevenly distributed direct radiation intensity, as the response of leaf transpiration to absorbed radiation is almost linear.

When looking at a small scale, i.e. a few square meters, the scale of a lysimeter, diurnal transpiration will also vary because the plants in this small plot receive a varying amount of shade from the construction at different times of the day (Fig. 4.2). Measurements indicate that temporal variation in transpiration and variability in transpiration between plants, as caused by the greenhouse cover, is significant (R. de Graaf, PTG, pers. comm.).
To quantify the influence of the greenhouse cover on variability in transpiration, a model was developed that describes the spatial variability in direct radiation intensity as caused by the gutters and the ridges. In Appendix II this model is described.

4.3. Experiments

The transpiration models were tested with measurements of canopy transpiration of tomato, sweet pepper and cucumber. The data on sweet pepper and cucumber were collected by R. de Graaf at the PTG, in 1990, and the data on tomato by C. Stanghellini at the IMAG-DLO, in 1986 and in 1990 (Table 4.1). The measurements in 1986 were used by Marcelis (1989) for validation of his multilayer transpiration model.

Table 4.1 Some characteristics of the transpiration experiments on cucumber, sweet pepper and tomato, used for testing of the models

Crop	Experimental period	No. of compartments	No. of days	Comp.-days*
Tomato '86	April-May	1	9	9
Tomato '90	March - May	1	17	17
Sweet pepper	April - July	3	20	53
Cucumber	Sept - Oct.	3	10	30

* Comp.-days is number of compartment-days

In the cucumber, sweet pepper, and tomato '90 experiments, data were averaged to 10 minute records, in the tomato '86 experiment to 15 minute records. The number of records, average transpiration rates and average climatic conditions in the experiments are summarised in Table 4.2.

Table 4.2 Average climatic conditions (for $Q_g > 0.1$ J m^{-2} s^{-1}) and average instantaneous crop transpiration in the four experiments. Q_g outside greenhouse.

Crop	Nr. of records	Q_g (J m^{-2} s^{-1})	D_a (kPa)	Air temp. (°C)	CO_2 conc. (µmol mol^{-1})
Tomato '86	506	382.	0.58	22.2	-
Tomato '90	1373	310.	0.62	22.0	387.
Sweet pepper	5226	307.	0.83	25.6	556.
Cucumber	2118	203.	0.67	23.0	560.

Climate variables and parameters used as input to the models are summarised in Tables 4.3, 4.4 and 4.5.

4.3.1. Tomato '86

Climate variables

Global radiation was measured inside the greenhouse. To estimate the partitioning into diffuse and direct components and into PAR and NIR, outside global radiation was estimated by dividing the global radiation by 0.65, the estimated average greenhouse cover transmissivity. Then, the fluxes of PAR and NIR, and their separation into diffuse and direct components were calculated from global radiation as described in Chapter 2.

Table 4.3 Climate variables, greenhouse and crop parameters in the tomato '86 experiment, as used in the simulations

	Derivation / Value	Remark
Climate variables		
Global radiation	measured	measured inside greenhouse
Fraction diffuse in global radiation	calculated	as described in Ch. 2
PAR ($J\ m^{-2}\ s^{-1}$)	calculated	as described in Ch. 2
NIR ($J\ m^{-2}\ s^{-1}$)	calculated	as described in Ch. 2
Temperature inside greenh. (T_{in})	measured	
CO_2 concentration	not measured	
VPD greenhouse air	measured	
Roof temperature	measured	
Ground temperature	calculated	assumed at 20 °C
Greenhouse parameters		
Transmissivity	0.65	
Azimuth	East-West	
Ground reflectivity	0.25	assumed
Crop parameters		
Leaf Area Index	measured	
Row dimensions	not measured	
Row azimuth	East-West	

Greenhouse environment

Heating pipes were located above the canopy. Greenhouse cover temperature was measured. Ground temperature was assumed at 20 °C

Crop data

Plant density was 2.3 plants per m2. Leaf Area Index was at 2.

Transpiration measurements

4 plants were placed on the weighing scale. The scale covered the whole width of the row. Data used for the simulations covered 9 days at the end of April and in the beginning of May.

4.3.2. Tomato '90

Climate variables

Global radiation was measured outside the greenhouse. The fluxes of PAR, NIR and UV, and their separation into diffuse and direct components were calculated from global radiation as described in Chapter 2.

Table 4.4 Climate variables, greenhouse and crop parameters in the tomato '90 experiment used in the simulations.

	Derivation / Value	Remark
Climate variables		
Global radiation	measured	
Fraction diffuse in global radiation	calculated	as described in Ch. 2
PAR ($J\ m^{-2}\ s^{-1}$)	calculated	as described in Ch. 2
NIR ($J\ m^{-2}\ s^{-1}$)	calculated	as described in Ch. 2
Temperature inside greenh. (T_{in})	measured	
CO_2 concentration	measured	
VPD greenhouse air	measured	
Roof temperature	calculated	$T_{roof} = T_{out} + 0.33\ (T_{out}-T_{in})^*$
Ground temperature	calculated	assumed at 20 °C
Greenhouse parameters		
Transmissivity	$T_{r,dif} = 0.65$; $T_{r,dir}$ calculated	$T_{r,dir}$ calculated from parameters of PTG glasshouse
Azimuth	East-West	
Ground reflectivity	0.4	assumed
Crop parameters		
Leaf Area Index	measured	
Row dimensions	measured	
Row azimuth	East-West	

* Temperature outside greenhouse was measured

Greenhouse environment

Heating pipes (the 'slaves') were located under the canopy. Thermal screens were closed at night until the beginning of May.

Crop data

Plant density was 2.3 plants per m^2. At the beginning of the measurements (Day 95) the LAI was about 1.8 and the crop height about 1.40 cm. From Day 102 to Day 134 the LAI varied between 1.9 and 2.2. It was assumed here that from Day 134 onwards the LAI was 2.2 until Day 150.

Transpiration measurements

Four plants were placed on the weighing scale. The scale covered the whole width of the row. The weight of the plant was recorded every 2 minutes.

4.3.3. Sweet pepper

Climate variables

Global radiation was measured outside the greenhouse. The flux diffuse global radiation was measured with a shadowband pyranometer. The measurements were corrected for the fraction of diffuse radiation that is obscured by the shadowband according to the procedure given by Dehne (1984). The fluxes PAR, NIR and UV, and the separation into diffuse and direct components in these fluxes were calculated from global radiation as described in Chapter 2.

Table 4.5. Climate variables, greenhouse and crop parameters in the sweet pepper and cucumber experiments used in the simulations.

	Derivation / Value	Remark
Climate variables		
Global radiation	measured	
Fraction diffuse in global radiation	measured	
PAR ($J\ m^{-2}\ s^{-1}$)	calculated	as described in Ch. 2
NIR ($J\ m^{-2}\ s^{-1}$)	calculated	as described in Ch. 2
Temperature inside greenh. (T_{in})	measured	
CO_2 concentration	measured	
VPD greenhouse air	measured	
Roof temperature	calculated	$T_{roof} = T_{out} + 0.33\ (T_{out}-T_{in})$*
Ground temperature	calculated	assumed at 20 °C
Greenhouse parameters		
Transmissivity	$T_{r,dif} = 0.61$; $T_{r,dir}$ calculated	$T_{r,dir}$ calculated from parameters of PTG glasshouse
Azimuth	North-South	
Ground reflectivity	0.4	assumed
Crop parameters		
Leaf Area Index	estimated	
Row dimensions	estimated	
Row azimuth	North-South	

* Temperature outside greenhouse was measured

Greenhouse environment

Heating pipes were located both under and above the canopy, 4 and 2 per span, respectivly.

Crop data

Plant distance within a plant row was 0.40 cm. Plant density was 3.13 plants per m2. The LAI was estimated from measurements on plant height at Day 131 and Day 148. By comparing these heights with the heights and LAI of a sweet pepper crop that was grown in 1988 the LAI of the actual crop was estimated. Thus the LAI was assumed to be 3.5 at the first measurements (Day 115) and to increase to 7 at Day 223. Crop height at Day 115 was estimated at 1.60 m, and to increase to 2.25 m until Day 223.

Transpiration measurements

Three weighing scales were located in each compartment. Each scale carried 3 plants in a plant row, being placed under half of the width of a crop row. The weighing scales were located under the gutters. The weight of the plants was recorded every minute, and from these weights 10 minutes averages of transpiration rate were calculated. Measurements of two weighing scales, one located in the west side of a crop row and one located in the east side of a crop row, were averaged.

4.3.4. Cucumber

Climate variables

Measurements and calculation on radiation outside the greenhouse were performed as described above for the sweet pepper experiment.

Greenhouse environment

Heating pipes were located both under and above the canopy, 4 and 2 per span, respectivly.

Crop data

Plant distance within a plant row was 0.40 cm. Plant density was 1.56 plants per m^2. The LAI was measured and varied in the period of measurements between 2.2 and 3.4. The crop was assumed to have already reached the supporting wire (at 2.15 m) at the beginning of the measurements.

Transpiration measurements

Measurements were done the same way as for the sweet pepper crop. Weighing scales carried 2 plants, and covered half the width of the crop row. Measurements of two weighing scales, one located in the west side of a crop row and one located in the east side of a crop row, were averaged.

4.3.5. Some remarks on the derivation of data for model input

Sweet pepper and cucumber experiments

In the sweet pepper and cucumber experiments greenhouse transmissivity (compartment 302) was estimated from the data of another glasshouse compartment (210) at the PTG, of which detailed measurements were available, but with some change in the characteristics of the construction. In compartment 302 screens were folded under the gutter, whereas in 210 they were located (as is normally the case) under the beams. The dimensions of beams and gutters were changed to account for the presence of the folded screens. With Bots' model diffuse transmissivity was than calculated to be 0.68. As not all radiation intercepting elements in the compartment were taken account of in the calculation with Bot's model, and the glasshouse was measured to have a diffuse PAR transmissivity of about 0.61 (G. van Holsteijn, PTG, pers.

comm., 1994), a correction was applied by multiplying both direct and diffuse transmissivity with 0.90. Calculated diffuse transmissivity, T_{rdif}, was decreased from 0.68 to 0.61 hereby.

Roof temperature was calculated from both inside and outside air temperature. It was assumed to be equal to the outside temperature plus one-third the temperature difference with the inside air temperature, as the sensible heat conductivity on the outside of the glass is about twice as high as on the inside (G. Bot, IMAG-DLO, pers. comm., 1993). This was checked with detailed calculations by a greenhouse model as done by F. Zwart (IMAG-DLO). The simple calculation correlated well with the detailed calculation, with most differences being less than 4 °C.

Tomato experiments

Diffuse transmissivity of the glasshouse of the tomato '90 experiment has been measured to be 0.65 (Stanghellini, pers. comm.). T_{rdif} of the glasshouse of the tomato '86 experiment was taken at 0.65, following Marcelis (1989). Therefore, direct radiation transmissivities were assumed to be the same as those of compartment 210 at the PTG.
For the tomato '90 experiment the roof temperature was calculated as with the sweet pepper and cucumber experiments. When either the energy or shading screen, or both, was closed, roof temperature was assumed to be equal to air temperature.

4.4. Sensitivity analysis

4.4.1. Introduction

Sensitivity analysis is not only an important aspect of model development and testing, it also is an important tool in analysing and understanding the complex interactions in the greenhouse-crop system. Here some model runs were performed with the multilayer transpiration model, in combination with the C_i/C_s stomatal submodel or the negative-exponential submodel.

Standard conditions for all simulated responses were: air VPD at 1 kPa, air temperature at 22 °C, pipe temperature 25 °C (with specific surface 0.09), roof temperature 10 °C, ground temperature 20 °C. A closed canopy with LAI at 3 was assumed, with spherical leaf angle distribution. Boundary layer conductance (g_b) was at 0.01 m s^{-1}.
Standard values of the parameters of the C_i/C_s-model for stomatal conductance (Eqn 4.17) were: $Fcics$ = 0.8, $fc1$ = 0.1 and g_{res} = 0.001.
Values of the parameters of the negative-exponential function for stomatal conductance (Eqn 4.18) were: g_{smax} = 0.020, $cd1$ = 0.98, $cd2$ = 0.012, $cd3$ = 0.25, $cd4$ = 0.

4.4.2. Results

Negative-exponential submodel

Responses to incident global radiation

The effects of the parameters that affect the response of canopy transpiration, E_{cr}, to incident global radiation (Q_g) are shown in Fig. 4.3. They all had a large effect on E_{cr}. Parameters g_{smax} and $cd2$ had an almost similar effect.

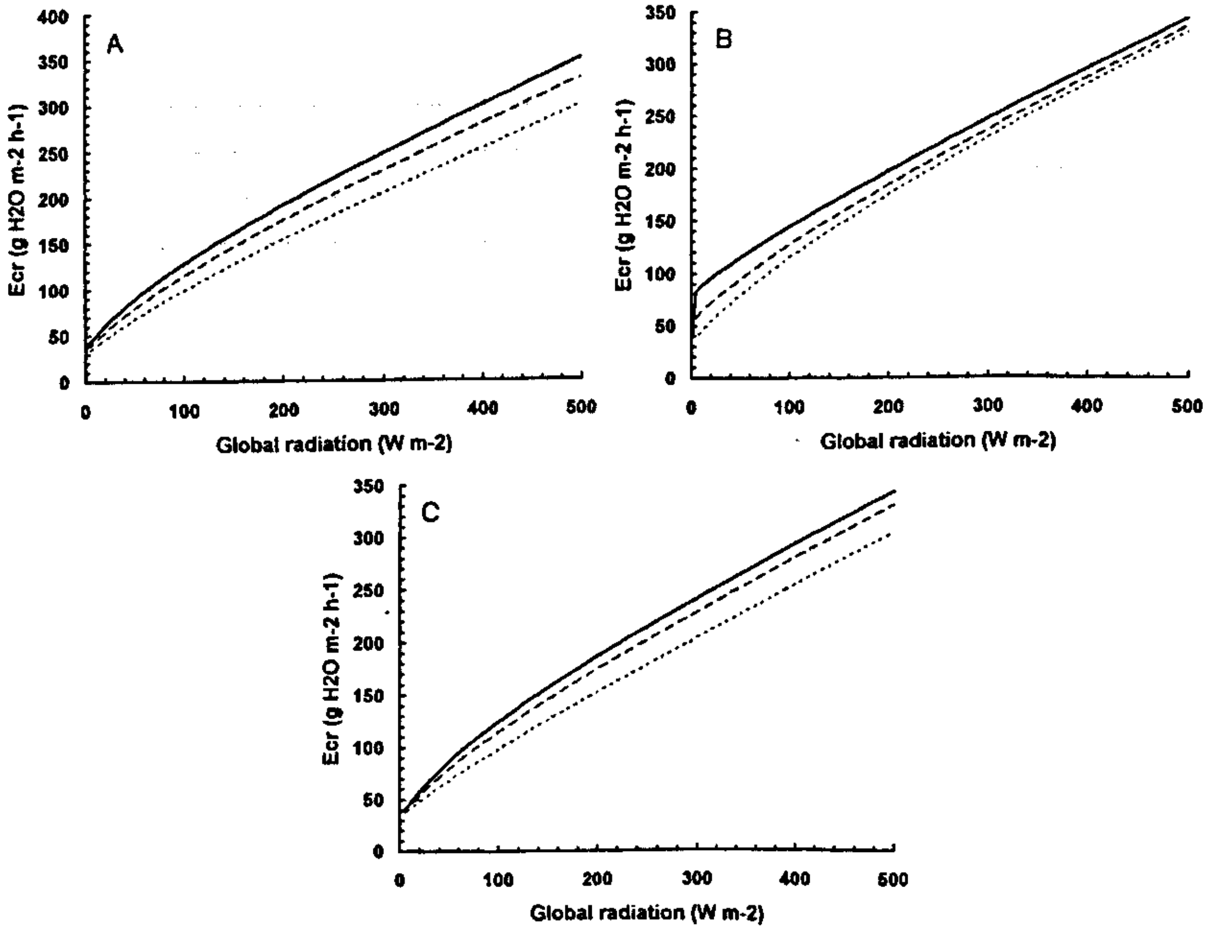


Figure 4.3 Simulated responses of crop transpiration (g H_2O m^{-2} h^{-1}) to incident global radiation (W m^{-2}) using the negative-exponential submodel for stomatal conductance (Eqn 4.18).

A. Variation of parameter g_{smax}: 0.025 (solid line), 0.020 (dashed line) and 0.015 (dotted line).

B. Variation of parameter *cd1*: 0.98 (solid line), 0.95 (dashed line), and 0.90 (dotted line).

C. Variation of parameter *cd2*: 0.015 (solid line), 0.012 (dashed line), and 0.08 (dotted line).

The conditions at the leaf level for the standard run are shown in Fig. 4.5. At low global radiation levels leaves at the top of the canopy were colder because roof temperature was lower than ground temperature. Note that the VPD at the leaf surface, D_s, was simulated to be generally lower for leaves in the upper part of the canopy, as they were transpiring more. At high radiation levels the situation became reversed.

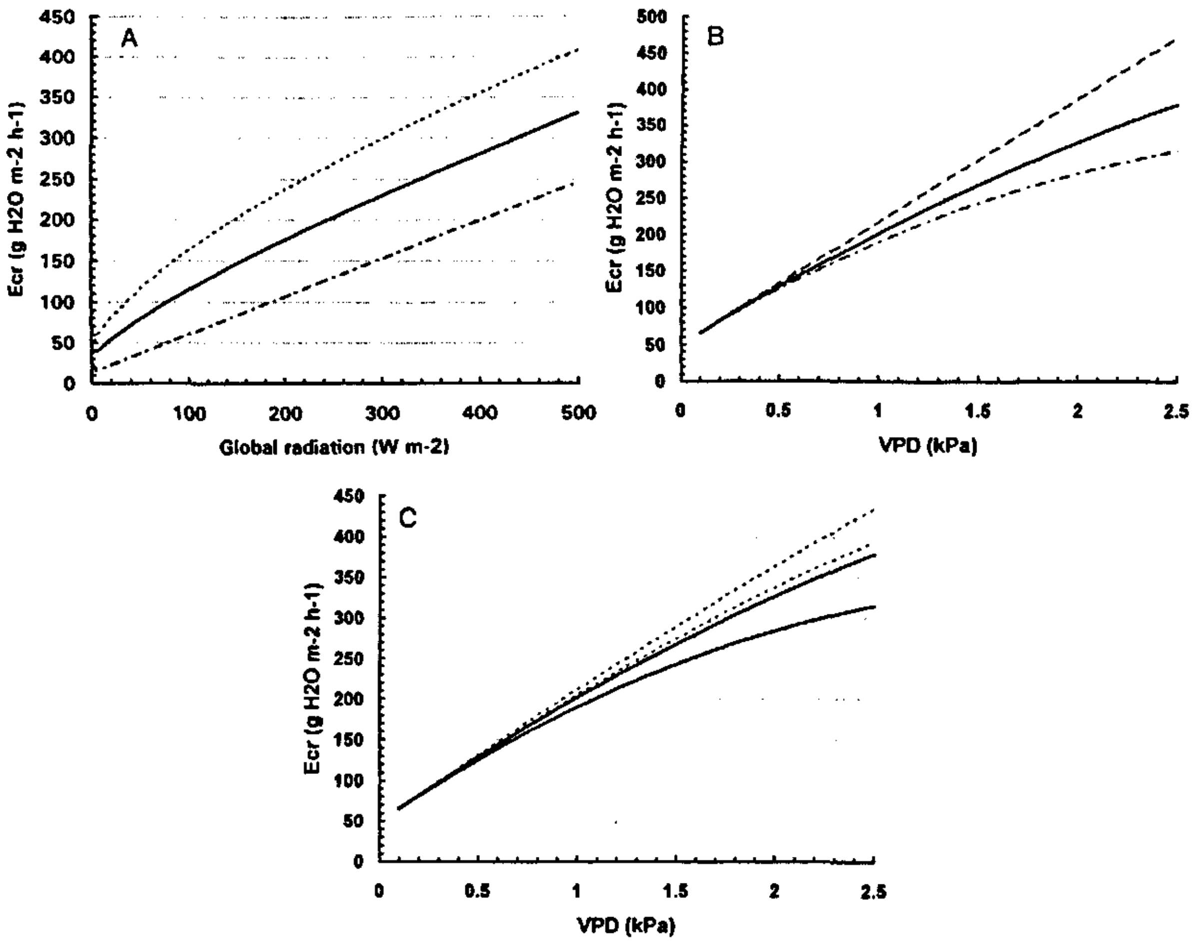


Figure 4.4 Simulated responses of crop transpiration (g H_2O m^{-2} h^{-1}) using the negative-exponential submodel for stomatal conductance.

A. Response to incident global radiation (W m^{-2}) with air VPD at 0.5 (dash-dot line), 1.0 (solid line) and 1.5 kPa (dotted line).

B. Response to air VPD: variation of parameter cd3 of Eqn 4.18 for g_s: cd3 = 0.05 (dashed line), 0.25 (solid line) and 0.4 (dash-dot line).

C. Response to VPD: in the negative-exponential stomatal model VPD was assumed to be either air VPD or leaf surface VPD for the same value of parameter *cd3*. Solid lines: air VPD, dotted lines: leaf surface VPD. Parameter *cd3* was either 0.25 (upper line of each pair) or 0.4 (lower one).

Response to VPD

VPD has a large effect on E_{cr} (Fig. 4.4). This effect was simulated to be relatively larger at lower levels of Q_g. E_{cr} was significantly affected depending on whether VPD in the negative-exponential model of g_s was assumed to be either air VPD or leaf surface VPD.

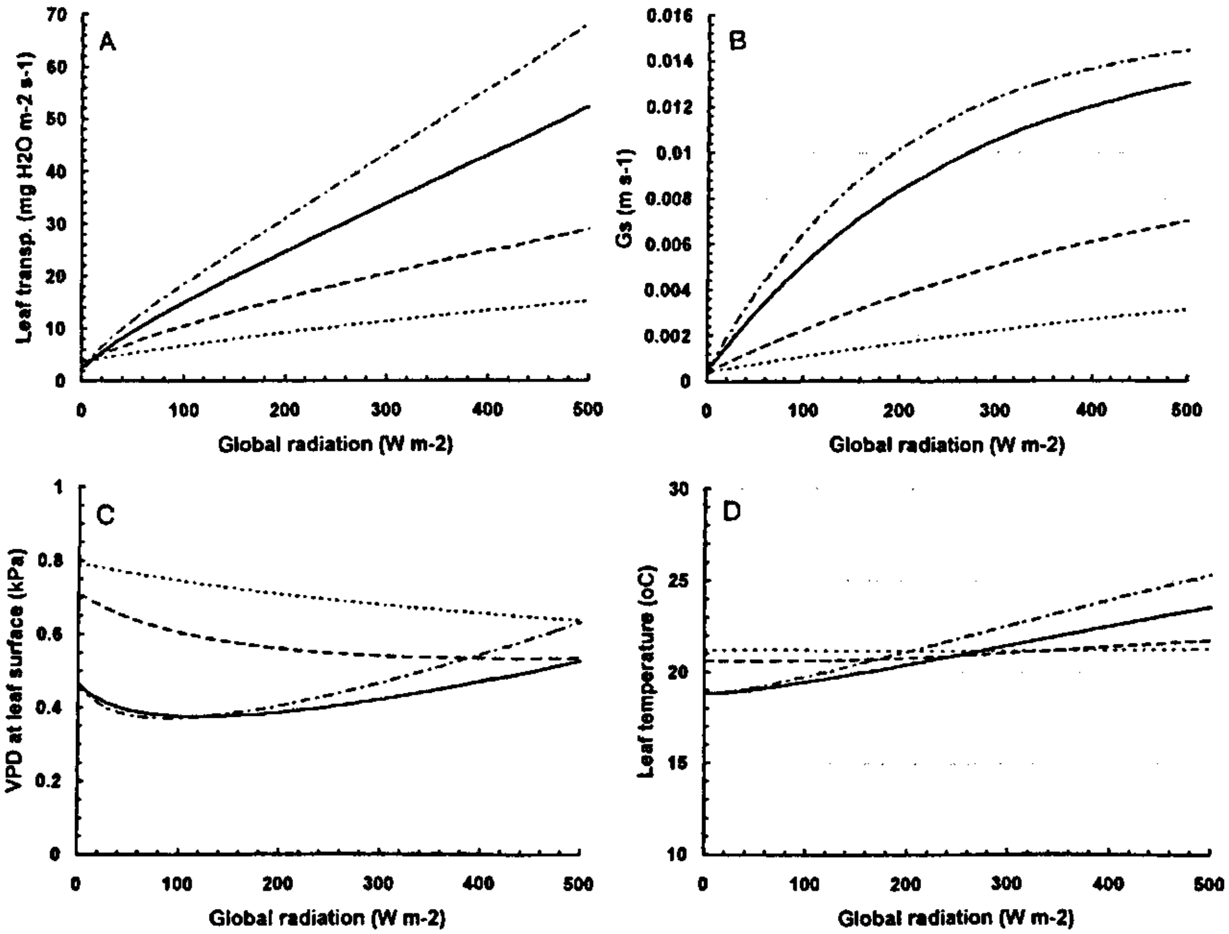


Figure 4.5 Simulated responses at the leaf level to global radiation incident at the canopy surface, using the negative-exponential submodel for stomatal conductance, for the standard conditions.

A: leaf transpiration,

B: stomatal conductance,

C: VPD at the leaf surface, and

D: leaf temperature.

Solid line: average leaf at the top of the canopy, *dashed line*: average leaf in the middle of the canopy, *dotted line*: average leaf at the bottom of the canopy, *dash-dot line*: sunlit leaves directed towards sun.

Other responses

The effect of some other factors on E_{cr} was assessed at two levels of global radiation: 100 and 400 W m^{-2} (Table 4.6), using the negative-exponential submodel for stomatal conductance.

Table 4.6 Simulated effect on canopy transpiration (E_{cr}, g H_2O m^{-2} h^{-1}) of some factors. One given factor was changed while the others were kept at their standard values. The effect is expressed as the percentage change relative to the standard conditions. At the standard conditions at 100 W m^{-2} global radiation E_{cr} was 115 g H_2O m^{-2} h^{-1}, and at 400 W m^{-2} it was 281 g H_2O m^{-2} h^{-1}

Factor	Global radiation	
	100 W m^{-2}	400 W m^{-2}
Pipe temperature: 25 -> 50 °C	+3	+2
Roof temperature: 10 -> 15 °C	+8	+5
Leaf Area Index: 3 -> 2	-24	-22
Leaf angle distribution: spherical -> horizontal	+1	+2
Reflection ground: 0 -> 50 %	+6	+8
Boundary layer conductance (g_b): 0.01 -> 0.005	-21	-19

It was calculated that LAI had a large effect on E_{cr}; a decrease from 3 to 2 decreased E_{cr} by about 23 %, which was the same as the decrease in absorbed radiation. Roof temperature affected E_{cr} significantly at low levels of global radiation, whereas pipe temperature did have little effect due to its low specific surface. The change in leaf angle distribution had little effect on total radiation absorption, consequently the change in E_{cr} was negligible. Only at lower LAI's became this effect important (not shown). Halving the boundary layer conductance decreased E_{cr} by about 20 %. Such a decrease in g_b could be possible when leaf size is greatly increased, for instance when considering leaves of cucumber instead of tomato or sweet pepper.

The C_i/C_s-submodel for stomatal conductance

The response of simulated crop transpiration to incident global radiation followed a concave curve (Fig. 4.6), as opposed to the curve of the response simulated with the negative-exponential model. Decreasing the C_i/C_s-ratio decreased stomatal opening and consequently the rate of transpiration. Parameter *fc1* caused similar changes in the response of E_{cr} to VPD as parameter *cd3* in the negative-exponential submodel.

Simulated crop gross photosynthesis (P_{gc}, g CO_2 m^{-2} h^{-1}) decreased with increasing air VPD for *fc1* at 0.5. The rate of decrease was similar for both values of *Fcics*. Relative decreases were, for the parameter values chosen, maximal 20 %. The increase in P_{gc} for *fc1* at 0.1 was due to decreasing leaf temperatures, which increased the initial slope of the average leaf light response curve more than it decreased the maximal value of leaf gross photosynthesis.

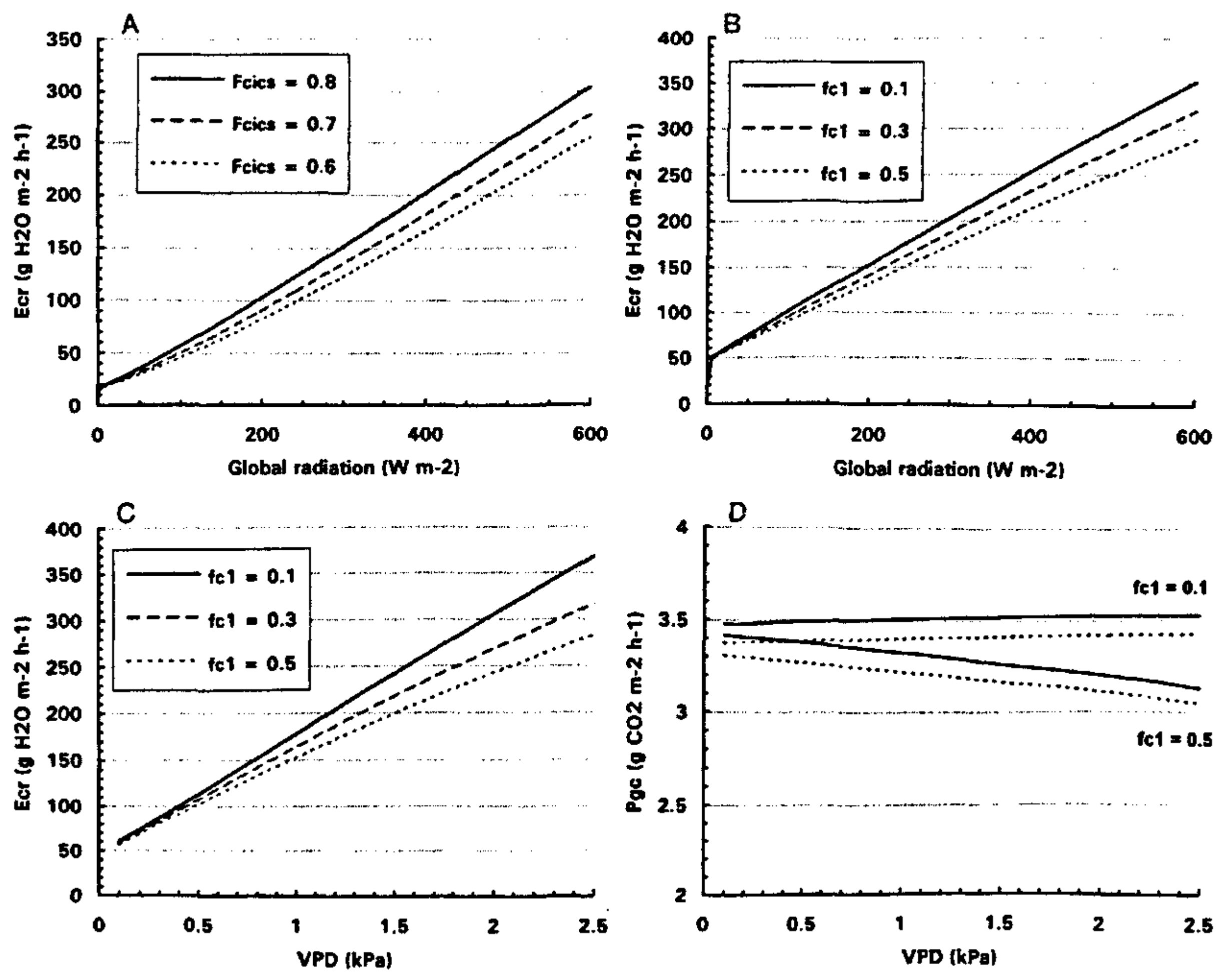


Figure 4.6 Simulated responses of crop transpiration (E_{cr}, g H_2O m^{-2} h^{-1}) or crop gross photosynthesis (P_{gc}, CO_2 $m^{-2}h^{-1}$) using the C_i/C_s-submodel for stomatal conductance. g_{res} was at 0.001 m s^{-1}.

A. Response of E_{cr} to incident global radiation: variation of parameter *Fcics* for *fc1* at 0.1.

B. Response of E_{cr} to incident global radiation: variation of parameter *fc1* for *Fcics* at 0.7.

C. Response of E_{cr} to air VPD, with Q_g at 250 Wm^{-2}: variation of parameter *fc1* for *Fcics* at 0.7.

D. Response of P_{gc} to air VPD, with Q_g at 250 W m^{-2}, for *Fcics* = 0.8 (solid line) or 0.7 (dotted line), and *fc1* at 0.1 (upper line) or at 0.5 (lower line).

4.5. Model results

Comparisons of model results and measurements were in most cases evaluated by calculation of r^2, the standard error of estimate (SEE), and/or by regression of simulated E_{cr} on measured E_{cr}. The SEE was also expressed as a percentage of the averaged measured transpiration.
r^2 was calculated by

$$r^2 = 1 - \frac{\sum (y_i - y_{i,est})^2}{\sum (y_i - y_{av})^2}$$

where y_i is measured transpiration, y_{est} is the simulated transpiration and y_{av} the average of measured transpiration. Note that r^2 can have a negative value when simulating transpiration is worse than just taking the average. With the regression of simulated E_{cr} on measured E_{cr} the slope of the fit and the intercept were calculated.

4.5.1. The row and greenhouse cover effect

The effect on transpiration of plants standing in a row, and the effects of receiving varying amounts of shade from the glasshouse cover during the day, were calculated for a sunny day at 25 April (25 MJ m^{-2} total global radiation), for a crop with row height 1.4 m, and row width 1.2 m, and north-south orientation. It was calculated that the row effect caused a clearly discernible dip in the rate of transpiration around noon (a closed canopy would have a sinusoidal pattern of transpiration) (Fig. 4.7A). Because the row was simulated to stand beneath the gutter, the dip was enlarged as a result of the shading of the gutter directly overhead. The effect of the heterogeneous direct transmissivity alone on transpiration is shown in Fig. 4.7B. At some parts of the day noticeable deviations occurred from the pattern of transpiration calculated in the standard way.

When one half of the row was considered large deviations with the transpiration rate of a closed canopy arose, in which the effect of the row structure was somewhat larger than the effect of glasshouse construction shade (Fig. 4.7C).

Measured and simulated sweet pepper crop transpiration for Day 194 are depicted in Fig. 4.8A,B. Measurements were from one weighing scale, and were closely approximated by the simulations (using the Ball et al. stomatal model). In this simulation account was taken of the place of weighing scales in the row and their position relative to gutters and ridges.
In the simulations it was as a standard assumed that the canopy was horizontally homogeneous (closed canopy), and that the direct radiation transmission by the greenhouse cover was evenly distributed over the canopy.

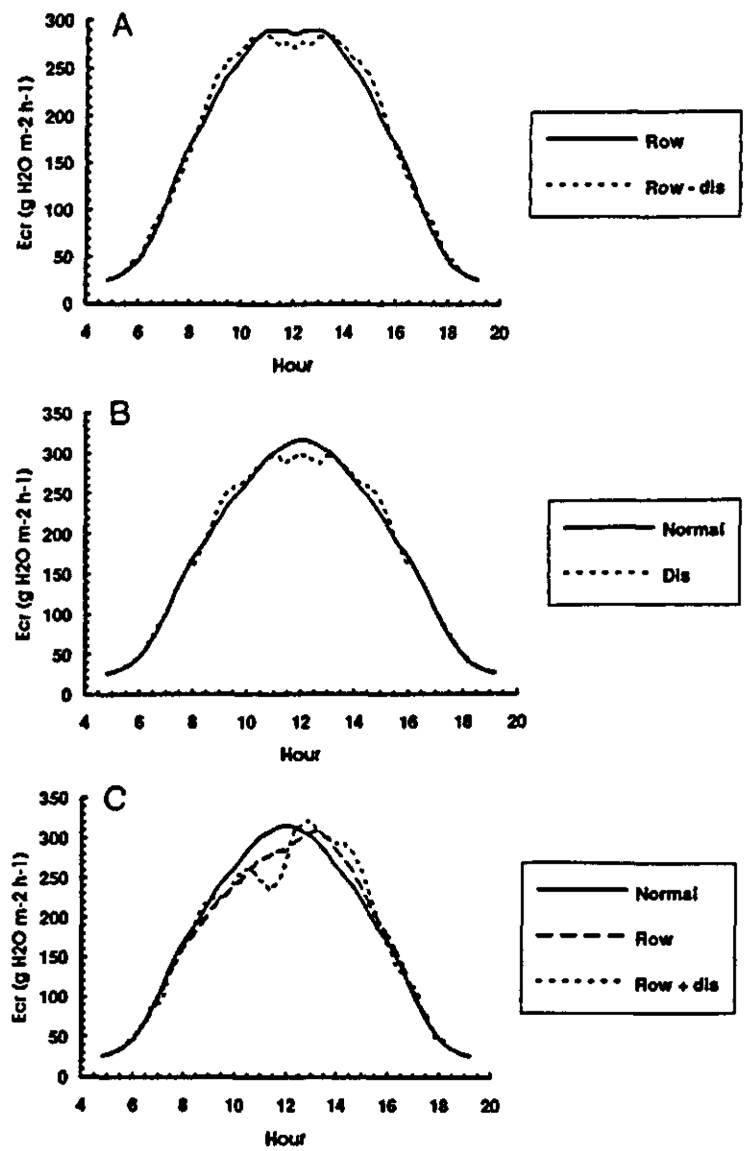


Figure 4.7 Simulated diurnal transpiration of a crop at 25 April, in north-south oriented glasshouse compartment 302. **A.** Transpiration of a row crop assuming evenly distributed shade of the glasshouse construction ('Row'), and of a row crop taking account of the patterns of shades of gutters and ridges ('Row - dis'). Cover shade calculation with direct transmissivity "point-model" (see Appendix II). **B.** Transpiration of the plant stand, not taking account of the row effect, but assuming uniform direct radiation transmissivity ('Normal'), and when taking of the distributed shade of the construction ('Dis'). Calculation with direct transmissivity "area-model" (see Appendix II). **C.** Calculated transpiration of a closed canopy assuming uniform direct transmissivity of the glasshouse construction ('Normal'), of a plant stand covering the half the width of the row, but with uniform direct transmissivity ('Row'), and transpiration of a plant stand covering half the width of a row, but taking into account the patterns of shades of gutters and ridges ('Row + dis'). Calculation with direct transmissivity "point-model" (see Appendix II).
LAI at 3, row height = 1.4 m, row width = 1.2 m. Total daily global radiation was 25 MJ m^{-2}. The daily course of global radiation was generated as described by Gijzen (1992).

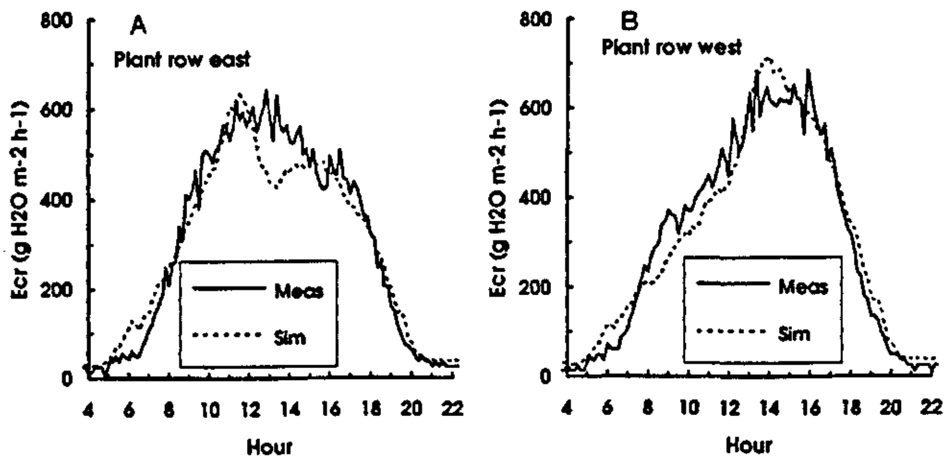


Figure 4.8 Measured and simulated crop transpiration (g H_2O m^{-2} h^{-1}) of sweet pepper at Day 194, in compartment 3. Measurements are from one weighing scale. In the simulations account is taken of the row structure and of the shading effects of gutter and ridge on the plants placed on the weighing scale. **A.** Weighing scale located at east side of crop row. **B.** Weighing scale located at west side of crop row.

4.5.2. Relation of transpiration to absorbed radiation

To compare transpiration rates of the crops, the measured daytime daily total transpiration is plotted against absorbed daily global radiation (Fig. 4.9). It appeared that the cucumber and sweet pepper crops transpired more per unit of absorbed radiation than the tomato crops. When a line was fitted through the points, and forced through the origin, 'transpiration efficiencies' were obtained of 256, 250, 408 and 335 g H_2O per MJ global radiation absorbed, for tomato '86, tomato '90, sweet pepper and cucumber, respectively (Table 4.7). When daytime transpiration was expressed on an energy basis, the ratio's of energy of transpiration to global radiation were 0.64, 0.63, 0.84 and 1.02, respectively.
Ratio daytime E_{cr} to $Q_{gd,abs}$

Table 4.7 Average measured daytime E_{cr} (g H_2O m^{-2} h^{-1}), and the ratio of daily crop transpiration to daily absorbed global radiation ($Q_{gd,abs}$, MJ m^{-2} d^{-1}). The ratio was determined by a linear fit, forced through the origin, of daytime E_{cr} to $Q_{gd,abs}$.

Crop	Average measured daytime E_{cr} (g H_2O m^{-2} h^{-1})	Ratio daytime E_{cr} to $Q_{gd,abs}$ (g H_2O/MJ)
Tomato '86	133	256
Tomato '90	105	250
Sweet pepper	207	408
Cucumber	93	335

Daily measured E_{cr} was also expressed as a ratio to daily crop transpiration of a wet big-leaf (E_{wet}), i.e. a leaf that has an infinitely high stomatal conductance. All crops showed about

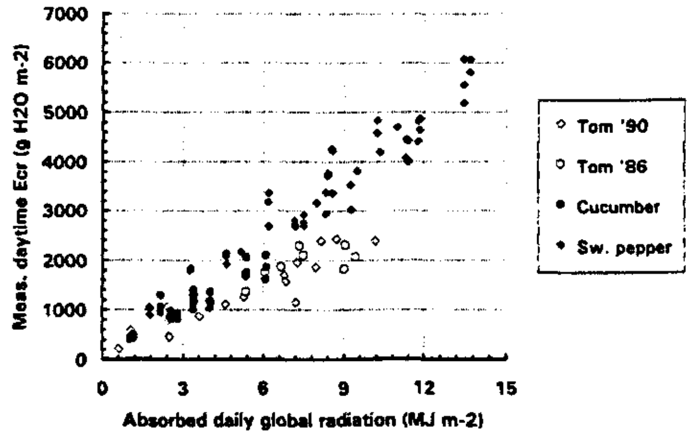


Figure 4.9 The relation between the measured daytime daily total of transpiration (g H_2O m^{-2} d^{-1}) and absorbed daily global radiation (MJ m^{-2} d^{-1}), for the tomato '86, tomato '90, sweet pepper and cucumber crops

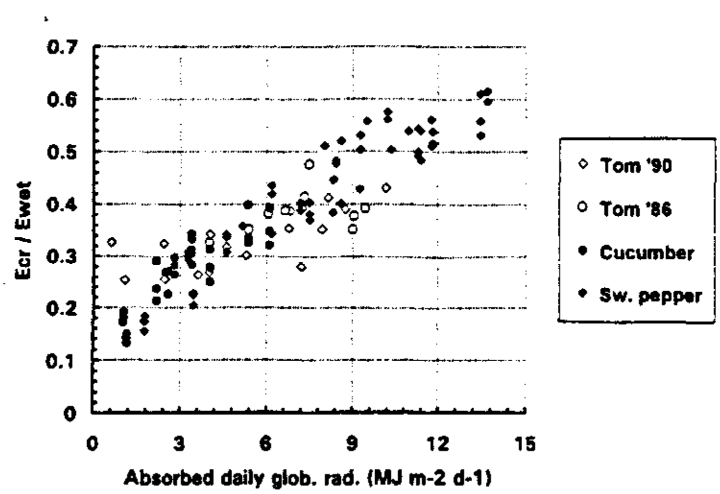


Figure 4.10 The relation between the ratio of measured daily daytime transpiration to simulated daily transpiration of a wet big-leaf (E_{cr}/E_{wet}) and absorbed daily global radiation (MJ m^{-2} d^{-1})

4.5.3. Test of some models of stomatal response

Regression models of the negative-exponential type of the response of g_s to various climatic factors were developed for cucumber, tomato and sweet pepper by Bakker (1991), Nederhoff *et al.* (1992) and Nederhoff & de Graaf (1993). Here, crop transpiration was simulated using the negative-exponential stomatal model for g_s parameterized according to these authors, in conjunction with the multilayer canopy transpiration model.

In most cases measured crop transpiration was significantly overstimated, high values of the intercept were obtained or a high value of the SEE. With the tomato '86 experiment the best fit was obtained, using data of Bakker (1991).

Table 4.8 Results of using the negative-exponential stomatal conductance model in the multilayer crop transpiration model. The stomatal submodel was parameterized with data from Bakker (1991), Nederhoff et al. (1992) and Nederhoff & de Graaf (1993). Lineair regression was done of calculated E_{cr} on measured E_{cr}, giving an intercept and a slope.

Experiment	Parameter source⁺	Intercept	Slope (-)	r^2	SEE	
		g m^{-2} h^{-1}			g m^{-2} h^{-1}	%
Tomato '86	1	8.	0.94	0.83	39	29
	3	16.	1.54	0.34	77	58
Tomato '90	1	28.	0.97	0.73	41	39
	3	14.	1.41	0.19	72	68
Sweet pepper	1	63.	0.90	0.75	87	42
	2	7.	1.17	0.79	80	39
Cucumber	1	15.	1.41	-0.05	63	68
	3	-1.	1.53	0.02	61	66

⁺Parameter source 1: Bakker (1991), 2: Nederhoff *et al.* (1992), 3: Nederhoff & de Graaf (1993)

4.5.4. Results of tuning of the models

The models were calibrated by fitting the parameters of the submodels for stomatal conductance in such a way that the sum of squares of the differences between measured and simulated crop transpiration was minimal. The datasets on which calibration took place consisted of 20 minute records, except for the tomato '86 experiment, where 15 minute records were used. Optimisation of the parameters (for daytime transpiration) was done according to the 'Simplex - down hill'-method.

Night transpiration

The value of the parameter for maximal leaf conductance at night, g_{md} was adjusted by fitting, by eye, simulated night transpiration to measured night transpiration. The values found are given in Table 4.9. They are global estimates as nightly transpiration could not be simulated accurately, as ground surface temperature and cover temperature could not be estimated accurately. The values of g_{md} differed significantly between experiments.

Table 4.9 Values of maximal leaf conductance at night (g_{md}) found by fitting (by eye) simulated E_{cr} to measured E_{cr}.

Crop	g_{md} (m s^{-1})
Tomato '86	0.001
Tomato '90	0.001
Sweet pepper	0.0005
Cucumber	0.0015

Fit of the relation based on global radiation and VPD

Results of this fit are given in Table 4.10. Relative high SEE values were obtained.

Table 4.10 Results of the fit of the relation $E_{cr} = a\ Q_g + b$ VPD on measured transpiration rates.

Crop	Parameters		r^2	SEE	
	a	b		g H_2O m^{-2} h^{-1}	%
Tomato '86	0.14	16	0.86	37	28
Tomato '90	0.11	18	0.69	44	42
Sweet pepper	0.28	18	0.77	84	41
Cucumber	0.13	20	0.73	31	33

Fit of the Makkink-formula

Results of this fit are given in Table 4.11. As evidenced by the low values of r^2 and the high values of SEE, this formula did not work out very well. The Makkink-formula has good predicting abilities for fields crops, which is presumably due to the fact that in the field high temperature and high VPD are strongly correlated. In the greenhouse these factors are much less coupled.

Table 4.11 Results of the fit of the Makkink-formula on measured transpiration rates.

Crop	Parameter C	r^2	SEE	
			g H_2O m^{-2} h^{-1}	%
Tomato '86	0.61	0.84	39	29
Tomato '90	0.51	0.64	48	46
Sweet pepper	0.75	0.49	84	41
Cucumber	0.73	0.55	40	43

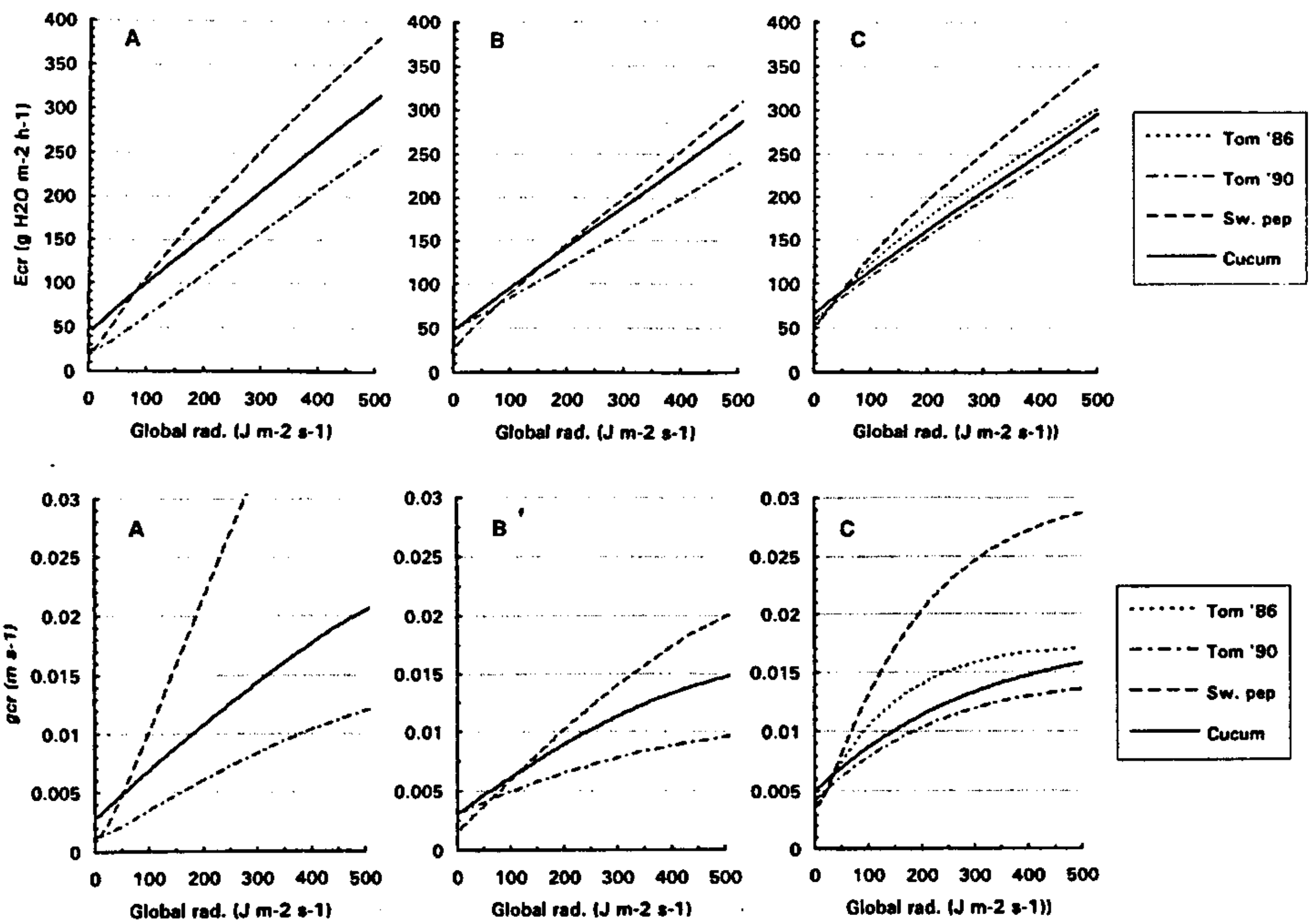


Figure 4.11. Simulated responses of E_{cr} and g_{cr} to incident global radiation (inside greenhouse), and of g_{cr} to VPD and CO_2 concentration of the greenhouse air, using the multilayer canopy transpiration model and using parameter values for the stomatal conductance models as obtained by fitting to the experiments.

Standard values of parameters and variables: LAI at 3, spherical leaf angle distribution, solar elevation at 45°, fraction diffuse 0.50, Q_g at 300 J m^{-2} s^{-1}, CO_2 concentration at 350 µl l-1, air temperature at 22 °C, VPD of greenhouse air at 1.0 kPa, temperature of cover, ground and pipes at 10, 20 and 25 °C, respectively.

A. Using stomatal model of Ball et al..

B. Using the Ci/Cs-model.

C. Using the negative-exponential model.

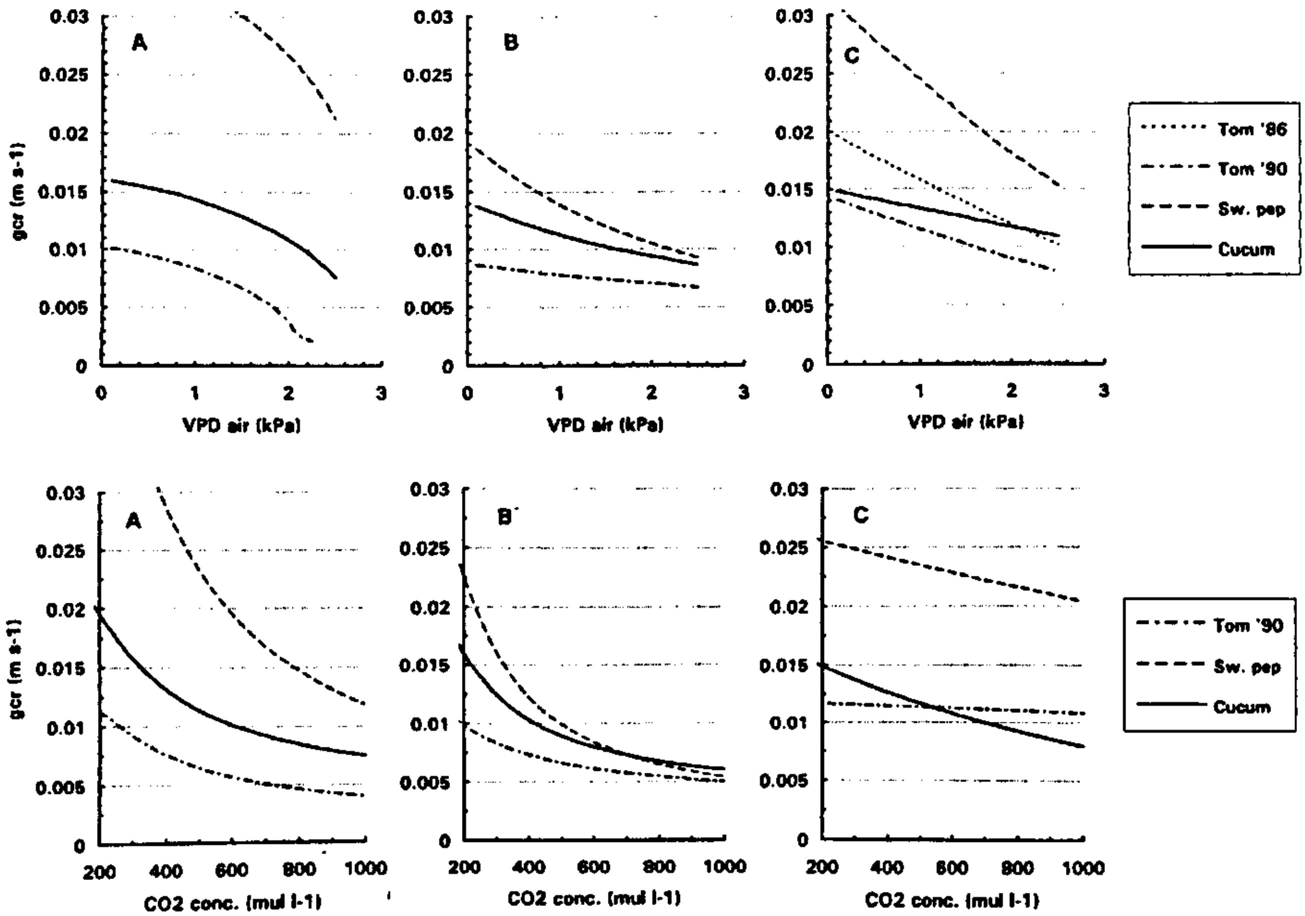


Figure 4.11. Continued

The stomatal model of Ball et al

With the model of Ball *et al.* parameter *b* was kept constant at such a value (fit by eye) that the transition between nighttime and daytime canopy transpiration was gradual. It appeared that the value of *b* had a large influence on crop transpiration.

Parameter *m* was very high in the sweet pepper experiment (Table 4.12). This value gave maximal leaf stomatal conductances higher than 0.03 m s^{-1} and high canopy conductance (Fig. 4.11). Note that the sensitivity of canopy transpiration to g_s becomes very small for g_s above 0.03 m s^{-1}. Apparently the high value of *m* was obtained because the canopy model tended to underestimate the measured transpiration rate at most radiation levels. Both in the tomato experiment and in the cucumber experiment a low value for *m* was found.

Table 4.12 Results of the fit of the Ball et al-stomatal conductance model.

Crop	Parameters		r^2	SEE	
	m (-)	b (m s^{-1})		g H_2O m^{-2} h^{-1}	%
Tomato '90	5.0	0.00025	0.86	30	29
Sweet pepper	15.4	0.00025	0.91	55	27
Cucumber	6.4	0.001	0.91	18	19

Table 4.13 Results of the fit of the parameters of the C_i/C_s stomatal conductance model $(C_i - \Gamma) / (C_s - \Gamma) = Fcics \exp(-fc1\ D_s)$ and $g_l = 1.6\ P_g / (C_s - C_i) + g_{res}$.

Crop	Parameters		r^2	SEE	
	Fcics	fc1		g H_2O m^{-2} h^{-1}	%
Tomato '90	0.56	-	0.83		
	0.50	0.36	0.86	29	28
Sweet pepper	0.90	-	0.90		
	0.98	0.28	0.92	41	19
Cucumber	0.83	-	0.84		
	0.72	0.35	0.91	18	19

The C_i/C_s stomatal conductance model

Here the value of g_{res}, the "offset-variable" in the stomatal model, was estimated by making a smooth transition between nighttime transpiration and daytime transpiration (i.e. by fitting by eye). The values of g_{res} obtained in this way were 0.001, 0.0005 and 0.001 m s^{-1}, for tomato '90, sweet pepper and cucumber, respectively.
As with the model of Ball *et al* high values of r^2 were obtained. The addition of parameter *fc1* increased the goodness of fit significantly. Low values of *Fcics* were obtained in the tomato experiment, and high values in the sweet pepper experiment (Table 4.13, Fig. 4.11). In the latter experiment the value of *Fcics* was estimated too high, as it implicated a maximal value of g_{smax} of more than 0.1 m s^{-1} and a very high canopy conductance (Fig. 4.11). Thus, as with the model of Ball *et al.*, measured crop transpiration could not be reached unless excessive values of g_s had to be assumed.

The negative-exponential submodel

With the negative-exponential model a similar goodness of fit was obtained as with the two photosynthesis submodels (Table 4.14). A scatterplot of simulated transpiration against measured transpiration in shown in Fig. 4.12. The value of g_{smax} was highest in the sweet pepper experiment and lowest in the two tomato experiments. In all experiments the inclusion of the leaf-air VPD effect somewhat increased the goodness of fit. It had a significant effect on the value of g_{smax}.The effect of leaf-air VPD seemed more or less similar between crops and experiments. Addition of a CO_2 effect on stomatal conductance (assuming an exponential decrease with CO_2 concentration) decreased somewhat the standard error in the sweet pepper experiment and in the cucumber experiment. Introduction of both the row-effect and the effect of distributed shading by the greenhouse cover in the simulation had a negligible effect on the goodness of fit, in all experiments.

Table 4.14 Results of the fit of the negative-exponential function describing leaf stomatal response to absorbed PAR, leaf-air VPD and CO_2 concentration of the greenhouse air
$g_s = g_{smax}$ (1-cd1 exp(- cd2 PARabs)) exp(-cd3 D_l) exp(-cd4 CO_2).
Intercept in g H_2O m^{-2} h^{-1}.
Fits were done with and without inclusion of responses to VPD and CO_2. Limits for parameters values imposed to the fitting algorithm were: $0.005<g_{smax}<0.030$, $0.8<cd1<1.0$, $0.005<cd2<0.030$, $0.01<cd3<0.8$.

Crop	Parameters					Interc.	Slope	r^2	SEE	
	g_{smax}	cd1	cd2	cd3	cd4 (*10^{-3})				g m^{-2} h^{-1}	%
Tomato '86	0.005	0.80	0.030	-		-35.	1.18	0.81	42	31
	0.012	0.95	0.030	0.45		-15.	1.07	0.86	36	27
Tomato '90	0.005	0.81	0.030	-		-13.	1.08	0.87	28	27
	0.013	0.93	0.012	0.42		-5.	1.03	0.89	26	25
	0.013	0.92	0.013	0.40	0.11	-6.	1.03	0.89	26	25
Sw. pepper	0.019	1.	0.013	-		-17.	0.98	0.89	58	27
	0.030	1.	0.023	0.36		0.	1.05	0.90	56	26
	0.030	1.	0.018	0.50	0.28	1.	0.97	0.92	51	24
Cucumber	0.009	0.90	0.009	-		-8.	1.06	0.89	20	21
	0.020	0.94	0.006	0.36		-3.	1.02	0.90	19	20
	0.016	0.90	0.012	0.22	0.88	-4.	1.02	0.92	18	18

By "manual" searching a single set of parameter values of *cd1*, *cd2*, *cd3* and *cd4* was sought that gave the best fit to all experiments. Only the value of g_{smax} was fitted to the experiments. Approximately the best fit was obtained with *cd1* = 0.98, *cd2* = 0.02, *cd3* = 0.4 and *cd5* = 0.0003. This fit was only slightly worse than the fits shown in Table 4.14. Values of r^2 and SEE were, for tomato '86, tomato '90, sweet pepper and cucumber, 0.85 and 28 %, 0.88 and 25 %, 0.91 and 24 %, and 0.89 and 21 %, respectively. Values of g_{smax} were 0.012, 0.012, 0.014 and 0.023 m s^{-1}.

The linear response function

When this function was used for describing the stomatal respons to PAR a goodness of fit was obtained that was similar to that of the other submodels (Table 4.15). The incorporation of the response to leaf-air VPD in the stomatal model caused a significant increase in goodness of fit in the tomato '86 and sweet pepper experiment. The overall goodness of fit obtained was somewhat less than with the negative-exponential model for the PAR and leaf-air VPD response.

Table 4.15 Results of the fit of the linear response function describing leaf stomatal response to absorbed PAR and VPD
$g_s = \min\{cf1\ PARabs, g_{smax}\}\ \exp(-cf2\ D_l)$.
Intercept in g H_2O m^{-2} h^{-1}. Fits were done with and without inclusion of the VPD response. Limits for parameters values imposed to the fitting algorithm were: $0.005<g_{smax}<0.030$, $0.00005<cf1<0.002$, $0.001<cf2<0.8$.

Crop	Parameters			Interc.	Slope	r^2	SEE	
	g_{smax}	cf1 (* 10^{-3})	cf2				g m^{-2} h^{-1}	%
Tomato '86	0.005	0.29		-30.	1.24	0.81	43	32
	0.008	0.28	0.37	-17.	1.09	0.86	36	27
Tomato '90	0.005	0.10		-23.	1.16	0.86	29	28
	0.010	0.15	0.41	-15.	1.10	0.88	28	26
Sw.pepper	0.023	0.21	-	-23.	1.06	0.89	57	27
	0.030	0.42	0.56	-2.	0.98	0.91	52	24
Cucumber	0.005	0.16	-	-20.	1.15	0.85	23	24
	0.024	0.20	0.58	-22.	1.14	0.86	23	23

The big-leaf model

Fits obtained with the big-leaf model appeared to be less good as with the multilayer models (Table 4.16). A negative-exponential response to PAR was adopted here, but no improved fit was obtained when a non-rectangular hyperbola was used for the PAR response (not shown). The reason for the differential goodness of fit was not quite understood. The responses of canopy conductance to low levels of global radiation (< 100 J m^{-2} s^{-1}) were quite different between the big-leaf model and the multilayer canopy + negative exponential stomatal model-combination. This could have played a major role. In all data sets the number of records with radiation levels lower than 100 J m^{-2} s^{-1} was quite significant.

Table 4.16 Results of the fit of the big-leaf canopy transpiration model. Intercept in g H_2O m^{-2} h^{-1}. Canopy conductance was described by
$g_{cr} = g_{lm,av}\ LAI\ (1 - cn1\ \exp(-cn2\ PARabs))\ \exp(-cn3\ D_a)$
Limits for parameters values imposed to the fitting algorithm were: $0.005<g_{lm,av}<0.030$, $0.6<cn1<1.0$, $0.015<cn2<0.030$, $0.2<cn3<0.8$

Crop	Parameters				Interc.	Slope	r^2	SEE	
	$g_{lm,av}$	cn1	cn2	cn3				g m^{-2} h^{-1}	%
Tomato '86	0.025	1.0	0.034	0.74	-70.	1.29	0.67	56	41
Tomato '90	0.030	0.99	0.036	0.80	-44.	1.19	0.74	41	39
Sw. pepper	0.029	1.0	0.015	0.80	-32.	1.04	0.82	74	34
Cucumber	0.030	0.91	0.026	0.58	-46.	1.28	0.65	36	38

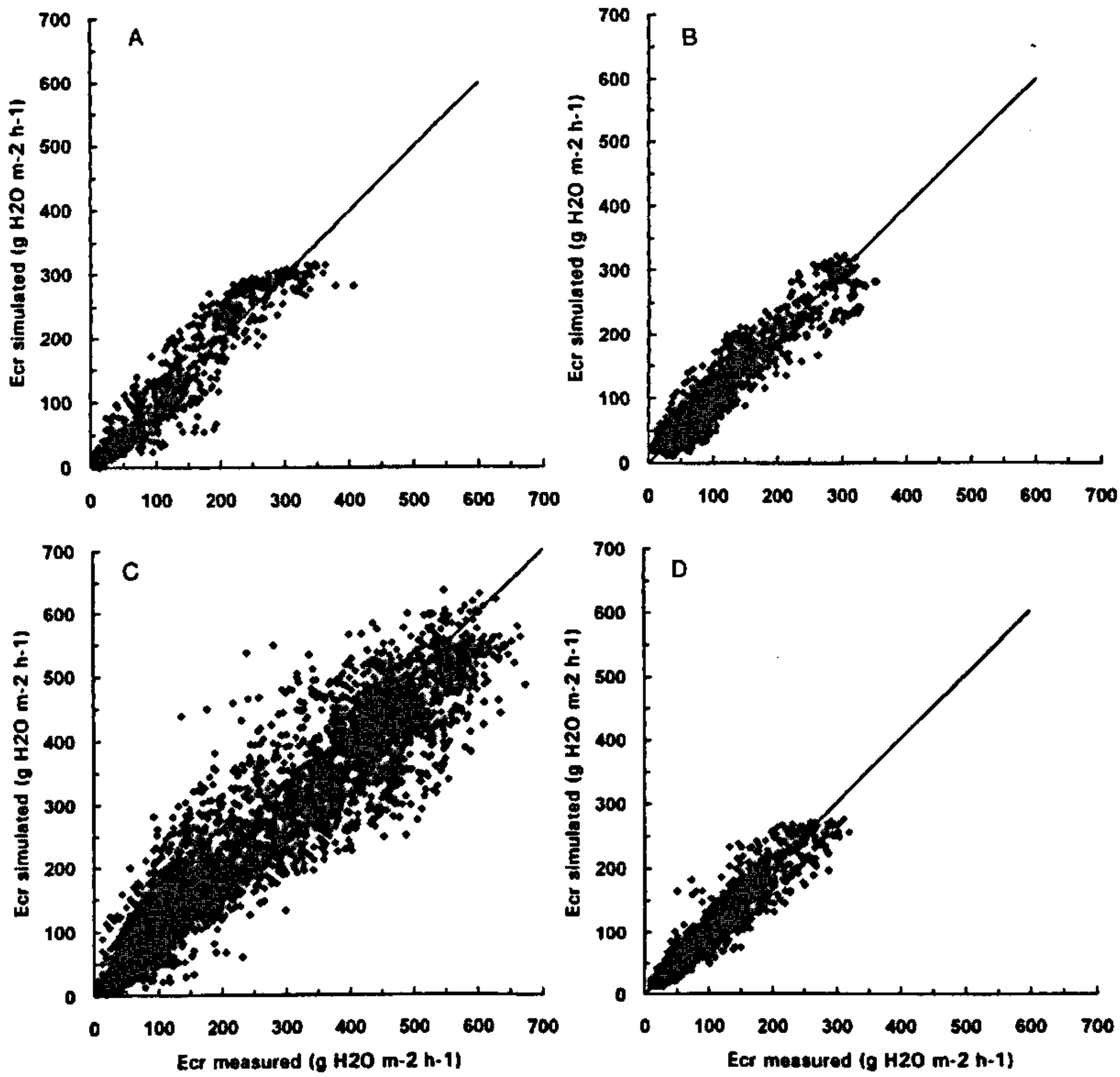


Figure 4.12 Regression of fitted crop transpiration on measured crop transpiration, for 10 minute intervals. Fitted crop transpiration was obtained using the negative-exponential submodel for stomatal conductance. **A** tomato '86, **B** tomato '90, **C** sweet pepper, and **D** cucumber.

4.6. General discussion

The results of applying published parameter values in the multilayer canopy model were not very good. This may have been caused by other non-stomatal factors, like an erroneous estimate of the boundary layer conductance (g_b), or by not taking into account effects of the water content on stomatal behaviour. However, the results also indicate that one must be cautious in applying stomatal conductance parameters and models from elsewhere. The results of testing and parameter fitting indicate that stomatal characteristics (at least the maximal conductance, g_{smax}) can vary quite a deal between crops and experiments and/or season; in present datasets these three factors could not be separated. Another factor that could reduce the potential use of parameters could be the type of leaves on which measurements were done (e.g. leaves in the top of the canopy, or all over the canopy). As an illustration, in Figure 4.13 are shown the stomatal responses to PAR, as found by the parameter fitting and as published in the literature (Bakker, 1991; Nederhoff & De Graaf, 1993; and Nederhoff *et al.*, 1992).

In general, high values of r^2 were obtained for all models when fitting in the four experiments, with the highest values in the sweet pepper experiment. In this experiment global radiation intensities and transpiration rates were also highest, so that in this experiment accurateness of the estimation of stomatal responses had least effect on overall goodness of fit.

All the stomatal submodels gave the same global picture of the light response characteristics of the crops. The tomato crops had relatively low maximal conductances as indicated by a low ratio E_{cr}/E_{wet}, a low value of m, a low C_i/C_s-ratio, and a low value of g_{smax}. Note that overestimation of the photosynthetic capacity of a crop will result in underestimation of the value of m and of C_i/C_s, and consequently will overestimate the stomatal limitation of photosynthesis.

From results of the fits with the C_i/C_s-model and the descriptive stomatal conductance models it seemed that stomatal sensitivity to leaf-air VPD was about equal in all experiments. This is in accordance with the results from Bakker (1991), who found that tomato, cucumber and sweet pepper had equal stomatal sensitivity to VPD. Note that, at least in present experiments, the response to leaf-air VPD found here by fitting could have partly substituted a response to plant internal water status. High air-VPD normally coincides with high transpiration rates and a lower internal plant water content. The latter factor could induce stomatal closure. This response was not included in the model.

All models tended to give too high value of maximal conductance in the sweet pepper experiment. This was apparently caused by the fact that the models tended to underestimated transpiration. Underestimation occurred especially in the last period of the sweet pepper experiment (period end of July, beginning of August). In this period the tops of the plant were growing close to or were pressed against the glass of the greenhouse cover. Conditions experienced by the tops of the plants could have been different from what has been measured. A second cause could be that the initial response of the stomatal models, especially the model of Ball *et al* and the Ci/Cs-model, to PAR at low levels was apparently too low. To compensate for, that the fitting routine had to adopt too high maximal stomatal conductances. Another cause for the underestimation could be that leaves in the lower part of the canopy were transpiring more than was simulated (all leaves in the canopy were simulated to have the same responses to environmental conditions). Due to the high LAI attained by the crop, the mass of leaves lower in the canopy had a large effect on canopy transpiration. E.g. variation of parameter *cd1* in the negative-exponential submodel had significant effect on canopy transpiration.

The parameters of the photosynthesis submodels could only be estimate roughly. Assumptions on the photosynthetic capacity and on how stomatal response will vary (or not) with depth in the canopy have a large influence on their value. The models need to be validated with concurrent measurements of leaf conductance and photosynthesis of leaves both in the upper and the lower part of the canopy. At present, the two transpiration-photosynthesis models can only roughly estimate the degree of limitation of photosynthesis by transpiration.

The inclusion of the row effect had a negligible effect on goodness of fit. Probably the most important reason for this was the fact that the period of time in which the crops had a marked row structure formed a relatively small fraction of the whole periods covered by the datasets.

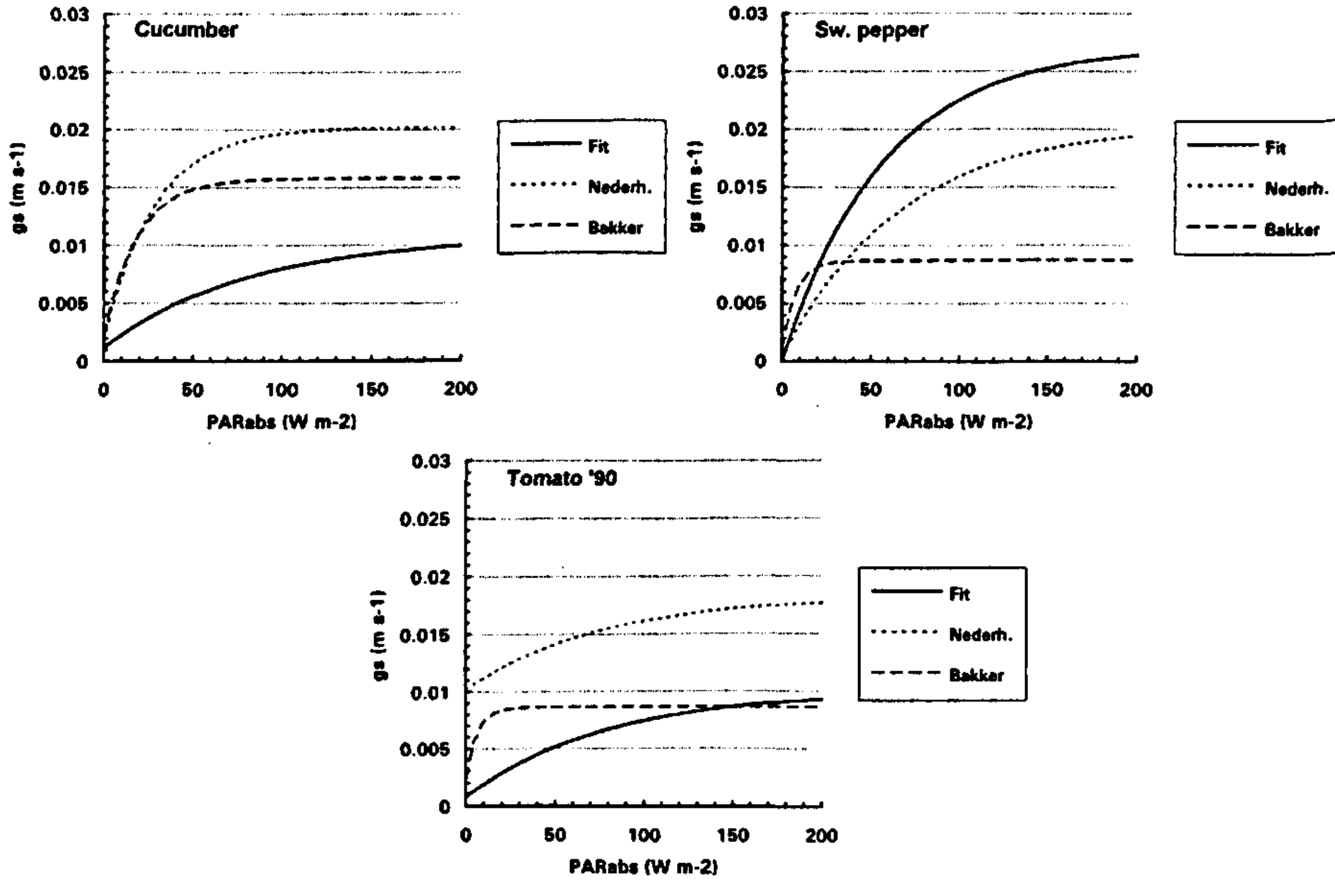


Figure 4.13 Comparison of the responses of stomatal conductance to absorbed PAR using the negative-exponential response function, when parameterized according to the fit obtained here, according to Bakker (1991) and to Nederhoff *et al.* (1992) (sweet pepper) and Nederhoff and De Graaf (1993) (cucumber and tomato). A leaf-air VPD of 0.5 was assumed.

4.7. Conclusions

In the modelling of canopy transpiration focus was on stomatal response. It appears that stomatal responses are quite variable under various circumstances, so that caution must be taken when using models and/or parameters for conditions different from those at which they were obtained. Predicting stomatal behaviour becomes increasingly important at lower radiation levels.

A reasonable fit was obtained with the both the photosynthesis based stomatal conductance models and with the descriptive stomatal conductance models. The predictive power of these models is not known as the experiments on which the fitting was performed covered a limited period. The models need to be further validated with measurements.

For prediction of transpiration rates use of the negative-exponential submodel of stomatal conductance is best suited. The parameters for the stomatal response can be compared with literature data, or can be based on literature data. The simplicity of the response function enables the total canopy transpiration model to be easily included in other models on a higher integration level. It has also a high execution speed.

The photosynthesis based submodels can be used in more explanative models aimed at analysing experiments but need better parameterization. These submodels are rather complex, because of the iteration procedure and the calculations of conditions at the leaf surface. At present not enough insight exists in their predictive power on the relation between water status and photosynthesis, as no data were available on the actual photosynthetic characteristics of the crops. More experimental data are needed before they can be included in climate control algorithms that need to estimate accurately the interaction between water status and photosynthesis.

4.8. References

Aphalo, P.J. & P.G. Jarvis, 1991.
Do stomata respond to relative humidity? Plant Cell and Environment 14: 127-132.

Bakker, J.C., 1991.
Leaf conductance of four glasshouse vegetable crops as affected by air humidity. Agricultural and Forest Meteorology 55: 22-36.

Ball, J.T., Woodrow, I.E. & J.A. Berry, 1987.
A model predicting stomatal conductance and its contribution to the control of photosynthesis under different environmental conditions. In: J. Bigges (ed.), Progress in photosynthesis research, Vol. IV.5, Martinus Nijhoff Publishers, Dordrecht, 221-224.

Bot, G.P.A., 1983.
Greenhouse climate; from physical processes to a dynamic model. Dissertation, Agricultural University, Wageningen, 240 pp.

Caemmerer, S. von & G.D. Farquhar, 1981.
Some relationships between the biochemistry of photosynthesis and the gas exchange of leaves. Planta 153: 376-387.

Collatz, G.J., J.T. Ball, C. Grivet & J.A. Berry, 1991.
Physiological and environmental regulation of stomatal conductance, photosynthesis and transpiration: a model that includes a laminar boundary layer. Agricultural and Forest Meteorology 54: 107-136.

Dehne, K., 1984.
Diffuse solar radiation measured by the shade ring method improved by a correction formula. In: On instrumentation and observing methods. Papers for TECEMO, Noordwijkerhout. WMO Report 15: 263-267.

Farquhar, G.D., S. von Caemmerer & J. A. Berry, 1980.
A biochemical model of photosynthetic CO_2 assimilation in leaves of C3 species. Planta 149: 78-90.

Gijzen, H., 1992
Simulation of photosynthesis and dry matter production of greenhouse crops. CABO-DLO/TPE Simulation Report nr. 28. 69+49 pp.

Goudriaan, J.,1989.
Collegediktaat simulatie gewasgroei. Vakgroep Theoretische Produktie-Ecologie, Landbouwuniversiteit Wageningen.

Harley, P.C. & J.D. Tenhunen, 1991.
Modelling the photosynthetic response of C3 leaves to environmental factors. In: K.J. Boote & R.S. Loomis (eds). Modelling crop photosynthesis: from biochemistry to canopy. CSSA special publication no. 19. Madison, Wisconsin, USA. pp. 17-39.
Harley, P.C., R.B. Thomas, J.F. Reynolds & B.R. Strain, 1992.
Modelling photosynthesis of cotton grown in elevated CO_2. Plant, Cell and Environment 15: 271-282.
Jolliet, O., 1993.
Modelling of water uptake, transpiration and humidity in greenhouses, and their effects on crops. Acta Horticulturae 328: 69-78.
Jones, H.G., 1983.
Plants and microclimate. Cambridge University Press, Cambridge. 323 pp.
Kirschbaum, M.U.F & G.D. Farquhar, 1984.
Temperature dependence of whole-leaf photosynthesis in *Eucalyptus pauciflora* Sieb. ex Spreng. Australian Journal of Plant Physiology 11: 519-538.
Leunig, R., 1990.
Modelling stomatal behaviour and photosynthesis of *Eucalyptus grandis*. Australian Journal of Plant Physiology 17: 159-175.
Marcelis, L.F.M., 1989.
Simulation of plant-water relations and photosynthesis of greenhouse crops. Scientia Horticulturae 41: 9-18.
Monteith, J.L. & M.H. Unsworth, 1990.
Principles of environmental physics. Second Edition. Edward Arnold, London, 291 pp.
Morison, J.I.L. & R.M. Gifford, 1983.
Stomatal sensitivity to carbon dioxide and humidity: a comparison of two C3 and two C4 grass species. Plant Physiology 71: 789-796.
Nederhoff, E.M. & R. de Graaf, 1993.
Effects of CO_2 on leaf conductance and canopy transpiration rate of greenhouse grown cucumber and tomato. Journal of Horticultural Science 68: 925-937.
Nederhoff, E.M., A.A. Rijsdijk, R. de Graaf, 1992.
Leaf conductance and rate of crop transpiration of greenhouse grown sweet pepper (*Capsicum annuum* L.) as affected by carbon dioxide. Scientia Horticulturae 52: 283-301.
Spitters, C.J., 1986.
Separating the diffuse and direct components of global radiation and its implication for modelling canopy photosynthesis. Part II. Calculation of canopy photosynthesis. Agricultural and Forest Meteorology 28: 231-242.
Stanghellini, C., 1985.
Transpiration and temperature of greenhouse crops, in relation to internal and external resistance's. Acta Horticulturae 174: 87-95.
Stanghellini, C., 1987.
Transpiration of greenhouse crops. An aid to climate management. Ph.D. Dissertation, Wageningen Agricultural University, Wageningen. 150 pp.
Zhang, L. & R. Lemeur, 1992.
Effect of aerodynamic resistance on energy balance and Penman-Monteith estimates of evapotranspiration in greenhouse conditions. Agricultural and Forest Meteorology 58: 209-228.

5. Some model simplifications

5.1. Introduction

Often models can be simplified to a larger or smaller degree, depending on the purpose of the model. This will be frequently the case when they are used in larger models on a higher integration level. Also very simple models could be useful when only a raw impression is needed. Here, two submodels for calculating photosynthesis were developed or adapted such that they can be used as a substitute of more complex models. In the previous chapter another simplified model was described, i.e. the big-leaf transpiration model.
The first model is a leaf photosynthesis submodel developed by Goudriaan *et al.* (1985) and Goudriaan (1989), called here the 'summary leaf photosynthesis model'. This is a simple and elegant model, and is as such already used in several crop growth models (e.g. the ECP model). However, the leaf photosynthetic response to temperature, and to a lesser extent that to CO_2, are somewhat unrealistic. The model was adapted for that. By the adaptation the so-called carboxylation resistance (cf. Gijzen ,1992) can be better parameterized based on literature data.
The second model is a big-leaf model for canopy photosynthesis. In this model the gradient in PAR absorbed by the canopy is neglected. The model is a further elaboration of an idea of Evans & Farquhar (1991).
The third model is a simple model of the dry matter production. This model can be used for quick estimation of dry matter production based on global radiation outside the greenhouse.

5.2. The summary leaf photosynthesis model

This model is described by Goudriaan *et al.* (1985) and Goudriaan (1989). In the model CO_2 from the outside air must pass the resistance's of the boundary layer, stomata and the carboxylation resistance. The latter one is a 'chemical' resistance and a measure of the potential for carboxylation. The CO_2-limited rate of net photosynthesis is calculated as

$$P_{n,c} = (C_a - \Gamma^*) / (1.37\, R_b + 1.6\, R_s + R_c) \qquad (5.1)$$

where

C_a	= the CO_2 concentration in the ambient air	(μmol mol^{-1})
Γ^*	= the CO_2 compensation point	(μmol mol^{-1})
R_c	= the carboxylation resistance	(s m^{-1})

The temperature dependency of the inverse of R_c (a 'conductance') is in current model versions rather schematically described by a triangle, as an approximation of an optimum curve. This optimum curve was replaced here by calculations based on the kinetics of the Rubisco-enzyme.

CO_2 binding by Rubisco can be described by Michaelis-Menten kinetics

$$V_c = \frac{c_p V_{cmax}}{K_m + c_p} \tag{5.2}$$

where

V_c	= the carboxylation velocity	($mg\ CO_2\ m^{-2}\ s^{-1}$)
C_p	= the CO_2 concentration in the chloroplast stroma	($mg\ CO_2\ m^{-3}$)
V_{cmax}	= the maximal carboxylation velocity	($mg\ CO_2\ m^{-2}\ s^{-1}$)
K_m	= the effective M-M constant	($mg\ CO_2\ m^{-3}$).

At low CO_2 concentration (near the value of Γ^*) this can be simplified to

$$V_c = \frac{c_p V_{cmax}}{K_m} \tag{5.3}$$

where the ratio K_m/V_{cmax} is the carboxylation resistance (Goudriaan, 1989). Thus, R_c can be calculated based upon published data on temperature dependencies of K_m and V_{cmax}. K_m is dependent on O_2 partial pressure, temperature and the M-M constants of Rubisco for carboxylation (K_c) and oxygenation (K_o) (Farquhar *et al.* (1980). By assuming K_c = 31 Pa and K_o = 15.5 kPa (Kirschbaum and Farquhar, 1984) and an O_2 partial pressure of 21 kPa, K_m was calculated to be 1300 mg CO_2 m^{-3} at 25 °C, and its Q_{10} equal to 1.7.

The second modification was with respect to the CO_2 response of the light saturated rate of gross photosynthesis, P_{gm} ($mg\ CO_2\ m^{-2}\ s^{-1}$). The dependency of P_{gm} on CO_2 was originally described by a Blackman-curve (Goudriaan, 1989)

$$P_{gm} = min\ \{P_{n,c}, P_{mm}\} + R_d \tag{5.4}$$

where

R_d	= the leaf dark respiration rate	($mg\ CO_2\ m^{-2}\ s^{-1}$)
P_{mm}	= the maximal endogenous capacity	($mg\ CO_2\ m^{-2}\ s^{-1}$)

This response was modelled based on data of field bean (Goudriaan, pers. comm., 1990). The modelled response resulted in a rather abrupt decrease of the effect of increasing CO_2 concentration on leaf photosynthesis above about 500 µmol mol^{-1}. A similar response of P_{gm}, although with less sharply bend transitions, was observed by, among others, Harley and Tenhunen (1992) in soybean. In present model the Blackman-curve was replaced by a non-rectangular hyperbola,

$$P_{gm} = \frac{P_{nc} + P_{mm} - \sqrt{(P_{nc} + P_{mm})^2 - 4\Theta\ P_{nc} P_{mm}}}{2\Theta} \tag{5.5}$$

where parameter Θ describes the degree of curvature. For Θ is 1 the Blackman-curve is obtained. Thus, by an appropriate choice of the value of Θ the fastness of saturation can be set. In Fig. 5.1 calculated leaf photosynthesis responses to temperature and CO_2 concentration are shown.

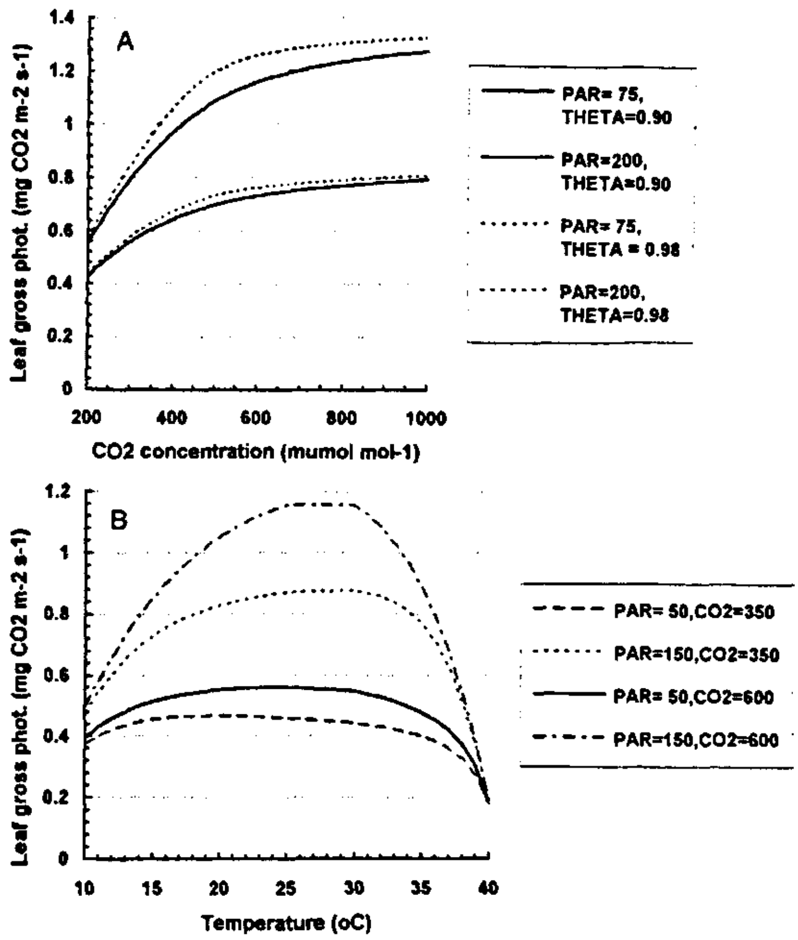


Figure 5.1 Simulated responses of leaf gross photosynthesis (mg CO_2 m^{-2} s^{-1}).

A) Response to CO_2 concentration (μmol mol^{-1}), at two PAR levels, with Θ at 0.90 and Θ at 0.98.

B) Response to temperature (°C), with Θ at 0.9.

5.3. The big-leaf photosynthesis model

This model is a further development of the idea of Evans & Farquhar (1991). These authors argued that photosynthesis-light response curve of the canopy could be obtained by swonming the light-response curves of individual leaf layers. They argued that when the photosynthetic capacity of a leaf reflects the irradiance it receives, and all leaf layers have the same curvature in the photosynthesis-light response curve, the canopy can be treated as a big-leaf, much the same way as a photosynthetic capacity of a single leaf is the sum of the differing photosynthetic capacities of chloroplasts.

Photosynthetic properties

The response of photosynthesis to absorbed PAR and to CO_2 concentration was described by

$$P_g = \frac{J(C_c - \Gamma_*)}{4(C_c + 2\Gamma_*)} \tag{5.6}$$

where

J	= electron transport rate	(µmol e⁻ m⁻² s⁻¹)
C_c	= CO_2 concentration in the chloroplasts	(µmol mol⁻¹)
Γ_*	= CO_2 compensation point in absence of dark respiration	(µmol mol⁻¹)

The expression $4\ (C_c+2\Gamma^*)$ in the denominator assumes that 4 electrons are used per CO_2 fixed (in the absence of O_2). C_c was estimated as (following Evans & Farquhar, 1991) as 0.67 times the CO_2 concentration in the ambient air

$$C_c = 0.67 * C_a \tag{5.7}$$

A non-rectangular dependence of the electron transport rate J on absorbed PAR was assumed

$$J = \frac{\alpha I + J_{max} - \sqrt{(\alpha I + J_{max})^2 - 4\Theta\ \alpha I J_{max}}}{2\Theta} \tag{5.8}$$

where J_{max} is the maximal rate of electron transport (µmol electrons $m^{-2}\ s^{-1}$), α is the electron yield of absorbed photons at low light intensities (mol mol^{-1}), and where parameter Θ describes the degree of curvature. Following Farquhar (1988), α was calculated from

$$\alpha = (1-f)/2 \tag{5.9}$$

where f is a 'loss factor', and where 2 in the denominator indicates a yield of 2 electrons per photon absorbed by the photosystems. The loss factor comprises absorption of radiation by non-photosynthetic tissues and a loss of the overall efficiency of sunlight compared with the maximal efficiency of red light. The value of loss factor f was chosen to be 0.3, making α equal to 0.35. This value of α resulted in a quantum yield (mol CO_2 per mol photons absorbed) of 0.087.

No limitation of Rubisco was assumed, and because all leaf layers were assumed to have the same value of Θ, canopy gross photosynthesis was obtained by using the J_{max} of the whole canopy in Eqn 5.8 and applying Eqn 5.6 to the whole canopy. Following Farquhar & Evans (1991) the J_{max} of the canopy was taken as the average leaf-J_{max} times the Leaf Area Index.

Following Farquhar (1988) the slightly curved temperature response of Γ_* was approximated as

$$\Gamma^* = 1.7 * T_{leaf} \tag{5.10}$$

and the optimum temperature response of J_{max} as

$$J_{max} = J_{max,25} * T_{leaf} / 25 \tag{5.11}$$

where T_{leaf} is the canopy temperature, and $J_{max,25}$ the value of J_{max} at 25 °C leaf temperature. A base temperature of 0 °C was assumed for J_{max}; the decline in J_{max} at temperatures higher than about 30 °C was neglected here. Leaf temperature was assumed to be equal to air temperature.

PAR absorption

Absorption of PAR by the big-leaf canopy was calculated from the calculation of the fluxes of absorbed diffuse and direct PAR as done by Spitters (1986). Total absorbed PAR was calculated as

$$Q_{p,dif,abs} = Q_{p,dif} * (1-\rho_{dif}) * (1-exp(-K_{dif}*LAI)) + Q_{p,r,abs} \quad (5.12)$$

$$Q_{p,dirt,abs} = Q_{p,dir} * (1-\sigma) * (1-exp(-K_{dirbl}*LAI)) \quad (5.13)$$

$$Q_{p,dir,abs} = Q_{p,dir} * (1-\rho_{dir}) * (1-exp(-K_{dir}*LAI)) \quad (5.14)$$

where

$Q_{p,dif,abs}$	= total absorbed diffuse PAR	($J\ m^{-2}\ s^{-1}$)
$Q_{p,dirt,abs}$	= absorbed direct PAR (incl. secondary diffuse)	($J\ m^{-2}\ s^{-1}$)
$Q_{p,dir,abs}$	= absorbed direct PAR (not scattered)	($J\ m^{-2}\ s^{-1}$)
$Q_{p,r,abs}$	= absorbed ground reflected PAR	($J\ m^{-2}\ s^{-1}$)
K_{dif}	= extinction coefficient for diffuse PAR	(-)
K_{dir}	= extinction coefficient for direct PAR	(-)
K_{dirbl}	= extinction coeff. for direct PAR and black leaves	(-)
σ	= scattering coefficient	(-)
ρ_{dif}	= reflection coefficient of canopy for diffuse PAR	(-)
ρ_{dir}	= reflection coefficient of canopy for direct PAR	(-)

The ground reflected PAR was originating from both diffuse PAR and direct PAR being diffused upon reflection

$$Q_{p,r,abs} = \rho_{gr} * (Q_{p,dif} * (1-\rho_{dif}) * exp(-K_{dif}*LAI) + Q_{p,dir} * (1-\rho_{dir}) * exp(-K_{dir}*LAI)) \quad (5.15)$$

where ρ_{gr} is the reflection coefficient of the ground surface.

The fraction sunlit leaf area (f_{sl}) was calculated as

$$f_{sl} = 1 / k_{dirbl} * (1 - exp(-K_{dirbl} * LAI)) \quad (5.16)$$

The flux PAR absorbed by the shaded part of the big-leaf ($Q_{p,sh}$, $J\ m^{-2}\ s^{-1}$) consists of both diffuse PAR and the diffused direct PAR

$$Q_{p,sh} = Q_{p,dif,abs} + (Q_{p,dirt,abs} - Q_{p,dir,abs}) \quad (5.17)$$

The flux PAR absorbed by the sunlit part of the big-leaf ($Q_{p,sl}$, J m^{-2} s^{-1}) consists of both the absorbed total diffuse PAR and the absorbed non-scattering direct flux PAR

$$Q_{p,sl} = Q_{p,sh} + (1-\sigma) * k_{dirbl} * Q_{p,dir} \tag{5.18}$$

Modelled responses and comparison with the multilayer canopy.

The calculated responses were compared with calculated responses of a multilayered canopy model. In the multilayer model leaf photosynthesis was calculated according to Farquhar et al. (1980), with the rate of electron transport and the electron transport limited rate of carboxylation being calculated according to Eqns 5.8 and 5.6. PAR extinction and absorption were calculated according to Spitters (1986), consequently the same total diffuse and direct PAR absorption were calculated as in the big-leaf canopy model.

Calculated canopy gross photosynthesis responses (P_{gc}) to incident PAR and CO_2 concentration were quite similar (Fig. 5.2). The big-leaf model tended to overestimate P_{gc}, compared with the multilayer model, at higher PAR levels. The overestimation was somewhat higher at lower fractions diffuse. Apparently, the neglection of the unequal distribution over the canopy of the absorbed direct PAR flux was the major cause for the discrepancy. However, the overestimation was not considered serious, as PAR levels in Dutch greenhouses are for the major part below 1000 μmol m^{-2} s^{-1}. It is concluded that the big-leaf canopy model is sufficient accurate for use as a submodel in models on a higher integration level that need less detailed calculations on canopy photosynthesis.

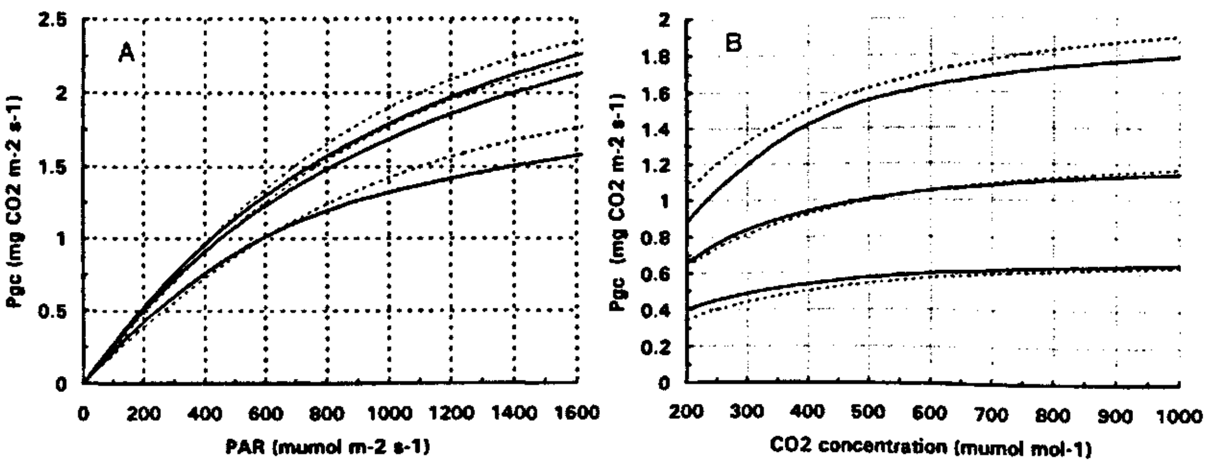


Figure 5.2 Responses of canopy gross photosynthesis (P_{gc}, mg CO_2 m^{-2} s^{-1}) to incident PAR (μmol m^{-2} s^{-1}) calculated with the big-leaf canopy model for crop photosynthesis and with a multilayer canopy photosynthesis model. See text for further explanation.
LAI at 3, scattering coefficient 0.15, K_{dif} at 0.74, leaf temperature 22 °C, solar elevation 45°, fraction PAR diffuse at 0.5, zero ground reflection. A spherical leaf angle distribution was assumed. Solid lines: simulations with the multilayered canopy model, dashed lines: simulations with the big-leaf model.
(Left) Response to incident PAR (μmol m^{-2} s^{-1}) at CO_2 concentrations 350, 700 and 1000 μmol mol^{-1}.
(Right) Response to CO_2 concentration (μmol mol^{-1}) at incident PAR levels 250, 500 and 1000 μmol photons m^{-2} s^{-1}.

5.4. A simple model for dry matter production

A simple model was developed of dry matter production based on global radiation outside the greenhouse. The core of the model is a relation between net assimilate production by a standard crop (g sugars (CH_2O) per m^2 per day) and global radiation outside the greenhouse. Then by taking an appropriate factor for converting assimilates to dry matter, total dry matter production can be estimated. And furthermore, when the partitioning of dry matter to fruits, and fruit dry matter content are known, fruit fresh weight production can be estimated.

The relation between net assimilate production and global radiation outside the greenhouse was developed by calculating the average rate of net assimilate production (i.e. the daily amount of sugars left over after maintenance costs have been subtracted) for a standard year. This year is the 'select'-year, which contains selected months from weather recorded at De Bilt, from the period 1971-1980, with hourly records of global and diffuse global radiation (Breuer & van de Braak, 1989). (cf. Gijzen, 1992). The canopy photosynthesis model was similar to the multilayer model described in the previous chapter (Ch. 5.3). A LAI of 3 was assumed, K_{dif} at 0.74, J_{max} and V_{cmax} at 250 µmol m^{-2} s^{-1} and 125 µmol CO_2 m^{-2} s^{-1}, respectively. CO_2 concentration was assumed to be 350 µmol mol^{-1}, maximal daytime temperature 23 °C (a sinusoidal course at daytime was assumed), and nighttime temperature 18 °C. Maintenance costs were taken at 8 g CH_2O m^{-2} d^{-1} at 25 °C; these costs are a gross estimate of the maintenance costs for crops like cucumber and tomato. (A full grown year-round sweet pepper crop can have a much higher standing biomass weight then the two former crops, and consequently may have higher maintenance costs.). The characteristics of an existing with high diffuse radiation transmissivity were used (75 %, the 'light' greenhouse in Gijzen, 1992); both diffuse and direct PAR transmissions were multiplied by 70/75 to obtain a more representative greenhouse with 70 % average PAR transmissivity. Ground reflection was assumed at 0.25.

The relation of the calculated rate of net assimilate production (P_{nd}, g CH_2O m^{-2} d^{-1}) to the daily global radiation outside the greenhouse (S_d, MJ m^{-2} d^{-1}) is shown in Fig. 5.3. P_{nd} has rather low values for low values of S_d, because the same maintenance costs have been assumed throughout the year. In reality these costs will be lower in the beginning of the year because the crop is not yet full grown at that time; also at the end of the year these costs will also be lower, as at short daylengths the metabolic activity of the crop will be lower.

A simple relation between P_{nd} and S_d may be obtained when the values of P_{nd} at low S_d are neglected (they are presumably somewhat higher), and a line is fitted through the points, while forcing it through the origin. A line fitted by eye has a slope of about 1.4 g CH_2O per MJ global radiation:

$$P_{nd} = 1.4 * Sd \qquad (5.19)$$

To estimate P_{nd} for other greenhouses one can multiply it by the ratio of its diffuse transmissivity to the diffuse transmissivity of the greenhouse used here. This is the same as applying the 1 % light-rule ("1 % more light is 1 % more production"). It was also calculated by Gijzen (1992) that dry matter production summed over the year followed this rule. It is difficult to give a simple correction factor for CO_2 concentrations other than 350 µmol mol^{-1}. The factor depends on the time course of CO_2 concentration over the day and longer periods, due to interaction with the light level. 15 to 30 % higher rates of dry matter production may be obtained at a constant CO_2 concentration level of 700 µmol mol^{-1}.

Rates of dry matter productions (dW/dt, g DM m^{-2} d^{-1}) of cucumber, tomato and sweet pepper can be calculated by dividing P_{nd} by the respective assimilate requirements, e.g. by values of about 1.3, 1.25 and 1.35, respectively (see Chapter 3). This would give rates of about 1.1, 1.1 and 1 g DM per MJ global radiation outside the greenhouse.

Estimates of the fruit fresh weight productions can be done from estimates of the partitioning of dry matter to the fruits, and from the fruits fresh weights. Here, partitioning of dry matter to fruits for cucumber, tomato and sweet pepper is estimated to be 0.7, 0.7 and 0.6, respectively (based on the experiments referred to in Chapter 3). Average fruit dry matter contents are estimated to be 2.8, 5.5 and 8 %, respectively. Then, for the standard greenhouse, the following gross estimates are obtained of the fruit fresh weight productions per MJ global radiation outside the greenhouse: for cucumber 1.4 / 1.3 * 0.7 * 100 / 2.8 = 27 g per MJ, for tomato 1.4 / 1.25 * 0.7 * 100 / 5.5 = 14 g per MJ, and for sweet pepper 1.4 / 1.35 * 0.6 * 100 / 8 = 8 g per MJ.

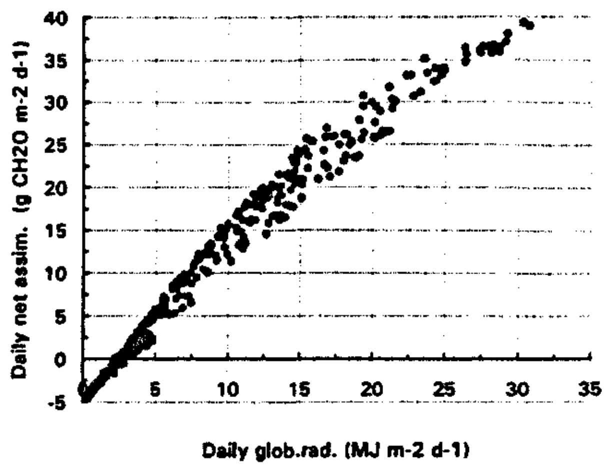


Figure 5.3 The relation between calculated daily net assimilate production (P_{nd}, g CH_2O m^{-2} d^{-1}) and daily global radiation (S_d, MJ m^{-2} d^{-1}) outside a greenhouse with 70 % diffuse PAR transmissivity, for the 'select-year'. See text for other conditions.

5.5. References

Boote, K.J, & R.S. Loomis, 1991.
The prediction of canopy assimilation. In: K.J. Boote & R.S. Loomis (eds). Modeling crop photosynthesis: from biochemistry to canopy. CSSA special publication no. 19. Madison, Wisconsin, USA, 109-140.

Breuer, J.J.G. & N.J. van de Braak, 1989.
Reference year for Dutch greenhouses. Acta Hort. 248: 101-108.

Evans, J.R, & G.D. Farquhar, 1991.
Modelling canopy photosynthesis from the biochemistry of the C3 chloroplast. In: K.J. Boote & R.S. Loomis (eds). Modeling crop photosynthesis: from biochemistry to canopy. CSSA special publication no. 19. Madison, Wisconsin, USA, 1-15.

Farquhar, G.D., 1988. Models relating subcellular effects of temperature to whole plant responses. In: S.P. Long & F.I. Woodward (eds). Plants and temperature. The Company of Biologists Ltd., 395-409.

Farquhar, G.D., S. von Caemmerer & J. A. Berry, 1980.
A biochemical model of photosynthetic CO_2 assimilation in leaves of C_3 species. Planta 149: 78-90.

Goudriaan, J.,1989.
Collegediktaat simulatie gewasgroei. Vakgroep Theoretische Produktie-Ecologie, Landbouwuniversiteit Wageningen.

Goudriaan, J., H.H. van Laar, H. van Keulen & W. Louwerse, 1985.
Photosynthesis, CO_2 and plant production. In: W. Day and R.K. Atkin (eds). Wheat growth and modeling. NATO ASI Series, Serie A: Life Sciences, Vol. 86, 107-122.

Gijzen, H., 1992.
Simulation of photosynthesis and dry matter production of greenhouse crops. CABO-DLO/TPE Simulation Report nr. 28, 69+49 pp.

Kirschbaum, M.U.F. & G.D. Farquhar, 1984.
Temperature dependence of whole-leaf photosynthesis in *Eucalyptus pauciflora Sieb. ex Spreng.*. Aust. J. Plant Physiol. 11: 519-538

6. Note on the programs and their listings

The source codes of both the programs and routines given in the listings and other routines not listed can be obtained upon request. Most of the auxiliary routines (e.g. for reading and writing to and from files and screen) are not listed.

Model INTKAM (Appendix III)

Model INTKAM is the integral model for calculating crop photosynthesis, transpiration, water uptake and dry matter production. Here the listing of the main program is given plus the main I/O routine. The canopy transpiration routine CANOPF is included. In this subroutine stomatal conductance (subroutine LFTRAN) is determined directly by environmental conditions, and is not dependent on leaf photosynthesis. Leaf photosynthesis is calculated according to the Farquhar *et al* model (subroutine FARPHOT, see Appendix *VII*). The routines for water uptake and plant water content were taken from Marcelis (1989). Note that in the model plant water content has no effect on stomatal conductance. The listings of some general simulation routines are found in Appendix XII.

Model ASTRAKAM and additional routines (Appendix IV)

Here the listing is given of the multilayered canopy model for both transpiration and crop photosynthesis. The canopy transpiration routine CANOP2 is similar to CANOPF and is not included in the listing. Leaf photosynthesis is calculated with the summary model of leaf photosynthesis (subroutine LPHOT, see Appendix *VI*).

Listing of photosynthesis-based leaf transpiration routines (Appendix V)

The listing is given of the leaf transpiration routines in which stomatal conductance is calculated as dependent on leaf photosynthesis. The leaf photosynthesis routines are based on the model of Farquhar *et al.*. Routines FARPHOT2 and FARPHOT3 are derived from and quite similar to subroutine FARPHOT (See Appendix *VII*), and are not given in the listing.

Subroutine LPHOT (Appendix VI)

The listing is given of the summary leaf photosynthesis model, based on Goudriaan et al. (1985). Function LPHCUR as described by Gijzen (1992) can be updated following this subroutine.

Subroutines containing the model of Farquhar et al (Appendix VII)

Here the listing is given of subroutine FARPHOT, and of routines called by FARPHOT. Subroutine FARPHOT is used in model INTKAM.

Subroutine BIGLTR (Appendix VIII)

Subroutine BIGLTR is the big-leaf model for canopy transpiration.

Subroutine BIGLPH (Appendix IX)

Subroutine BIGLPH is the big-leaf model for canopy photosynthesis.

Listing of the points-model for greenhouse cover shading of a row crop (Appendix X)

Listing of distributed direct radiation transmission model. Account is taken of shades received by the ridge-gutter system. For various points within a plant stand it is calculated whether shade is received or not.

Listing of the area-model for greenhouse cover shading of a plant stand. (Appendix XI)

Listing of distributed direct radiation transmission model. Account is taken of shades received by both the ridge-gutter system and the beam system. The volume of the crop stand that is receiving shade of a construction element is a measure of the transmission by that particular element.

General simulation routines (Appendix XII)

Here the listings are given of some the general simulation routines called by the routines in the previous listings.

Explanation of variables and parameters (Appendix XIII)

The meaning and dimension of the main variables occurring in the models are listed here.

Appendix I:
Accounting for the row effect

In the following procedure the vertical profile of absorbed direct PAR was calculated in a row, and 'transplanted' in the standard subroutine (subroutine CANOP) that calculates the transpiration and photosynthesis of a closed canopy. The absorption profile of NIR in the row was not taken into account, as model development on this aspect was not advanced far enough yet.

The calculation procedure consisted of the following steps:

1) a rectangular grid of 40 points horizontal times 20 point vertical was projected in a row with rectangular transsection;
2) at each point (i,j) the fraction sunlit leaf area (F_{sl}) and the intensity of absorbed total direct PAR ($I_{p,dirt,abs}$) were calculated following the method as done in the model for row crop photosynthesis described by Gijzen & Goudriaan (1989);
3) F_{sl} and $I_{p,dirt,abs}$ were averaged over each horizontal array of 40 grid points (i=1,40);

$$I_{p,dirt,abs,av} = \frac{1}{40} \sum_{i=1}^{i=40} I_{p,dirt,abs} \tag{I.1}$$

$$F_{sl,av} = \frac{1}{40} \sum_{i=1}^{i=40} F_{sl} \tag{I.2}$$

for all 20 horizontal arrays (j=1,20), and values were put into tables;

4) the values in these three tables were used in subroutine CANOP, were they substituted the standard calculation of the extinction in a closed canopy of F_{sl} and the flux $I_{p,dirt,abs}$.

Appendix II: Calculation of distributed direct radiation transmission

Distributed direct radiation transmission

In the model of Bot (1983) one single value for the transmission of the greenhouse cover for direct radiation is calculated, that is the average for the whole greenhouse area (i.e. in zero dimensions). In his model, direct radiation transmission, $T_{r,dir}$, is calculated as the combined effects of three components of the construction and of the glass panes, as dependent on the solar position and the azimuth of the greenhouse

$$T_{r,dir} = T_{r,ridgut}\, T_{r,bar}\, T_{r,beam}\, T_{r,glas} \qquad (II.1)$$

where

$T_{r,ridgut}$	= direct radiation transmission of the ridge-gutter system	(-)
$T_{r,bar}$	= direct radiation transmission of the bars	(-)
$T_{r,beam}$	= direct radiation transmission of the beams	(-)
$T_{r,glas}$	= direct radiation transmission of the glass panes	(-)

The ridge-gutter system and the beam system are major causes of spatial variation of direct radiation intensity in the greenhouse. The effect of shadows casts by these systems on transpiration of a crop stand was calculated with two models.

The first model, the points-model, was a two-dimensional model for the transmission of direct radiation of the ridge-gutter system. For various points in the crop stand it was calculated whether shadow was received by the ridge-gutter system or not.

In the second model, the area-model, the areas of the shade casts by the ridge-gutter system and the beam system on the crop stand were calculated. The transmissions of other components were averaged over the whole greenhouse area, i.e. were still zero-dimensional.
Note that only the points-model can be used in conjunction with calculation of the row effects on radiation absorption (see Appendix I).

The points-model

Geometrics

In this model spatial variation in direct radiation intensity is considered in the XZ-plane, i.e. the plane perpendicular to the ridges and gutters (Fig. II.1). The origin of the coordinate system is located directly under the middle of gutter nr. 0.
In the first step of the calculation procedure, the shades of gutter and ridges were projected on the horizontal projection line directly underneath the gutters, whenever the solar position had changed. Algorithms for calculating the projected widths could be taken from the model of Bot.

In the second step, the amount of direct radiation received by any object was calculated by choosing a number of points on this object and projecting them on the projection line, in the direction of the solar beam. For each projection it was determined whether it fell on a shaded part ($T_{r,ridgut} = 0$) or sunlit part ($T_{r,ridgut} = 1$) of the projection line.
Thus, by averaging all the values of $T_{r,ridgut}$, the total amount of direct radiation received by the object can be calculated.

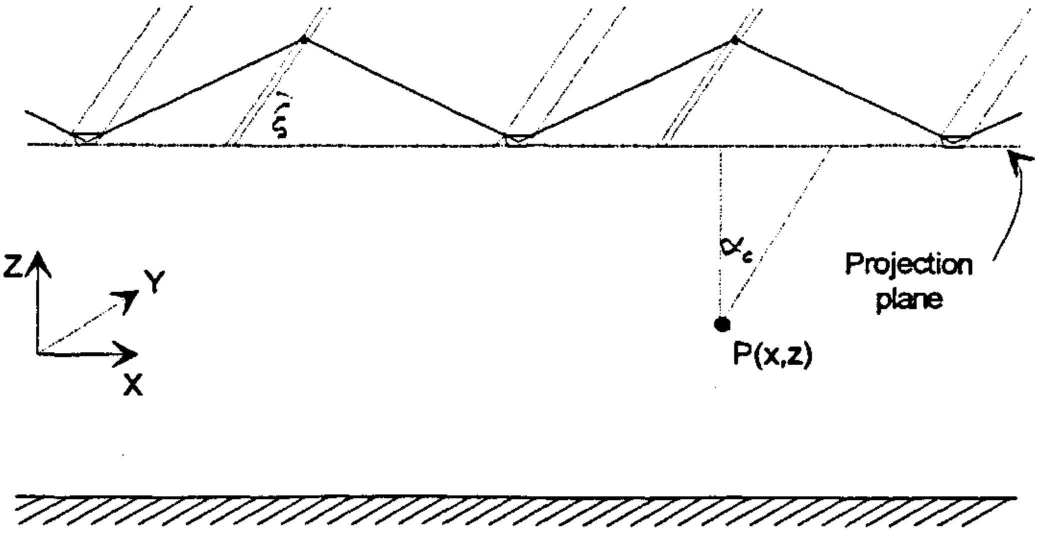


Figure II.1 Scheme of the calculation of distributed greenhouse cover radiation transmission. Shades of the greenhouse cover are projected onto the projection plane (into the direction of converted sun elevation ξ. By projection of P on the projection plane it is determined whether any point P in the XZ-plane is receiving shade.

In this procedure any more greenhouse element running parallel to the Y-axis can be added in a simple way. Furthermore, the model is easily extended to three dimensions, so that also shading patterns of greenhouse elements running parallel to the X-axis (e.g. the beams), or of three dimensional objects can be described.

Radiation absorption by the canopy

The intensity of the direct beam in the sunlit patches is:

$$I_{dir,rg} = I_{dir,o} * T_{r,bar} * T_{r,beam} * T_{r,glas} \qquad (II.2)$$

where

$I_{dir,o}$ = direct radiation outside greenhouse ($J\ m^{-2}\ s^{-1}$)

and the fraction of the ground or crop area that is receiving this flux is equal to $T_{r,ridgut}$.

The effect of the distributed direct radiation transmission on the direct radiation absorption by part of the crop was calculated with the following steps, and along the same lines as the procedure for the row effect, described in Appendix I:

1) a rectangular grid of 40 points horizontal times 20 point vertical was projected on a given part of the crop (plant stand) with rectangular transsection;
2) at each point (i,j) the transmission of the ridge-gutter system ($T_{r,ridgut}$) was determined by projecting the point onto the projection line;

3) at each point (i,j) the fraction sunlit leaf area (F_{sl}) and the intensity of absorbed total direct PAR ($I'_{p,dirt,abs}$) were calculated following the method as done in the model for row crop photosynthesis described by Gijzen & Goudriaan (1989); here the prime indicates that these fluxes were calculated from direct PAR in the sunlit patches ($I_{dir,rg}$); when the plant stand under consideration was part of a closed canopy, a row crop with zero path width was assumed;
4) F'_{sl} and $I'_{p,dirt,abs}$ were averaged over each horizontal array of 40 grid points (i=1,40),

$$I'_{p,dirt,abs,av} = \frac{1}{40}\sum_{i=1}^{i=40} T_{r,ridgut}\, I'_{p,dirt,abs} \qquad (II.3)$$

$$F'_{sl,av} = \frac{1}{40}\sum_{i=1}^{i=40} T_{r,ridgut}\, F'_{sl} \qquad (II.4)$$

for all 20 horizontal arrays (j=1,20), and values were put into tables;

5) the values in these two tables were used in subroutine CANOP, were they substituted the standard calculation of the extinction in a closed canopy of F_{sl} and the flux $I_{p,dirt,abs}$.

The area-model

In this model the shades casts by the ridge-gutter system and the beam system on a crop stand are calculated. The areas of the shades were calculated for any given sun position. For this purpose the crop stand was represented as a block with height, width (in the X-direction (running perpendicular to ridge-gutter system)) and depth (in Y-direction (running parallel to ridge-gutter system)). For the ridge-gutter system the block was transsected in the XZ-plane and the area of the shades cast on the rectangle calculated (quadrangle ABCD in Fig. II.2). All the possible shade area's were summed to yield total area of the shade (S_{rg}) of the ridge-gutter system. Transmission of the ridge-gutter system was then calculated as

$$T_{r,ridgut} = 1 - S_{rg} / A_{crop,xz} \qquad (II.5)$$

where $A_{crop,xz}$ is the area of the rectangular crop transsection in the XZ-plane.

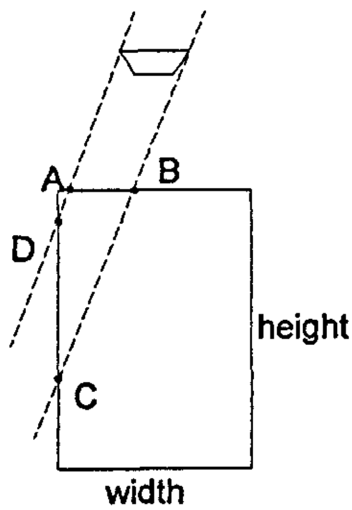


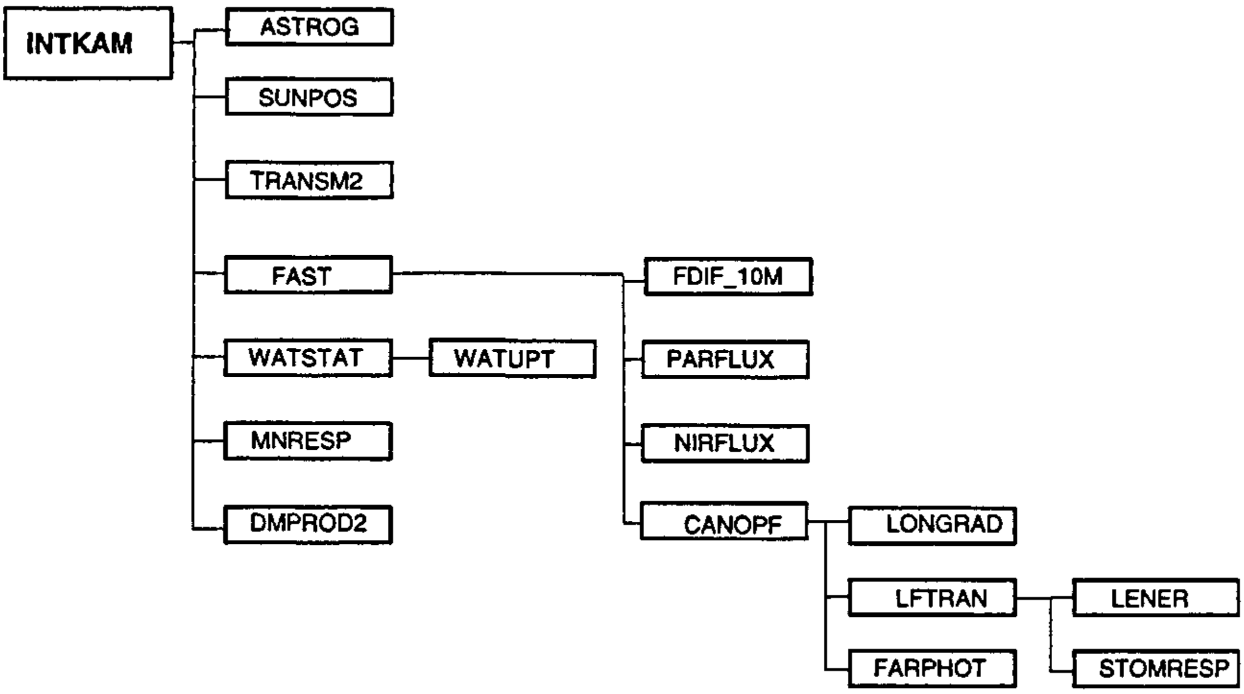
Figure II.2 Schematic representation of shade cast by a gutter on a crop stand. The crop stand is transsected in the XZ-plane. The area ABCD was calculated.

For the beam system direct transmissivity was calculated along the same lines

$$T_{r,beam} = 1 - S_{rg} / A_{crop,yz} \tag{II.6}$$

where $A_{crop,yz}$ is the area of the rectangular crop transsection in the YZ-plane. Direct transmissions of the bar system and of the glass were derived from Bot's model. Direct radiation received by the crop stand was calculated using Eqn II.1.

Appendix III:
Listing of model INTKAM



Routine	Calculation of
ASTROG	astronomical variables
SUNPOS	sun position
TRANSM2	transmissivity greenhouse cover
FAST	some fast processes
FDIF_10M	fraction diffuse in global radiation
PARFLUX	fluxes total, diffuse and direct PAR
NIRFLUX	fluxes total, diffuse and direct NIR
CANOPF	canopy transpiration and energy balance
LONGRAD	thermal radiation
LFTRAN	leaf transpiration, energy balance and stom. conductance
LENER	leaf energy balance
STOMRESP	stomatal response
FARPHOT	leaf gross photosynthesis (model Farquhar et al.)
WATSTAT	crop water content
WATUPT	water uptake of crop
MNRESP	maintenance respiration
DMPROD2	dry matter production

```
***********************************************************************
* Program: INTKAM
* Author: H. Gijzen, AB-DLO, Wageningen
* Version: 1.0
* Date: June 1994
* Purpose: Calculation of crop dry matter production and crop water balance
*
* Description:
*  Crop photosynthesis, transpiration and water uptake are in short time
*  time steps. Crop gross photosynthesis is calculated based on
*  Gijzen (1992); leaf photosynthesis is calculated based on the
*  model of Farquhar et al. (1980).
*  Calculation of dry matter production from gross assimulates is done
*  as in SUCROS87.
*  Calculation of transpiration is based on, and water uptake following
*  Marcelis (1989).
*
* Subroutines called:
*   (simulation)
*     ASTROG    - astronomical variables
*     CANOPF    - canopy transpiration and energy balance
*     DMPROD2   - dry matter production
*     FDIF_10M  - fraction diffuse in global radiation
*     MNRESP    - maintenance costs
*     NIRFLUX   - diffuse and direct NIR and UV outside greenhouse
*     PARFLUX   - diffuse and direct PAR
*     SUNPOS    - sun position
*     TRANSM2   - transmissivity greenhouse cover
*     WATSTAT   - water content of crop
*
*   (general)
*     ENVINT    - obtaining data and parameters from files and user
*     DTIMER    - timer variables
*     HTIMER2   - timer variables
*     LINTNM    - linear interpolation
*
* Input:
*     data file          (unit IUDAT)
*     data info file     (unit IUDATIF)
*     timer file         (unit IUTIM)
*     parameter file     (unit IUPAR)
*     transmissivity file (unit IUTRAN)
*
* Output:
*     file with instantaneous values of parameters      (unit IUOUT)
*     file with output of carbon balance (daily values) (unit IUOUTC)
*     file with output of water balance (daily values)  (unit IUOUTW)
*
*     Names of output files are derived from data file:
*       E.g. data file 'KOM88A.DAT'  (name maximal 6 alphanum. characters)
*      -> file name instant. values:  'KOM88A' + runstring +  '.CSV'
*      -> file name carbon balance:   'KOM88A' + runstring + 'C.CSV'
*      -> file name water balance:    'KOM88A' + runstring + 'W.CSV'
*      where 'runstring' is an alphanumeric character
*
* Comments:
*   Simulation is done for a growing season
*   Time control:
*      - two time loops are used: a day loop, and within the
*        day loop a fast loop for calculations within the day
*      - program increments in fast loop
*        time counter (DAYMIN = DAYMIN + DELTMIN)
*        time steps in DELTMIN minutes
*      - finish of simulation when end-of-file is encountered or
*        when finish time is reached
*
*   Command line arguments ENVINT:
*      EXP: string (CH*5) for experiment name
*      DAT: data file
*      PAR: parameter file
*      TIM: timer file
*      RUN: alphanumeric character for run identification
*
*     (optional)
*      FI: file with info about layout data file (default DM.FI)
***********************************************************************
```

```
      PROGRAM INTKAM
      IMPLICIT REAL(A-Z)

*     Simulation control
      INTEGER ITASKD
      LOGICAL LIGHT
      COMMON /GENCOM/ LIGHT
      LOGICAL DAYTASKS
      LOGICAL INI, TERMNL
      LOGICAL OUTPUT
      LOGICAL FILE_END
      INTEGER ITASKH
      LOGICAL OUTPUTH
      INTEGER IOPHASE

      LOGICAL command
      COMMON /IO_0/ command

      COMMON /GENCOM2/ SOLHR

      CHARACTER*5 EXPRNT

*     File I/O
      INTEGER IUOUTH
      COMMON /IO_UNIT_OUT1/ IUOUTH
      INTEGER IUOUTC
      COMMON /IO_UNIT_OUT2/ IUOUTC
      INTEGER IUOUTW
      COMMON /IO_UNIT_OUT3/ IUOUTW
      CHARACTER*40 FILOUTH, FILOUTC, FILOUTW
      COMMON /IO_NAME_OUT/ FILOUTH, FILOUTC, FILOUTW

      INTEGER IUDAT, IUDATIF, IUTIM, IUPAR, IUTRAN
      CHARACTER*40 FILTRN
      CHARACTER*40 DATAFIL, INFOFIL, PARFIL
      CHARACTER*40 TIMFIL
      COMMON /IO_UNIT_IN/ IUDAT, IUDATIF, IUTIM, IUPAR, IUTRAN
      COMMON /IO_NAME_IN/ DATAFIL, INFOFIL, TIMFIL, PARFIL, FILTRN

      INTEGER I,IF

      DIMENSION LAITB(80)
      INTEGER NLAITB

      DIMENSION FLVTB(80), FSTTB(80), FRTTB(80), FSOTB(80)
      INTEGER NFLVTB, NFSTTB, NFRTTB, NFSOTB

      DIMENSION WLVTB(80), WSTTB(80), WRTTB(80), WSOTB(80)
      INTEGER NWLVTB, NWSTTB, NWRTTB, NWSOTB

      DIMENSION TROOF_NIGHTTB(80)
      INTEGER NTROOFNTB
      DIMENSION TROOF_DAYTB(80)
      INTEGER NTROOFDTB
      LOGICAL TROOF_KNOWN, TEMPAIR_OUT_KNOWN
      INTEGER IWAR_TROOF

*     Timer variables
      INTEGER DAYMIN, DATA_DAYMIN
      INTEGER DELTMIN, OUTDELMIN
      INTEGER IDAY
      INTEGER IYEAR, STYEAR, FINYEAR, STARTDAY, FINDAY
      INTEGER OUTDELDAY
      INTEGER TMIN80, TOTDAY80

* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      DATA IUDAT, IUDATIF, IUTIM, IUPAR, IUTRAN / 11, 12, 13, 14, 15 /
      DATA IUOUTH, IUOUTW, IUOUTC / 20, 21, 22 /
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      OPEN( UNIT = 99, FILE = 'T99.O', STATUS = 'UNKNOWN' )
      PI = 3.1415926
      RADN = PI/180.

*     Daily timestep (dummy)
      DELT = 1.
* =============================================================
*     Initialization
```

```
* ==============================================================
*---- Reading of parameter values and opening data file
      ITASKD = 1
      ITASKH = 1

      IOPHASE = 1
      CALL ENVINT( IOPHASE, EXPRNT,
     &  DAYNR, SOLHR, HOUR, TMIN80, DATA_DAYMIN, DAYMIN, SINELV,
     &  IYEAR, STYEAR, FINYEAR,
     &  STARTDAY, FINDAY, OUTDELDAY, DELTMIN, OUTDELMIN,
     &  LAT, TIMCOR, SUNRISE, SUNSET, DAYL, REFGR,
     &  AZIMGR, TRDIF, TRCOR_UV,
     &  LAITB, NLAITB, KDIF, KDIFBL, SCP, SCN, PHOTREDCOF, GB, GCUT,
     &  GLRADO, CO2AIR, TEMPAIR, VPDAIR,
     &  SSPT, SSPB, TPIPE,
     &  TROOF_NIGHTTB, TROOF_DAYTB, NTROOFNTB, NTROOFDTB,
     &  TROOF_DAY, TROOF_NIGHT,  TGROUND,
     &  TEMPAIR_OUT, SCREEN,
     &  TROOF_KNOWN, TEMPAIR_OUT_KNOWN,
     &  MAINLV, MAINST, MAINRT, MAINSO, Q10MN, REFTMP,
     &  ASRQLV, ASRQST, ASRQRT, ASRQSO,
     &  FLVTB, FSTTB, FRTTB, FSOTB,
     &  NFLVTB, NFSTTB, NFRTTB, NFSOTB,
     &  WLVTB, WSTTB, WRTTB, WSOTB,
     &  NWLVTB, NWSTTB, NWRTTB, NWSOTB,
     &  WLVI, WSTI, WRTI, WSOI,
     &  WATCONI, WATCONMAX, RWATCONWI, PSIWIL, PSIROOTM,
     &  RESWAT, RIONUPT,
     &  FILE_END )

*     Reading transmissivity properties greenhouse
      CALL TRANSM2( ITASKH, IUTRAN, FILTRN,
     &              AZIMGR, AZIMS, ELEVN,
     &              TRDIF, TRCOR_UV, TRCON, TRGLAS )

*---- user interaction, opening output files, writing info in headers
      IOPHASE = 3
      CALL ENVINT( IOPHASE, EXPRNT,
     &  DAYNR, SOLHR, HOUR, TMIN80, DATA_DAYMIN, DAYMIN, SINELV,
     &  IYEAR, STYEAR, FINYEAR,
     &  STARTDAY, FINDAY, OUTDELDAY, DELTMIN, OUTDELMIN,
     &  LAT, TIMCOR, SUNRISE, SUNSET, DAYL, REFGR,
     &  AZIMGR, TRDIF, TRCOR_UV,
     &  LAITB, NLAITB, KDIF, KDIFBL, SCP, SCN, PHOTREDCOF, GB, GCUT,
     &  GLRADO, CO2AIR, TEMPAIR, VPDAIR,
     &  SSPT, SSPB, TPIPE,
     &  TROOF_NIGHTTB, TROOF_DAYTB, NTROOFNTB, NTROOFDTB,
     &  TROOF_DAY, TROOF_NIGHT,  TGROUND,
     &  TEMPAIR_OUT, SCREEN,
     &  TROOF_KNOWN, TEMPAIR_OUT_KNOWN,
     &  MAINLV, MAINST, MAINRT, MAINSO, Q10MN, REFTMP,
     &  ASRQLV, ASRQST, ASRQRT, ASRQSO,
     &  FLVTB, FSTTB, FRTTB, FSOTB,
     &  NFLVTB, NFSTTB, NFRTTB, NFSOTB,
     &  WLVTB, WSTTB, WRTTB, WSOTB,
     &  NWLVTB, NWSTTB, NWRTTB, NWSOTB,
     &  WLVI, WSTI, WRTI, WSOI,
     &  WATCONI, WATCONMAX, RWATCONWI, PSIWIL, PSIROOTM,
     &  RESWAT, RIONUPT,
     &  FILE_END )


      WRITE( IUOUTC, '(A,A,A,A,A)')
     & ' DAYNR, DGLRADO, DGLOBRAD, DPAR, DPARABS,',
     & ' DTGA, DMAINT, GTW, CWLV, CWST, CWRT, CWSO, CTWT '

      WRITE( IUOUTW, '(A,A,A,A,A)')
     & ' DAYNR, DGLRADO, DGLOBRAD, DPAR, DPARABS,',
     & ' DRADABS, DNETRAD, DNETRAD_D, DTRANSP, DTRANS_D '

*     Variable column names
      WRITE(IUOUTH,'(A,A,A,A,A,A)')
     &    ' DAYNR, Hour, GLRADO, GLOBRAD, NETRAD, RADABS, PAR, ',
     &    ' CO2air, TEMPAIR, VPDair, ',
     &    ' PGROS, TRAN_SIM, RWUPT, WATCON, GSTOT '
```

```
*---- Initial calculations
*     Conversion of degrees into radians
      LAT = LAT * RADN
      AZIMGR = AZIMGR * RADN

      INI = .TRUE.
*---- Timers
      CALL DTIMER( INI,
     &  IYEAR, STYEAR, FINYEAR, STARTDAY, FINDAY, IDAY, DAYNR, TOTDAY80,
     &  OUTDELDAY, OUTPUT, TERMNL
     & )

      CALL HTIMER2( INI,
     &  TMIN80, TOTDAY80, DELTF, DELTMIN, OUTDELMIN,
     &  DAYMIN, HOUR, OUTPUTH, DAYTASKS )

*----    Initialization

      CALL FAST(
     &  ITASKH, OUTPUTH, DELTF,
     &  DAYNR, HOUR,
     &  SOLARC, ELEVN, SINELV,
     &  TRDIF, TRCOR_UV, TRCON, TRGLAS,
     &  LAI, KDIFBL, KDIF, SCP, SCN, GB, GCUT,
     &  GLRADO, CO2AIR, TEMPAIR, VPDAIR,
     &  SSPT, SSPB, TPIPE, TROOF, TGROUND, REFGR, PHOTRED,
     &  PGROS, TRAN_SIM,
     &  GLOBRAD, NETRAD, RADABS, PAR, RWUPT, WATCON, PSIPL, GSTOT,
     &  DGLRADO, DGLOBRAD, DPAR, DPARABS, DRADABS,
     &  DTGA, DTRANS, DTRANS_D, DNETRAD, DNETRAD_D
     &  )

*     Initialization of water status
      CALL WATSTAT( ITASKH,
     &    WATCONI, WATCONMAX, RWATCONWI, PSIWIL, PSIROOTM, RESWAT,
     &    RIONUPT,
     &    DELTF, HOUR, TEMPAIR, TRAN_SIM, RWUPT, WATCON, PSIPL )


*     Initialization dry weights
      CALL DMPROD2( ITASKD, DAYNR, DELT,
     &             WLVI, WSTI, WRTI, WSOI,
     &             FLV, FST, FRT, FSO,
     &             ASRQLV, ASRQST, ASRQRT, ASRQSO,
     &             DTGA, DMAINT,
     &             GLV, GST, GRT, GSO, GTW,
     &             WLV, WST, WRT, WSO, TWT,
     &             CWLV, CWST, CWRT, CWSO, CTWT )

      INI = .FALSE.

      DO WHILE( .NOT. TERMNL )

************************************************************************
*     Daily calculations
************************************************************************

* ===================================================================
*     Integration
* ===================================================================
      ITASKD = 3

      CALL DMPROD2( ITASKD, DAYNR, DELT,
     &             WLVI, WSTI, WRTI, WSOI,
     &             FLV, FST, FRT, FSO,
     &             ASRQLV, ASRQST, ASRQRT, ASRQSO,
     &             DTGA, DMAINT,
     &             GLV, GST, GRT, GSO, GTW,
     &             WLV, WST, WRT, WSO, TWT,
     &             CWLV, CWST, CWRT, CWSO, CTWT )

* ===================================================================
*     Calculations driving variables
* ===================================================================
      ITASKD = 2

      CALL ASTROG( DAYNR, LAT, SOLARC, SINLD, COSLD, DECL,
```

```
     &              DAYL, DSINBE )

      SUNRISE = 12. - 0.5 * DAYL
      SUNSET = 12. + 0.5 * DAYL

*     Leaf Area Index
      LAI = LINTNM( 'LAITB', LAITB, NLAITB, DAYNR )

*     Weight of plant parts
      WLV = LINTNM( 'WLVTB', WLVTB, NWLVTB, DAYNR )
      WST = LINTNM( 'WSTTB', WSTTB, NWSTTB, DAYNR )
      WRT = LINTNM( 'WRTTB', WRTTB, NWRTTB, DAYNR )
      WSO = LINTNM( 'WSOTB', WSOTB, NWSOTB, DAYNR )

*     Partitioning of dry matter
      FLV = LINTNM( 'FLVTB', FLVTB, NFLVTB, DAYNR )
      FST = LINTNM( 'FSTTB', FSTTB, NFSTTB, DAYNR )
      FRT = LINTNM( 'FRTTB', FRTTB, NFRTTB, DAYNR )
      FSO = LINTNM( 'FSOTB', FSOTB, NFSOTB, DAYNR )

      TROOF_NIGHT = LINTNM( 'TROOF_NIGHTTB', TROOF_NIGHTTB,
     &                              NTROOFNTB, DAYNR )
      TROOF_DAY = LINTNM( 'TROOF_DAYTB', TROOF_DAYTB,
     &                              NTROOFDTB, DAYNR )

*     Reduction of photosynthetic capacities with height in canopy
*     i.e. when PHOTREDCOF = 0.23 then photosynthetic capacities are reduced
*     to 50% at LAI depth 3, and 25% at LAI depth 6
      PHOTRED = EXP( - PHOTREDCOF * LAI )

* ================================================================
*     Rate calculations
* ================================================================

****** Calculation within day *****************************************

*----    Resetting

      ITASKH = 5

      CALL FAST(
     &  ITASKH, OUTPUTH, DELTF,
     &  DAYNR, HOUR,
     &  SOLARC, ELEVN, SINELV,
     &  TRDIF, TRCOR_UV, TRCON, TRGLAS,
     &  LAI, KDIFBL, KDIF, SCP, SCN, GB, GCUT,
     &  GLRADO, CO2AIR, TEMPAIR, VPDAIR,
     &  SSPT, SSPB, TPIPE, TROOF, TGROUND, REFGR, PHOTRED,
     &  PGROS, TRAN_SIM,
     &  GLOBRAD, NETRAD, RADABS, PAR, RWUPT, WATCON, PSIPL, GSTOT,
     &  DGLRADO, DGLOBRAD, DPAR, DPARABS, DRADABS,
     &  DTGA, DTRANS, DTRANS_D, DNETRAD, DNETRAD_D
     &  )

      CALL MNRESP( ITASKH, DELTF, Q10MN, REFTMP,
     &           WLV, WST, WRT, WSO,
     &       MAINLV, MAINST, MAINRT, MAINSO,
     &       TEMPAIR, DMAINT )

      DO IF = 1, 100000

* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
*        Integration
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         ITASKH = 3

         CALL FAST(
     &  ITASKH, OUTPUTH, DELTF,
     &  DAYNR, HOUR,
     &  SOLARC, ELEVN, SINELV,
     &  TRDIF, TRCOR_UV, TRCON, TRGLAS,
     &  LAI, KDIFBL, KDIF, SCP, SCN, GB, GCUT,
     &  GLRADO, CO2AIR, TEMPAIR, VPDAIR,
     &  SSPT, SSPB, TPIPE, TROOF, TGROUND, REFGR, PHOTRED,
     &  PGROS, TRAN_SIM,
     &  GLOBRAD, NETRAD, RADABS, PAR, RWUPT, WATCON, PSIPL, GSTOT,
     &  DGLRADO, DGLOBRAD, DPAR, DPARABS, DRADABS,
```

```
     &  DTGA, DTRANS, DTRANS_D, DNETRAD, DNETRAD_D
     &  )

*----    Water status
         CALL WATSTAT( ITASKH,
     &    WATCONI, WATCONMAX, RWATCONWI, PSIWIL, PSIROOTM, RESWAT,
     &    RIONUPT,
     &    DELTF, HOUR, TEMPAIR, TRAN_SIM, RWUPT, WATCON, PSIPL )


*        Maintenance respiration (mg CH2O m-2 s-1)
         CALL MNRESP( ITASKH, DELTF, Q10MN, REFTMP,
     &            WLV, WST, WRT, WSO,
     &      MAINLV, MAINST, MAINRT, MAINSO,
     &      TEMPAIR, DMAINT )

* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
*        Calculation driving variables and rates
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         ITASKH = 2

*     Solar position; SINELV is sine of solar elevation,
*     AZIMS is azimuth of sun
         CALL SUNPOS (LAT, SINLD, COSLD, DECL, SOLHR,
     &                 ELEVN, AZIMS, SINELV )

*          Transmission greenhouse
           CALL TRANSM2( ITASKH, IUTRAN, FILTRN,
     &             AZIMGR, AZIMS, ELEVN,
     &             TRDIF, TRCOR_UV, TRCON, TRGLAS )

         IOPHASE = 4
         CALL ENVINT( IOPHASE, EXPRNT,
     &     DAYNR, SOLHR, HOUR, TMIN80, DATA_DAYMIN, DAYMIN, SINELV,
     &     IYEAR, STYEAR, FINYEAR,
     &     STARTDAY, FINDAY, OUTDELDAY, DELTMIN, OUTDELMIN,
     &     LAT, TIMCOR, SUNRISE, SUNSET, DAYL, REFGR,
     &     AZIMGR, TRDIF, TRCOR_UV,
     &     LAITB, NLAITB, KDIF, KDIFBL, SCP, SCN, PHOTREDCOF, GB, GCUT,
     &     GLRADO, CO2AIR, TEMPAIR, VPDAIR,
     &     SSPT, SSPB, TPIPE,
     &     TROOF_NIGHTTB, TROOF_DAYTB, NTROOFNTB, NTROOFDTB,
     &     TROOF_DAY, TROOF_NIGHT, TGROUND,
     &     TEMPAIR_OUT, SCREEN,
     &     TROOF_KNOWN, TEMPAIR_OUT_KNOWN,
     &     MAINLV, MAINST, MAINRT, MAINSO, Q10MN, REFTMP,
     &     ASRQLV, ASRQST, ASRQRT, ASRQSO,
     &     FLVTB, FSTTB, FRTTB, FSOTB,
     &     NFLVTB, NFSTTB, NFRTTB, NFSOTB,
     &     WLVTB, WSTTB, WRTTB, WSOTB,
     &     NWLVTB, NWSTTB, NWRTTB, NWSOTB,
     &     WLVI, WSTI, WRTI, WSOI,
     &     WATCONI, WATCONMAX, RWATCONWI, PSIWIL, PSIROOTM,
     &     RESWAT, RIONUPT,
     &     FILE_END )

*     Initialization of water status
*        End of simulations
         IF( FILE_END ) GOTO 199


*     Ground surface temperature assumed to be air temperature
         TGROUND = TEMPAIR
* ============================================================

         IF( SCREEN .GT. 50. ) THEN
               TROOF = TEMPAIR
           ELSE
             IF( .NOT. TROOF_KNOWN ) THEN
               IF( TEMPAIR_OUT_KNOWN ) THEN
                  TROOF = TEMPAIR_OUT + 0.33 * (TEMPAIR - TEMPAIR_OUT)
               ELSE
                TROOF = DCURTEMP( IWAR_TROOF, TROOF_DAY, TROOF_NIGHT,
     &                     TROOF_DAY, TROOF_NIGHT, DAYL, SOLHR )
               ENDIF
           ENDIF
```

```
          ENDIF
*          of SCREEN


          CALL FAST(
     &   ITASKH, OUTPUTH, DELTF,
     &   DAYNR, HOUR,
     &   SOLARC, ELEVN, SINELV,
     &   TRDIF, TRCOR_UV, TRCON, TRGLAS,
     &   LAI, KDIFBL, KDIF, SCP, SCN, GB, GCUT,
     &   GLRADO, CO2AIR, TEMPAIR, VPDAIR,
     &   SSPT, SSPB, TPIPE, TROOF, TGROUND, REFGR, PHOTRED,
     &   PGROS, TRAN_SIM,
     &   GLOBRAD, NETRAD, RADABS, PAR, RWUPT, WATCON, PSIPL, GSTOT,
     &   DGLRADO, DGLOBRAD, DPAR, DPARABS, DRADABS,
     &   DTGA, DTRANS, DTRANS_D, DNETRAD, DNETRAD_D
     &   )

*         Water status
          CALL WATSTAT( ITASKH,
     &     WATCONI, WATCONMAX, RWATCONWI, PSIWIL, PSIROOTM, RESWAT,
     &     RIONUPT,
     &     DELTF, HOUR, TEMPAIR, TRAN_SIM, RWUPT, WATCON, PSIPL )

*         Maintenance respiration (g CH2O m-2 s-1)
          CALL MNRESP( ITASKH, DELTF, Q10MN, REFTMP,
     &             WLV, WST, WRT, WSO,
     &       MAINLV, MAINST, MAINRT, MAINSO,
     &       TEMPAIR, DMAINT )

          IF( OUTPUTH ) THEN

*         3.6 converts mg s-1 to g h-1
          WRITE( IUOUTH, 901) DAYNR, HOUR,
     &       GLRADO, GLOBRAD, NETRAD, RADABS, PAR,
     &       CO2air, TEMPair, VPDair,
     &       PGROS * 3.6, TRAN_SIM * 3.6, RWUPT, WATCON, GSTOT

901       FORMAT( F5.0, ',', F8.3, ',', 5(F5.0,','),
     &             F6.0,',', F6.1, ',', F6.2, ',',
     &             F7.2, ',',  F6.1, ',', F6.1, ',', F7.0, ',', F8.3 )
          ENDIF
*          of OUTPUTH


*---      Time update
          CALL HTIMER2( INI,
     &        TMIN80, TOTDAY80, DELTF, DELTMIN, OUTDELMIN,
     &        DAYMIN, HOUR, OUTPUTH, DAYTASKS )
*         From standard time to solar time
          SOLHR = HOUR + TIMCOR

*         Exit loop when beginning of new day
          IF( DAYTASKS ) THEN
             GOTO 99
          ENDIF
      END DO
*       end of within day loop

99    CONTINUE

* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
*     Terminal (end of day) calculations
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      ITASKH = 4

      CALL FAST(
     &  ITASKH, OUTPUTH, DELTF,
     &  DAYNR, HOUR,
     &  SOLARC, ELEVN, SINELV,
     &  TRDIF, TRCOR_UV, TRCON, TRGLAS,
     &  LAI, KDIFBL, KDIF, SCP, SCN, GB, GCUT,
     &  GLRADO, CO2AIR, TEMPAIR, VPDAIR,
     &  SSPT, SSPB, TPIPE, TROOF, TGROUND, REFGR, PHOTRED,
     &  PGROS, TRAN_SIM,
     &  GLOBRAD, NETRAD, RADABS, PAR, RWUPT, WATCON, PSIPL, GSTOT,
     &  DGLRADO, DGLOBRAD, DPAR, DPARABS, DRADABS,
```

```
     &  DTGA, DTRANS, DTRANS_D, DNETRAD, DNETRAD_D
     &  )

*****  End of calculations within day **********************************

*      Dry matter productions (g m-2 d-1)
       CALL DMPROD2( ITASKD, DAYNR, DELT,
     &             WLVI, WSTI, WRTI, WSOI,
     &             FLV, FST, FRT, FSO,
     &             ASRQLV, ASRQST, ASRQRT, ASRQSO,
     &             DTGA, DMAINT,
     &             GLV, GST, GRT, GSO, GTW,
     &             WLV, WST, WRT, WSO, TWT,
     &             CWLV, CWST, CWRT, CWSO, CTWT )

      WRITE(*, '(A, I5 )') '+ ', IDAY

      IF( OUTPUT ) THEN
        WRITE( IUOUTC, 920)
     &   DAYNR, DGLRADO, DGLOBRAD, DPAR, DPARABS,
     &   DTGA, DMAINT, GTW, CWLV, CWST, CWRT, CWSO, CTWT
920     FORMAT( F6.0, ',', 4(F6.2,','), 3(F6.2,','), 5(F7.1,',') )

        WRITE( IUOUTW, 903)
     &   DAYNR, DGLRADO, DGLOBRAD, DPAR, DPARABS,
     &   DRADABS, DNETRAD, DNETRAD_D, DTRANS, DTRANS_D
903     FORMAT( F6.0, ',', 4(F6.2,','), 3(F6.2,','), 2(F7.1,',') )
      ENDIF

      CALL DTIMER( INI,
     & IYEAR, STYEAR, FINYEAR, STARTDAY, FINDAY, IDAY, DAYNR, TOTDAY80,
     & OUTDELDAY, OUTPUT, TERMNL
     & )


      END DO
*      end of DO WHILE .NOT. TERMNL

199    CONTINUE

* ==============================================================
*     Terminal section
* ==============================================================

      WRITE( *, '(A,A)' ) ' Output to : ', FILOUTH
      WRITE( *, '(A,A)' ) ' Output to : ', FILOUTC
      WRITE( *, '(A,A)' ) ' Output to : ', FILOUTW

      END
***********************************************************************
* SUBPROGRAM: FAST
* Comment: subroutine FAST collects several routines for execution
*          at short time steps
*
* Subprograms called:
*        CANOPF, FDIF_10M, NIRFLUX, PARFLUX,
***********************************************************************
      SUBROUTINE FAST(
     & ITASK, OUTPUT, DELT,
     & DAYNR, HOUR,
     & SOLARC, ELEVN, SINELV,
     & TRDIF, TRCOR_UV, TRCON, TRGLAS,
     & LAI, KDIFBL, KDIF, SCP, SCN, GB, GCUT,
     & GLRADO, CO2AIR, TEMPAIR, VPDAIR,
     & SSPT, SSPB, TPIPE, TROOF, TGROUND, REFGR, PHOTRED,
     & PGROS, TRAN_SIM,
     & GLOBRAD, NETRAD, RADABS, PAR, RWUPT, WATCON, PSIPL, GSTOT,
     & DGLRADO, DGLOBRAD, DPAR, DPARABS, DRADABS,
     & DTGA, DTRANS, DTRANS_D, DNETRAD, DNETRAD_D
     & )
      IMPLICIT REAL(A-Z)

      INTEGER ITASK

      LOGICAL INI
      LOGICAL LIGHT
      COMMON /GENCOM/ LIGHT
```

```
      LOGICAL OUTPUT
      LOGICAL DAYTASKS

      CHARACTER*5 EXPRNT

      INTEGER IUOUTH
      COMMON /IO_UNIT_OUT1/ IUOUTH

      COMMON /CLIMHUM/ VPair

      COMMON /ENERGY_EXCH1/ TCANOP
      COMMON /ENERGY_EXCH2/ HF_CR, HF_SC, HF_PC
      COMMON /ENERGY_EXCH3/ HFCRTOT, HFSCTOT, HFPCTOT, CONVH

* Parameters
      COMMON /LEAFPAR/ Rcut, Rb
      COMMON /LEAFPAR_mol/ Rcut_mol, Rb_mol
      COMMON /LEAFPAR2/ Gmaxd, GNVPD
      COMMON /LEAFPAR3/ Gmaxda, GNVPDa

* ==============================================================
*         Initialization
* ==============================================================
      IF( ITASK .EQ. 1 ) THEN

      RB = 1. / GB
      RCUT = 1. / GCUT

*     Resistance for thermal radiation
      RTHRAD = 200. / KDIFBL

      Rb_mol = Rb / 40.
      Rcut_mol = Rcut / 40.

* Initial value for stomatal conductance
      gsin = 2. * GMAXDA

      SQP = SQRT( 1. - SCP )

      INI = .TRUE.
      CALL CANOPF( INI, HOUR, PARDIF, PARDIR, NIRDIF,
     &       NIRDIR, UVDIF, UVDIR, ELEVN,
     &       LAI, KDIFBL, KDIF, SCP, SCN, CO2air, TEMPAIR, VPDair,
     &       SSPT, SSPB, TPIPE, TROOF, TGROUND,
     &       PHOTRED, GSin, RTHRAD, REFGR,
     &       PARABS, PARDIRTO, NIRABS, UVABS, NETRAD,
     &       GSTOT, GLTOT,
     &       PGROS, TRAN_SIM
     &       )
      INI = .FALSE.

      ELSEIF( ITASK .EQ. 5 ) THEN
* ==============================================================
*         Resetting
* ==============================================================

      DTRANS_D = 0.
      DTRANS_N = 0.
      DGLRADO = 0.
      DGLOBRAD = 0.
      DPAR = 0.
      DPARABS = 0.
      DRADABS = 0.
      DTGA = 0.
      DNETRAD_D = 0.
      DNETRAD_N = 0.

      ELSEIF( ITASK .EQ. 3 ) THEN
* ==============================================================
*         Integration
* ==============================================================

*     Calculation of daily total
         IF( LIGHT ) THEN
*           3.6 converts mg s-1 to g h-1
            DTRANS_D = DTRANS_D + TRAN_SIM * 3.6 * DELT
            DNETRAD_D = DNETRAD_D + NETRAD * DELT
```

```
            DGLRADO = DGLRADO + GLRADO * DELT
            DGLOBRAD = DGLOBRAD + GLOBRAD * DELT
            DPAR = DPAR + PAR * DELT
            DRADABS = DRADABS + RADABS * DELT
            DPARABS = DPARABS + PARABS * DELT
            DTGA = DTGA + PGROS * 3.6 * DELT
         ELSE
*           3.6 converts mg s-1 to g h-1
            DTRANS_N = DTRANS_N + TRAN_SIM * 3.6 * DELT
            DNETRAD_N = DNETRAD_N + NETRAD * DELT
         ENDIF

* ================================================================
*          Calculations driving variables
* ================================================================
      ELSEIF( ITASK .EQ. 2 ) THEN


*        Vapour pressure and saturated vapour pressure of
*        greenhouse air [kPa]
         VPSATair = .6107 * EXP( 17.4 * TEMPair / (TEMPair + 239.) )
         VPair = VPSATair - VPDair


         IF (GLRADO .LT. 0.1 ) THEN
            LIGHT = .FALSE.
            PARDIF = 0.
            PARDIR = 0.
            NIRDIF = 0.
            NIRDIR = 0.
            GLRADO = 0.
            UVDIR = 0.
            UVDIF = 0.
         ELSE
            LIGHT = .TRUE.

            SINELV = AMAX1( .05, SINELV )
            ELEVN = AMAX1( .05, ELEVN )

*           Atmospheric transmission
            ATMTR = GLRADO / (SOLARC * SINELV )

*           Direct and diffuse radiation outside greenhouse
            FRDIF = FDIF_10M( SOLARC, GLRADO, SINELV )
            GLOBDIFO = FRDIF * GLRADO
            GLOBDIRO = GLRADO - GLOBDIFO

*           Direct and diffuse PAR outside greenhouse
            CALL PARFLUX( ATMTR, GLRADO, ELEVN,
     &                       FRDIF, PAROUT, FRDIFPAR )

            PARDIFO = FRDIFPAR * PAROUT
            PARDIRO = PAROUT - PARDIFO

*           Direct and diffuse NIR and UV outside greenhouse
            CALL NIRFLUX( ATMTR, GLOBDIFO, GLOBDIRO,
     &                    PARDIFO, PARDIRO, NIRDIFO, NIRDIRO,
     &                    UVDIFO, UVDIRO )


*          Diffuse and direct PAR and NIR inside greenhouse
           PARDIF = PARDIFO * TRDIF
           UVDIF = UVDIFO * TRDIF * TRCOR_UV

           TRDIR = TRCON * TRGLAS
           PARDIR = PARDIRO * TRDIR
           UVDIR = UVDIRO * TRDIR * TRCOR_UV

           NIRDIR = NIRDIRO * TRDIR
           NIRDIF = NIRDIFO * TRDIF

         ENDIF
*        of GLRADO .LT. 0

         NIR = NIRDIR + NIRDIF

         PAR = PARDIF + PARDIR
```

```
         GLOBDIR = PARDIR + NIRDIR + UVDIR
         UV = UVDIF + UVDIR

         GLOBDIF = PARDIF + NIRDIF + UVDIF
         GLOBRAD = PAR + NIR + UV

* ===============================================================
*          Rate calculations
* ===============================================================
      WRITE( 99, *)
      WRITE( 99, '(4F9.4)') DAYNR, HOUR, PAR
          CALL CANOPF( INI, HOUR, PARDIF, PARDIR, NIRDIF,
     &        NIRDIR, UVDIF, UVDIR, ELEVN,
     &        LAI, KDIFBL, KDIF, SCP, SCN, CO2air, TEMPAIR, VPDair,
     &        SSPT, SSPB, TPIPE, TROOF, TGROUND,
     &        PHOTRED, GSin, RTHRAD, REFGR,
     &        PARABS, PARDIRTO, NIRABS, UVABS, NETRAD,
     &        GSTOT, GLTOT,
     &        PGROS, TRAN_SIM
     &        )

         RADABS = PARABS + NIRABS + UVABS

*        Energy flux associated with transpiration
         TRAN_ENER_SIM = TRAN_SIM * 2.5

99    CONTINUE

* ===============================================================
*          Terminal
* ===============================================================
      ELSEIF( ITASK .EQ. 4 ) THEN

      DTRANS = DTRANS_D + DTRANS_N
      DNETRAD = DNETRAD_D + DNETRAD_N

*     Conversion of radiation fluxes to MJ m-2
      CF = 3600. * 1.E-6
      DNETRAD = DNETRAD * CF
      DNETRAD_D = DNETRAD_D * CF
      DGLRADO = DGLRADO * CF
      DGLOBRAD  = DGLOBRAD * CF
      DPAR = DPAR * CF
      DPARABS = DPARABS * CF
      DRADABS = DRADABS * CF

      ENDIF
*       of ITASK

      RETURN
      END
***********************************************************************
* Phases in I/O
*   IOPHASE = 1
*   -  get date from system
*   -  get general info from command line or via Q&A
*       (inputfile, datafile, data-infofile, param file, timer file
*        exp. name, run options)
*   -  obtain info about layout datafile from data-infofile
*   -  get data from parameter file
*   -  get data from timer file
*   -  opening datafile
*   IOPHASE = 3
*   -  Q&A about change of parameters, timer values, run options
*   -  write info in headers of output files
*   IOPHASE = 4
*   -  reading variables from datafile
*
* Subprograms called:
*   OUTDAT : get time from system, and output to file
*   COMMAN : get command line arguments
*   INPUTI : get value of integer variable from user
*   INPUTR : get value of real variable from user
*   INPUTT : get string for character variable from user
*   READVAR : read value of real or integer variable from input file
*   READTB2 : read values for real array from input file
*   GFIDATA : get layout of time-series file
```

```
*     READDT2 : read time and real values from time-series file
************************************************************************
      SUBROUTINE ENVINT( IOPHASE, EXPRNT,
     &  DAYNR, SOLHR, HOUR, TMIN80, DATA_DAYMIN, DAYMIN, SINELV,
     &  IYEAR, STYEAR, FINYEAR,
     &  STARTDAY, FINDAY, OUTDELDAY, DELTMIN, OUTDELMIN,
     &  LAT, TIMCOR, SUNRISE, SUNSET, DAYL, REFGR,
     &  AZIMGR, TRDIF, TRCOR_UV,
     &  LAITB, NLAITB, KDIF, KDIFBL, SCP, SCN, PHOTREDCOF, GB, GCUT,
     &  GLRADO, CO2AIR, TEMPAIR, VPDAIR,
     &  SSPT, SSPB, TPIPE,
     &  TROOF_NIGHTTB, TROOF_DAYTB, NTROOFNTB, NTROOFDTB,
     &  TROOF_DAY, TROOF_NIGHT, TGROUND,
     &  TEMPAIR_OUT, SCREEN,
     &  TROOF_KNOWN, TEMPAIR_OUT_KNOWN,
     &  MAINLV, MAINST, MAINRT, MAINSO, Q10MN, REFTMP,
     &  ASRQLV, ASRQST, ASRQRT, ASRQSO,
     &  FLVTB, FSTTB, FRTTB, FSOTB,
     &  NFLVTB, NFSTTB, NFRTTB, NFSOTB,
     &  WLVTB, WSTTB, WRTTB, WSOTB,
     &  NWLVTB, NWSTTB, NWRTTB, NWSOTB,
     &  WLVI, WSTI, WRTI, WSOI,
     &  WATCONI, WATCONMAX, RWATCONWI, PSIWIL, PSIROOTM,
     &  RESWAT, RIONUPT,
     &  FILE_END )
      IMPLICIT REAL(A-Z)

      CHARACTER*5 EXPRNT
      LOGICAL EXPRNT_KNOWN

*     Simulation control
      LOGICAL INI
      LOGICAL FILE_END
      LOGICAL LIGHT
      COMMON /GENCOM/ LIGHT
      INTEGER IOPHASE

*     File I/O
      INTEGER IUOUTH
      COMMON /IO_UNIT_OUT1/ IUOUTH
      INTEGER IUOUTC
      COMMON /IO_UNIT_OUT2/ IUOUTC
      INTEGER IUOUTW
      COMMON /IO_UNIT_OUT3/ IUOUTW
      CHARACTER*40 FILOUTH, FILOUTC, FILOUTW
      COMMON /IO_NAME_OUT/ FILOUTH, FILOUTC, FILOUTW

      INTEGER IUDAT, IUDATIF, IUTIM, IUPAR, IUTRAN
      CHARACTER*40 FILTRN
      CHARACTER*40 DATAFIL, INFOFIL, PARFIL
      CHARACTER*40 TIMFIL
      COMMON /IO_UNIT_IN/ IUDAT, IUDATIF, IUTIM, IUPAR, IUTRAN
      COMMON /IO_NAME_IN/ DATAFIL, INFOFIL, TIMFIL, PARFIL, FILTRN

* ## ----------------------------------------------------------------
* Parameters
      COMMON /FARQ_PAR5/ VCMAX250, JMAX250, RD250, THETA,
     &                   LGHTCON, KC25, KO25

      COMMON /LEAFPAR/ Rcut, Rb
      COMMON /LEAFPAR_mol/ Rcut_mol, Rb_mol
      COMMON /LEAFPAR2/ Gmaxd, GNVPD
      COMMON /LEAFPAR3/ Gmaxda, GNVPDa

      COMMON /PARSTOM/ Gsmax, CD1, CD2, CD3, CD4, CD5, CD6

      LOGICAL DO_AIR, DO_LFAIR
      COMMON /PARSTOM2/ DO_AIR, DO_LFAIR

* ## ----------------------------------------------------------------
      LOGICAL VPDRESP_KNOWN
      INTEGER IVPDRESP

      CHARACTER*1 STR1
      CHARACTER*40 STR40
      INTEGER ITASK
      INTEGER IP, IPP
```

```
      INTEGER*2 ie

* Timer variables
      INTEGER DAYMIN, TMIN80, DATA_DAYMIN, DATA_TMIN80, DATA_DAYNR
      INTEGER IDAYNR
      INTEGER DELTMIN, OUTDELMIN
      INTEGER IYEAR, STYEAR, FINYEAR, STARTDAY, FINDAY
      INTEGER OUTDELDAY

      INTEGER NR_OF_VAR
      CHARACTER*60 FMT
      INTEGER IUNERR

      INTEGER IDUM
      CHARACTER*40 LABEL

      DIMENSION VAR( 20 )

      LOGICAL VAR_FOUND
      INTEGER HOUR_COL
      LOGICAL HOUR_FOUND

      LOGICAL YEAR_FOUND, YEAR_KNOWN
      INTEGER YEAR_COL

      LOGICAL DAYNR_FOUND, DAYNR_KNOWN
      INTEGER DAYNR_COL

      LOGICAL GLRADO_FOUND, GLRADO_KNOWN
      INTEGER GLRADO_COL

      LOGICAL GLOBDIFO_FOUND, GLOBDIFO_KNOWN
      INTEGER GLOBDIFO_COL

      LOGICAL CO2AIR_FOUND, CO2AIR_KNOWN
      INTEGER CO2AIR_COL

      LOGICAL TEMPAIR_FOUND, TEMPAIR_KNOWN
      INTEGER TEMPAIR_COL

      LOGICAL TEMPAIR_OUT_FOUND, TEMPAIR_OUT_KNOWN
      INTEGER TEMPAIR_OUT_COL

      LOGICAL VPDAIR_FOUND, VPDAIR_KNOWN
      INTEGER VPDAIR_COL

      LOGICAL PHOT_MEAS_FOUND, PHOT_MEAS_KNOWN
      INTEGER PHOT_MEAS_COL

      LOGICAL TRAN_MEAS_FOUND, TRAN_MEAS_KNOWN
      INTEGER TRAN_MEAS_COL

      LOGICAL TPIPE_FOUND, TPIPE_KNOWN
      INTEGER TPIPE_COL

      LOGICAL TROOF_FOUND, TROOF_KNOWN
      INTEGER TROOF_COL
      LOGICAL TROOFeqTAIR
      INTEGER IWAR_TROOF

      LOGICAL TGROUND_FOUND, TGROUND_KNOWN
      INTEGER TGROUND_COL
      LOGICAL TGROUNDeqTAIR

      LOGICAL SCREEN_FOUND, SCREEN_KNOWN
      INTEGER SCREEN_COL

      LOGICAL CHECK_EVEN

      DIMENSION LAITB(80)
      INTEGER NLAITB

      DIMENSION FLVTB(80), FSTTB(80), FRTTB(80), FSOTB(80)
      INTEGER NFLVTB, NFSTTB, NFRTTB, NFSOTB

      DIMENSION WLVTB(80), WSTTB(80), WRTTB(80), WSOTB(80)
      INTEGER NWLVTB, NWSTTB, NWRTTB, NWSOTB
```

```
      DIMENSION TROOF_NIGHTTB(80)
      INTEGER NTROOFNTB
      DIMENSION TROOF_DAYTB(80)
      INTEGER NTROOFDTB

*     I/O
      INTEGER*4 numarg
      INTEGER*2 iarg
      CHARACTER*40 comlintb( 10 )
      INTEGER strlentb( 10 )
      CHARACTER*160 str160
      CHARACTER*40 parsstrtb( 10 )

      LOGICAL INFOFIL_KNOWN, PARFIL_KNOWN, DATAFIL_KNOWN, TIMFIL_KNOWN

      LOGICAL command
      COMMON /IO_0/ command

      LOGICAL RUNSTRING_KNOWN
      CHARACTER*1 RUNSTRING

      SAVE

      PI = 3.1415926
      RADN = PI / 180.

      IF( IOPHASE .EQ. 1 ) THEN

* Get date from system
*VAX*
*      CALL OUTDATV( IOPHASE, IUOUT )
*PC*
      CALL OUTDAT( IOPHASE, IUOUTH )

      INI = .TRUE.

      EXPRNT_KNOWN = .FALSE.
      PARFIL_KNOWN = .FALSE.
      TIMFIL_KNOWN = .FALSE.
      DATAFIL_KNOWN = .FALSE.
      INFOFIL_KNOWN = .FALSE.

      VPDRESP_KNOWN = .FALSE.

      RUNSTRING_KNOWN = .FALSE.

* ================================================================
* First check whether command line input is done, in that case no
* user interaction is done
*VAX*
*         CALL COMMANV( comlintb, strlentb, numarg, command )
*PC*
         CALL COMMAN( comlintb, strlentb, numarg, command )

      IF( command ) THEN
         DO iarg = 1, numarg
            ie = strlentb( iarg )
            IF( comlintb(iarg)(1:4) .EQ. 'EXP:' ) THEN
                 READ( comlintb(iarg)(5:ie), '(A)') EXPRNT
                 EXPRNT_KNOWN = .TRUE.
            ELSEIF( comlintb(iarg)(1:4) .EQ. 'DAT:' ) THEN
                 READ( comlintb(iarg)(5:ie), '(A)') DATAFIL
                 DATAFIL_KNOWN = .TRUE.
            ELSEIF( comlintb(iarg)(1:3) .EQ. 'FI:' ) THEN
                 READ( comlintb(iarg)(4:ie), '(A)') INFOFIL
                 INFOFIL_KNOWN = .TRUE.
            ELSEIF( comlintb(iarg)(1:4) .EQ. 'PAR:' ) THEN
                 READ( comlintb(iarg)(5:ie), '(A)') PARFIL
                 PARFIL_KNOWN = .TRUE.
            ELSEIF( comlintb(iarg)(1:4) .EQ. 'TIM:' ) THEN
                 READ( comlintb(iarg)(5:ie), '(A)') TIMFIL
                 TIMFIL_KNOWN = .TRUE.
            ELSEIF( comlintb(iarg)(1:4) .EQ. 'RUN:' ) THEN
                 READ( comlintb(iarg)(5:5), '(A1)') RUNSTRING
                 RUNSTRING_KNOWN = .TRUE.
            ENDIF
```

```
         END DO
      ENDIF

* ================================================================
      STR40 = ' '
      IF( .NOT. EXPRNT_KNOWN ) THEN
         CALL INPUTT( ' Which experiment?  : ', STR40 )
         EXPRNT = STR40(1:5)
      ENDIF

      IF( .NOT. datafil_known ) THEN
         CALL INPUTT( ' Which data file?  : ', DATAFIL )
      ENDIF

* ================================================================
* Read information about datafile from information file

* File with information on layout of data files
      IF( .NOT. infofil_known ) THEN
          INFOFIL = 'DM.FI'
      ENDIF

* Get layout of data file
      CALL GFIDATA( IUDATIF, INFOFIL,
     &  NR_OF_VAR, FMT,
     &  YEAR_FOUND, YEAR_COL,
     &  DAYNR_FOUND, DAYNR_COL, HOUR_FOUND, HOUR_COL, GLRADO_FOUND,
     &  GLRADO_COL, GLOBDIFO_FOUND, GLOBDDIFO_COL,
     &  CO2AIR_FOUND, CO2AIR_COL, TEMPAIR_FOUND,
     &  TEMPAIR_COL, VPDAIR_FOUND, VPDAIR_COL, TPIPE_FOUND, TPIPE_COL,
     &  TROOF_FOUND, TROOF_COL, TGROUND_FOUND, TGROUND_COL,
     &  SCREEN_FOUND, SCREEN_COL, TEMPAIR_OUT_FOUND, TEMPAIR_OUT_COL,
     &  PHOT_MEAS_FOUND, PHOT_MEAS_COL,
     &  TRAN_MEAS_FOUND, TRAN_MEAS_COL )

       IF( .NOT. GLRADO_FOUND ) THEN
           GLRADO_KNOWN = .FALSE.
       ELSE
           GLRADO_KNOWN = .TRUE.
       ENDIF
       IF( .NOT. GLOBDIFO_FOUND ) THEN
           GLOBDIFO_KNOWN = .FALSE.
       ELSE
           GLOBDIFO_KNOWN = .TRUE.
       ENDIF
       IF( .NOT. CO2AIR_FOUND ) THEN
           CO2AIR_KNOWN = .FALSE.
       ELSE
           CO2AIR_KNOWN = .TRUE.
       ENDIF
       IF( .NOT. TEMPair_FOUND ) THEN
           TEMPair_KNOWN = .FALSE.
       ELSE
           TEMPair_KNOWN = .TRUE.
       ENDIF
       IF( .NOT. VPDAIR_FOUND ) THEN
           VPDAIR_KNOWN = .FALSE.
       ELSE
           VPDAIR_KNOWN = .TRUE.
       ENDIF
       IF( .NOT. PHOT_MEAS_FOUND ) THEN
           PHOT_MEAS_KNOWN = .FALSE.
       ELSE
           PHOT_MEAS_KNOWN = .TRUE.
       ENDIF
       IF( .NOT. TRAN_MEAS_FOUND ) THEN
           TRAN_MEAS_KNOWN = .FALSE.
       ELSE
           TRAN_MEAS_KNOWN = .TRUE.
       ENDIF
       IF( .NOT. TPIPE_FOUND ) THEN
           TPIPE_KNOWN = .FALSE.
       ELSE
           TPIPE_KNOWN = .TRUE.
       ENDIF
       IF( .NOT. TROOF_FOUND ) THEN
```

```
            TROOF_KNOWN = .FALSE.
        ELSE
            TROOF_KNOWN = .TRUE.
        ENDIF
        IF( .NOT. TGROUND_FOUND ) THEN
            TGROUND_KNOWN = .FALSE.
        ELSE
            TGROUND_KNOWN = .TRUE.
        ENDIF
        IF( .NOT. TEMPAIR_OUT_FOUND ) THEN
            TEMPAIR_OUT_KNOWN = .FALSE.
        ELSE
            TEMPAIR_OUT_KNOWN = .TRUE.
        ENDIF

* ================================================================
*  Read timer values
      IF( .NOT. timfil_known ) THEN
         TIMFIL = 'TIMER.DAT'
         CALL INPUTT ( ' Which timer file? : ', TIMFIL )
      ENDIF

      LABEL = ' '
      OPEN( UNIT = IUTIM, FILE = TIMFIL, STATUS  = 'OLD')
      CALL READVAR( IUTIM, LABEL, 'IYEAR', XDUM, IYEAR )
      CALL READVAR( IUTIM, LABEL, 'STYEAR', XDUM, STYEAR )
      CALL READVAR( IUTIM, LABEL, 'FINYEAR', XDUM, FINYEAR )
      CALL READVAR( IUTIM, LABEL, 'STARTDAY', XDUM, STARTDAY )
      CALL READVAR( IUTIM, LABEL, 'FINDAY', XDUM, FINDAY )
      CALL READVAR( IUTIM, LABEL, 'OUTDELDAY', XDUM, OUTDELDAY )
      CALL READVAR( IUTIM, LABEL, 'DELTMIN', XDUM, DELTMIN )
      CALL READVAR( IUTIM, LABEL, 'OUTDELMIN', XDUM, OUTDELMIN )
      CLOSE( IUTIM )

* ================================================================
*  Read parameter values
      IF( .NOT. parfil_known ) THEN
         PARFIL = 'INT.PAR'
         CALL INPUTT ( ' Which parameterfile? : ', PARFIL )
      ENDIF

      LABEL = ' '

      OPEN(UNIT = IUPAR, FILE = PARFIL, STATUS  = 'OLD')
        CHECK_EVEN = .TRUE.

        CALL READTB2( IUPAR, LABEL, 'LAITB', CHECK_EVEN, 80,
     &              LAITB, NLAITB )
        CALL READVAR(IUPAR, LABEL, 'KDIF', KDIF, IDUM )
        CALL READVAR(IUPAR, LABEL, 'KDIFBL', KDIFBL, IDUM )
        CALL READVAR(IUPAR, LABEL, 'SCP', SCP, IDUM)
        CALL READVAR(IUPAR, LABEL, 'SCN', SCN, IDUM)

*       Photosynthetic capacities at top of canopy
        CALL READVAR( IUPAR, LABEL, 'VCMAX250', VCMAX250, IDUM )
        CALL READVAR( IUPAR, LABEL, 'JMAX250', JMAX250, IDUM )
        CALL READVAR( IUPAR, LABEL, 'RD250', RD250, IDUM )
        CALL READVAR( IUPAR, LABEL, 'PHOTREDCOF', PHOTREDCOF, IDUM )
        CALL READVAR( IUPAR, LABEL, 'KC25', KC25, IDUM )
        CALL READVAR( IUPAR, LABEL, 'KO25', KO25, IDUM )
        CALL READVAR( IUPAR, LABEL, 'THETA', THETA, IDUM )
        CALL READVAR( IUPAR, LABEL, 'LGHTCON', LGHTCON, IDUM )

        CALL READVAR(IUPAR, LABEL, 'GB', GB,IDUM)
        CALL READVAR(IUPAR, LABEL, 'GCUT', GCUT,IDUM)

        CALL READVAR (IUPAR, LABEL, 'GSMAX', GSMAX, IDUM)
        CALL READVAR (IUPAR, LABEL, 'CD1', CD1, IDUM)
        CALL READVAR (IUPAR, LABEL, 'CD2', CD2, IDUM)
        CALL READVAR (IUPAR, LABEL, 'CD3', CD3, IDUM)
        CALL READVAR (IUPAR, LABEL, 'CD4', CD4, IDUM)
        CALL READVAR (IUPAR, LABEL, 'CD5', CD5, IDUM)
        CALL READVAR (IUPAR, LABEL, 'CD6', CD6, IDUM)

        IF( .NOT. VPDRESP_KNOWN ) THEN
           CALL READVAR( IUPAR, LABEL, 'VPDRESP', XDUM, IVPDRESP )
           VPDRESP_KNOWN = .TRUE.
```

```
      ENDIF

*     Response stomatal conductance to air VPD
      IF( IVPDRESP .EQ. 1 ) THEN
         DO_AIR = .TRUE.
         DO_LFAIR = .FALSE.
*     Response stomatal conductance to leaf-air VPD
      ELSEIF( IVPDRESP .EQ. 2 ) THEN
         DO_AIR = .FALSE.
         DO_LFAIR = .TRUE.
*     Response stomatal conductance to leaf surface VPD
      ELSEIF( IVPDRESP .EQ. 3 ) THEN
         DO_AIR = .FALSE.
         DO_LFAIR = .FALSE.
      ENDIF

      IF( DO_AIR ) THEN
         CALL READVAR( IUPAR, LABEL, 'Gmaxda', Gmaxda, IDUM )
         CALL READVAR( IUPAR, LABEL, 'GNVPDa', GNVPDa, IDUM )
      ELSE
         CALL READVAR( IUPAR, LABEL, 'Gmaxd', Gmaxd, IDUM )
         CALL READVAR( IUPAR, LABEL, 'GNVPD', GNVPD, IDUM )
      ENDIF

      CALL READVAR( IUPAR, LABEL, 'LATITUDE', LAT, IDUM )
      CALL READVAR( IUPAR, LABEL, 'TIMCOR', TIMCOR, IDUM )
      CALL READNAME( IUPAR, LABEL, 'TRANSMISSIVITY_FILE',
     &                     FILTRN)
      CALL READVAR( IUPAR, LABEL, 'AZIMGR', AZIMGR, IDUM)
      CALL READVAR( IUPAR, LABEL, 'REFGR', REFGR, IDUM)

      CALL READVAR( IUPAR, LABEL, 'CO2AIR', CO2air, IDUM )
      CALL READVAR( IUPAR, LABEL, 'TEMPAIR', TEMPair, IDUM )
      CALL READVAR( IUPAR, LABEL, 'VPDAIR', VPDair, IDUM )
      CALL READVAR( IUPAR, LABEL, 'TPIPE', TPIPE, IDUM )
      CALL READVAR( IUPAR, LABEL, 'SSPT', SSPT, IDUM )
      CALL READVAR( IUPAR, LABEL, 'SSPB', SSPB, IDUM )
      CALL READTB2( IUPAR, LABEL, 'TROOF_NIGHTTB', CHECK_EVEN, 80,
     &              TROOF_NIGHTTB, NTROOFNTB )
      CALL READTB2( IUPAR, LABEL, 'TROOF_DAYTB', CHECK_EVEN, 80,
     &              TROOF_DAYTB, NTROOFDTB )

*     Parameters dry matter production
      CALL READVAR( IUPAR, LABEL, 'MAINLV', MAINLV, IDUM )
      CALL READVAR( IUPAR, LABEL, 'MAINST', MAINST, IDUM )
      CALL READVAR( IUPAR, LABEL, 'MAINRT', MAINRT, IDUM )
      CALL READVAR( IUPAR, LABEL, 'MAINSO', MAINSO, IDUM )
      CALL READVAR( IUPAR, LABEL, 'ASRQLV', ASRQLV, IDUM )
      CALL READVAR( IUPAR, LABEL, 'ASRQST', ASRQST, IDUM )
      CALL READVAR( IUPAR, LABEL, 'ASRQRT', ASRQRT, IDUM )
      CALL READVAR( IUPAR, LABEL, 'ASRQSO', ASRQSO, IDUM )
      CALL READVAR( IUPAR, LABEL, 'Q10MN', Q10MN, IDUM )
      CALL READVAR( IUPAR, LABEL, 'REFTMP', REFTMP, IDUM )
      CALL READTB2( IUPAR, LABEL, 'FLVTB', CHECK_EVEN, 80,
     &              FLVTB, NFLVTB )
      CALL READTB2( IUPAR, LABEL, 'FSTTB', CHECK_EVEN, 80,
     &              FSTTB, NFSTTB )
      CALL READTB2( IUPAR, LABEL, 'FRTTB', CHECK_EVEN, 80,
     &              FRTTB, NFRTTB )
      CALL READTB2( IUPAR, LABEL, 'FSOTB', CHECK_EVEN, 80,
     &              FSOTB, NFSOTB )
      CALL READVAR( IUPAR, LABEL, 'Q10MN', Q10MN, IDUM )
      CALL READVAR( IUPAR, LABEL, 'WLVI', WLVI, IDUM )
      CALL READVAR( IUPAR, LABEL, 'WSTI', WSTI, IDUM )
      CALL READVAR( IUPAR, LABEL, 'WRTI', WRTI, IDUM )
      CALL READVAR( IUPAR, LABEL, 'WSOI', WSOI, IDUM )
      CALL READTB2( IUPAR, LABEL, 'WLVTB', CHECK_EVEN, 80,
     &              WLVTB, NWLVTB )
      CALL READTB2( IUPAR, LABEL, 'WSTTB', CHECK_EVEN, 80,
     &              WSTTB, NWSTTB )
      CALL READTB2( IUPAR, LABEL, 'WRTTB', CHECK_EVEN, 80,
     &              WRTTB, NWRTTB )
      CALL READTB2( IUPAR, LABEL, 'WSOTB', CHECK_EVEN, 80,
     &              WSOTB, NWSOTB )

*     Reading parameters water status
      CALL READVAR( IUPAR, LABEL, 'WATCONI', WATCONI, IDUM )
```

```
        CALL READVAR( IUPAR, LABEL, 'WATCONMAX', WATCONMAX, IDUM )
        CALL READVAR( IUPAR, LABEL, 'RWATCONWI', RWATCONWI, IDUM )
        CALL READVAR( IUPAR, LABEL, 'PSIWIL', PSIWIL, IDUM )
        CALL READVAR( IUPAR, LABEL, 'PSIROOTM', PSIROOTM, IDUM )
        CALL READVAR( IUPAR, LABEL, 'RESWAT', RESWAT, IDUM )
        CALL READVAR( IUPAR, LABEL, 'RIONUPT', RIONUPT, IDUM )

      CLOSE(IUPAR)

      OPEN( UNIT = IUDAT, FILE = DATAFIL, STATUS = 'OLD' )

*     Proceed to first record with data
      CALL READDT2( INI, IUDAT, FMT, YEAR_COL, DAYNR_COL, HOUR_COL,
     &              NR_OF_VAR, TMIN80,
     &              DATA_TMIN80, DATA_DAYNR, DATA_DAYMIN, VAR,
     &              IUNERR )

      INI = .FALSE.
* ==============================================================
      ELSEIF( IOPHASE .EQ. 3 ) THEN

      IF( .NOT. command ) THEN
         CALL INPUTR( ' Boundary layer conductance      : ', GB )
         CALL INPUTR(' GSMAX        : ', GSMAX )
         CALL INPUTR(' CD1          : ', CD1 )
         CALL INPUTR(' CD2          : ', CD2 )
         CALL INPUTR(' CD3          : ', CD3 )
         CALL INPUTR(' CD4          : ', CD4 )
         CALL INPUTR(' CD5          : ', CD5 )
         CALL INPUTR(' CD6          : ', CD6 )
      ENDIF


      FILOUTH = ' '
      IP = INDEX( DATAFIL, '.' )
      IPP = IP + 1
      WRITE( FILOUTH(1:IP-1), '(A)') DATAFIL(1:IP-1)
      WRITE( FILOUTH(IP:IP+4), '(A)') ' .CSV'
      IF( .NOT. RUNSTRING_KNOWN ) THEN
         WRITE (*, '(A,A)') ' Output file : ', FILOUTH
         WRITE(*, '(A,$)')
     &   ' Give alphanumeric character to fill in space of name : '
         READ (*,'(A1)')  RUNSTRING
      ENDIF
      WRITE (FILOUTH(IP:IP), '(A1)') RUNSTRING
*VAX*
*      OPEN( UNIT=IUOUTH,FILE=FILOUT, STATUS='NEW',
*     &     RECL = 250, CARRIAGECONTROL = 'LIST')
*PC*
      OPEN( UNIT=IUOUTH, FILE=FILOUTH, STATUS='UNKNOWN')

      WRITE(IUOUTH, '(A,A)') ' FILE: ', FILOUTH
*VAX*
*      CALL OUTDATV( IOPHASE, IUOUT )
*PC*
      CALL OUTDAT( IOPHASE, IUOUTH )
      WRITE(IUOUTH, '(A, A)') ' EXPRNT:', EXPRNT
      WRITE(IUOUTH, '(A,F6.0)') ' DAYNR:', DAYNR
      WRITE(IUOUTH, '(A, A1)') ' RUN:', RUNSTRING
      WRITE(IUOUTH, '(A,F6.3)') ' GB:', GB
      WRITE(IUOUTH, '(A,F8.5)') ' GCUT:', GCUT
      WRITE(IUOUTH, '(A,F5.0)') ' VCMAX250:', VCMAX250
      WRITE(IUOUTH, '(A,F5.0)') ' JMAX250:', JMAX250
      WRITE(IUOUTH, '(A,F6.2)') ' PHOTRDCOF:', PHOTRDCOF
      WRITE(IUOUTH, '(A,F6.2)') ' LGHTCON:', LGHTCON
      IF( .NOT. DO_AIR ) THEN
         WRITE(IUOUTH, '(A,F8.5)') ' Gmaxd:', Gmaxd
         WRITE(IUOUTH, '(A,F7.4)') ' GNVPD:', GNVPD
      ELSE
         WRITE(IUOUTH, '(A,F8.5)') ' Gmaxda:', Gmaxda
         WRITE(IUOUTH, '(A,F7.4)') ' GNVPDa:', GNVPDa
      ENDIF
      IF( IVPDRESP .EQ. 1 ) THEN
         WRITE( IUOUTH, '(A,A)' ) ' Stom._resp_to: ', 'air_VPD'
      ELSEIF( IVPDRESP .EQ. 2 ) THEN
         WRITE( IUOUTH, '(A,A)' ) ' Stom._resp_to: ', 'leaf_air_VPD'
      ELSEIF( IVPDRESP .EQ. 3 ) THEN
```

```
         WRITE( IUOUTH, '(A,A)' ) ' Stom._resp_to: ', 'leaf_surf_VPD'
      ENDIF
      WRITE( IUOUTH, '(A,F8.5)') ' GSMAX:', GSMAX
      WRITE( IUOUTH, '(A,F8.5)') ' CD1:', CD1
      WRITE( IUOUTH, '(A,F8.5)') ' CD2:', CD2
      WRITE( IUOUTH, '(A,F8.5)') ' CD3:', CD3
      WRITE( IUOUTH, '(A,F8.5)') ' CD4:', CD4
      WRITE( IUOUTH, '(A,F8.5)') ' CD5:', CD5
      WRITE( IUOUTH, '(A,F8.5)') ' CD6:', CD6
      WRITE( IUOUTH, '(A, I4)') ' DELTMIN: ', DELTMIN

      FILOUTC = ' '
      WRITE( FILOUTC(1:IPP-1), '(A)') FILOUTH(1:IPP-1)
      WRITE( FILOUTC(IPP:IPP+4), '(A)') 'C.CSV'
      IF( .NOT. command ) THEN
         CALL INPUTT( ' Output file carbon ', FILOUTC )
      ENDIF
*VAX*
*       OPEN( UNIT = IUOUTC, FILE = FILOUTD, STATUS = 'NEW',
*     &      CARRIAGECONTROL = 'LIST' )
*PC*
      OPEN( UNIT = IUOUTC, FILE = FILOUTC, STATUS = 'UNKNOWN' )
      WRITE( IUOUTC, '(A,A)') ' FILE: ', FILOUTC
      WRITE( IUOUTC, '(A,A)') ' CREATED_BY: ', 'INTKAM'
      WRITE( IUOUTC, '(A, A)') ' EXPRNT:', EXPRNT
      WRITE( IUOUTC, '(A,A1)')  ' RUN: ', RUNSTRING

      FILOUTW = ' '
      WRITE( FILOUTW(1:IPP-1), '(A)') FILOUTH(1:IPP-1)
      WRITE( FILOUTW(IPP:IPP+4), '(A)') 'W.CSV'
      IF( .NOT. command ) THEN
         CALL INPUTT( ' Output file water ', FILOUTW )
      ENDIF
*VAX*
*       OPEN( UNIT=IUOUTW,FILE = FILOUTW, STATUS='NEW',
*     &      RECL = 250, CARRIAGECONTROL = 'LIST')
*PC*
      OPEN( UNIT=IUOUTW,FILE = FILOUTW, STATUS='UNKNOWN')
      WRITE(IUOUTW, '(A,F6.0)') ' DAY:', DAYNR
      WRITE(IUOUTW, '(A,A1)') ' RUN:', RUNSTRING
* ==============================================================
      ELSEIF( IOPHASE .EQ. 4 ) THEN

* Read variables from data file
       CALL READDT2( INI, IUDAT, FMT, YEAR_COL, DAYNR_COL, HOUR_COL,
     &               NR_OF_VAR, TMIN80,
     &               DATA_TMIN80, DATA_DAYNR, DATA_DAYMIN, VAR,
     &               IUNERR )

       IF( IUNERR .EQ. -1 ) THEN
          FILE_END = .TRUE.
          RETURN
       ENDIF

* Give specific variables a value if possible
      IF( GLRADO_FOUND ) THEN
         GLRADO = VAR( GLRADO_COL )
      ENDIF
      IF( CO2AIR_FOUND ) THEN
         CO2AIR = VAR( CO2AIR_COL )
      ENDIF
      IF( TEMPair_FOUND ) THEN
         TEMPair = VAR( TEMPair_COL )
      ENDIF
      IF( VPDAIR_FOUND ) THEN
         VPDAIR = VAR( VPDAIR_COL )
      ENDIF
      IF( PHOT_MEAS_FOUND ) THEN
         PHOT_MEAS = VAR( PHOT_MEAS_COL )
      ENDIF
      IF( TRAN_MEAS_FOUND ) THEN
         TRAN_MEAS = VAR( TRAN_MEAS_COL )
      ENDIF
      IF( TPIPE_FOUND ) THEN
         TPIPE = VAR( TPIPE_COL )
      ENDIF
      IF( TROOF_FOUND ) THEN
```

```
         TROOF = VAR( TROOF_COL )
      ENDIF
      IF( TEMPAIR_OUT_FOUND ) THEN
         TEMPAIR_OUT = VAR( TEMPAIR_OUT_COL )
      ENDIF
      IF( SCREEN_FOUND ) THEN
         SCREEN = VAR( SCREEN_COL )
      ELSE
         SCREEN = 0.
      ENDIF

      ENDIF
* ENDIF IOPHASE 4
* ================================================================

      RETURN

      END
*************************************************************************
* SUBPROGRAM: CANOPF
* Type: SUBROUTINE
* Purpose: calculation of canopy transpiration, energy balance
*          and crop gross photosynthesis.
* Description:
*  The canopy is divided in several layers, according to
*  the Gaussian integration. The energy balance of each leaf layer
*  is calculated with the Penman-Monteith equation.
*  Leaf conductances are calculated
*  independent from rate of leaf photosynthesis. Leaf photosynthesis is
*  calculated with the summary model of leaf photosynthesis
*
* Control variables :  INI
*
* Input:
*  PARDIF   : (R4)  flux diffuse PAR                           [J m-2 s-1]
*  PARDIR   : (R4)  flux direct PAR                            [J m-2 s-1]
*  NIRDIF   : (R4)  flux diffuse NIR                           [J m-2 s-1]
*  NIRDIR   : (R4)  flux direct NIR                            [J m-2 s-1]
*  UVDIF    : (R4)  flux diffuse UV                            [J m-2 s-1]
*  UVDIR    : (R4)  flux direct UV                             [J m-2 s-1]
*  ELEVN    : (R4)  solar elevation                            [radians]
*  SINELV   : (R4)  sine of solar elevation                    [-]
*  LAI      : (R4)  Leaf Area Index                            [-]
*  KDIFBL   : (R4)  extinction coeff. canopy with black leaves [-]
*  SCP      : (R4)  scattering coeff. of leaves for PAR        [-]
*  SCN      : (R4)  scattering coeff. of leaves for NIR        [-]
*  CO2air   : (R4)  CO2 concentration of greenhouse air        [mul l-1]
*  TEMPAIR  : (R4)  temperature of greenhouse air              [oC]
*  VPair    : (R4)  vapour pressure greenhouse air             [kPa]
*  VPDair   : (R4)  vapour pressure deficit of greenhouse air  [kPa]
*  TPIPE    : (R4)  temperature of heating pipes               [oC]
*  SSPT     : (R4)  specific surf. of heating pipes above canopy [-]
*  SSPB     : (R4)  specific surf. of heating pipes below canopy [-]
*  TROOF    : (R4)  temperature of greenhouse cover            [oC]
*  TGROUND   : (R4)  temperature of greenhouse floor            [oC]
*  PHOTRED  : (R4)  factor for reduction of photosynth. capacities
*                   with depth in canopy                        [-]
*  GSin     : (R4)  initial estimate for stomatal conductance  [m s-1]
*  RTHRAD   : (R4)  resistance for thermal radiation at top of can. [s m-1]
*  REFGR    : (R4)  reflection coefficient of ground surface   [-]
*
* Output:
*  PGROS   : (R4)  canopy gross photosynthesis              [mg CO2 m-2 s-1]
*  TRANSP  : (R4)  canopy transpiration                     [mg H2O m-2 s-1]
*  GSTOT   : (R4)  total canopy conductance (sum of stom. cond.) [m s-1]
*  GLTOT   : (R4)  total canopy conductance
*                              (sum of stom. + cut. cond.) [m s-1]
*  PARABS    : (R4)  PAR absorbed by canopy                  [J m-2 s-1]
*  PARDIRTOT : (R4)  direct PAR absorbed by canopy           [J m-2 s-1]
*  NIRABS    : (R4)  NIR absorbed by canopy                  [J m-2 s-1]
*  NETRADABS : (R4)  net radiation of canopy                 [J m-2 s-1]
*
*  Subprograms called:
*          LFTRAN, LONGRAD, FARPHOT
*
*
*  Common blocks:
*    ENERGY_EXCH1, ENERGY_EXCH2, ENERGY_EXCH3, FARQ_PAR4
```

```
*     GENCOM
*
*  Comment:
*-- This routine is similar to subroutine CANOP2; that routine uses
*   subroutine LPHOT for calculation of leaf gross photosynthesis
*   A decrease in the photosynthetic capacity can be assumed
*-- Extinction and absorption of UV-radiation are calculated
*   on behalf of the calculation of total absorbed radiation. UV absorbed
*   by leaves is not used in the energy balance, as it is assumed to cancel
*   out the energy used by the photosynthesis process.
*-- An iteration can be done inside SUBROUTINE LFTRAN to find the equilibrium
*   leaf surface VPD or leaf-air VPD. The equilibrium conditions are stored
*   in arrays so that in a next call to CANOP2 iteration can start with
*   previous conditions
**************************************************************************
      SUBROUTINE CANOPF( INI, HOUR, PARDIF, PARDIR, NIRDIF,
     &   NIRDIR, UVDIF, UVDIR, ELEVN,
     &   LAI, KDIFBL, KDIF, SCP, SCN, CO2air, TEMPAIR, VPDair,
     &   SSPT, SSPB, TPIPE, TROOF, TGROUND,
     &   PHOTRED,
     &   GSin, RTHRAD, REFGR,
     &   PARABS, PARDIRTO, NIRABS, UVABS, NETRAD,
     &   GSTOT, GLTOT,
     &   PGROS, TRANSP
     & )

      IMPLICIT REAL(A-Z)

      INTEGER I, I2, ISN, IN, J, L
      INTEGER IGAUSS
      REAL XGAUSS(3),WGAUSS(3)

*     PO number of leaf classes; POL number of canopy layers
      INTEGER PO, POL
      PARAMETER( PO = 5, POL = 3 )
      DIMENSION GS_PROF( PO, POL )
      DIMENSION VPDS_PROF( PO, POL ), VPDLA_PROF( PO, POL )
      DIMENSION TL_PROF( PO, POL )

      LOGICAL LIGHT
      COMMON /GENCOM/ LIGHT

      LOGICAL INI

      COMMON /ENERGY_EXCH1/ TCANOP
      COMMON /ENERGY_EXCH2/ HF_CR, HF_SC, HF_PC
      COMMON /ENERGY_EXCH3/ HFCRTOT, HFSCTOT, HFPCTOT, CONVH

      COMMON /FARQ_PAR5/ VCMAX250, JMAX250, RD250, THETA,
     &                   LGHTCON, KC25, KO25

      SAVE

* Auxillary variables for Gaussian integration
      DATA XGAUSS /0.1127, 0.5000, 0.8873/
      DATA WGAUSS /0.2778, 0.4444, 0.2778/
      DATA IGAUSS /3/

      IF( INI ) THEN
          TCANOP = TEMPAIR
          DO I = 1, IGAUSS
             DO J = 1, PO
                GS_PROF( J, I ) = GSIN
                VPDS_PROF( J, I ) = VPDAIR
                VPDLA_PROF( J, I ) = VPDAIR
                TL_PROF( J, I ) = TEMPAIR
             END DO
          END DO
      ENDIF

*---- Heat fluxes between a leaf and pipes, roof and ground, repectively
*     when not obscured by other leaves
*     Heat fluxes assuming leaves are at air temperature
*     Pipes and Canopy; pipes at top and at bottom of canopy
      HF_PC_T = LONGRAD( SSPT, TPIPE, TEMPAIR )
      HF_PC_B = LONGRAD( SSPB, TPIPE, TEMPAIR )
```

```
*     Canopy and Roof
      HF_CR = LONGRAD( 1., TEMPAIR, TROOF )

*     Ground and Canopy
*     positive when TGROUND > TCANOP
      HF_SC = LONGRAD( 1., TGROUND, TEMPAIR )

*     Set variables to zero
      PGROS = 0.
      E = 0.
      GSTOT = 0.
      GLTOT = 0.
      TCANOP = 0.
      PARABS = 0.
      NIRABS = 0.
      CONVH = 0.

*     When there is no diffuse PAR go to 'night period'
      IF( .NOT. LIGHT ) GOTO 800

*     Prevent math overflow
      SINELV = AMAX1( 0.02, SIN(ELEVN) )

*---------------------------------------------------------------------
*     Absorption of radiation in upper leaf layer
*---------------------------------------------------------------------

*     Thermal radiation from upper layer of Canopy towards Roof [J m-2 s-1]
      HFCR = KDIFBL * HF_CR

*     Extinction coefficients of canopy for diffuse radiation (PAR and
*     NIR)
      SQP = SQRT( 1.0 - SCP )
      KDIF = KDIFBL * SQP
      SQN = SQRT( 1.0 - SCN )
      KDIFN = KDIFBL * SQN

*---  Direct light: average projection and range of projections
      OAV = 0.3 + (0.7 - 0.3)* SINELV
      RNG = 0.9 + 0.05 * SIN( 2. * ELEVN )

*     Reflection coefficient of canopy for diffuse PAR and NIR radiation
      REFHP = (1.0-SQP) / (1.0+SQP)
      REFHN = (1.0-SQN) / (1.0+SQN)

*     Extinction coefficient of canopy for direct PAR and NIR radiation
*     Clustering factor
      CLUSTF = KDIF / (KDIFBL * SQP )
      KDIRBL = OAV / SINELV * CLUSTF
      KDIR = KDIRBL * SQP
      KDIRN = KDIRBL * SQN

*     Reflection coefficient of canopy for direct PAR and NIR radiation
      REFHPD = REFHP * 2.0 * OAV / ( OAV + SINELV )
      REFHND = REFHN * 2.0 * OAV / ( OAV + SINELV )

*     Radiation reflected by ground surface
      PAR_REF = REFGR *
     & ( (1.-REFHP) * EXP( -KDIF * LAI ) * PARDIF
     & + (1.-REFHPD) * EXP( -KDIR * LAI ) * PARDIR )

      NIR_REF = REFGR *
     & ( (1.-REFHN) * EXP( -KDIFN * LAI ) * NIRDIF
     & + (1.-REFHND) * EXP( -KDIRN * LAI ) * NIRDIR )

      UV_REF = REFGR *
     & ( (1.-REFHP) * EXP( -KDIF * LAI ) * UVDIF
     & + (1.-REFHPD) * EXP( -KDIR * LAI ) * UVDIR )

*=====================================================================

*     Absorption of diffuse PAR and NIR radiation in upper layer [J m-2 s-1]
      PAR_DFT = KDIF * PARDIF * (1.-REFHP)
      NIR_DFT = KDIFN * NIRDIF * (1.-REFHN)

*     Absorption of total direct radiation (PAR and NIR) in upper
*     layer [J m-2 s-1]
```

```
      PARDIR_TT = (1.0-REFHPD) * PARDIR * KDIR
      NIRDIR_TT = (1.0-REFHND) * NIRDIR * KDIRN

*     Absorption of direct component of direct radiation (PAR and NIR)
*     in upper layer [J m-2 s-1]
      PARDIR_DT = (1.0-SCP) * PARDIR * KDIRBL
      NIRDIR_DT = (1.0-SCN) * NIRDIR * KDIRBL

*     Absorption of direct radiation (PAR and NIR) by leaves
*     perpendicular on direct beam in upper layer [J m-2 s-1]
      SUNPER = PARDIR * (1.0-SCP) / SINELV
      NSUNPER = NIRDIR * (1.0-SCN) / SINELV

*-----------------------------------------------------------------------
*     Gaussian integration over depth  of canopy by selecting IGAUSS
*     (three) different LAI's and computing absorption of radiation,
*     assimilation, transpiration and leaf temperature at these LAI
*     levels.
*-----------------------------------------------------------------------

      DO 100 I = 1, IGAUSS

*     Selecting of depth of canopy
      LAIC = LAI * XGAUSS(I)

*     Decrease in photosynthetic capacities with canopy depth
      PHOTCOR = ( 1.- XGAUSS(I) ) * (1. - PHOTRED ) + PHOTRED
      VCMAX25 = VCMAX250 * PHOTCOR
      JMAX25 = JMAX250 * PHOTCOR
      RD25 = RD250 * PHOTCOR

*     Fraction SunLit Leaf Area
      SLLA = EXP(-KDIRBL * LAIC)

*     Thermal resistance for leaf layer; for radiation from above
*     and below canopy
      RRAD_TOP = RTHRAD / EXP( -KDIFBL * LAIC )
      RRAD_BOT = RTHRAD / EXP( -KDIFBL * (LAI - LAIC) )

*     Thermal radiation per Layer from canopy towards roof [J m-2 s-1]
      HFCR_L = HF_CR * EXP(-KDIFBL*LAIC)

*     Absorption of thermal radiation (Heat Flow) per leaf Layer
*     from heating Pipes towards Canopy [J m-2 s-1]
      HFPC_L = HF_PC_T * KDIFBL * EXP(-KDIFBL*(LAIC))
     &         + HF_PC_B * KDIFBL * EXP(-KDIFBL*(LAI-LAIC))

*     Absorption of thermal radiation (Heat Flow) per leaf Layer
*     from Soil towards Canopy [J m-2 s-1]
      HFSC_L = HF_SC * KDIFBL * EXP(-KDIFBL*(LAI-LAIC) )

*     Absorption of PAR and NIR radiation per leaf layer [J m-2 s-1]
*     Diffuse radiation
      PAR_DF = PAR_DFT * EXP(-KDIF*LAIC) +
     &     PAR_REF * KDIF * EXP( -KDIF*(LAI-LAIC) )
      NIR_DF = NIR_DFT * EXP(-KDIFN*LAIC) +
     &     NIR_REF * KDIFN * EXP( -KDIFN*(LAI-LAIC) )

*     Total direct radiation
      PARDIR_T = PARDIR_TT * EXP(-KDIR*LAIC)
      NIRDIR_T = NIRDIR_TT * EXP(-KDIRN*LAIC)

*     Direct component of direct radiation
      PARDIR_D = PARDIR_DT * SLLA
      NIRDIR_D = NIRDIR_DT * SLLA

*---- Shaded leaves
*     Absorption of PAR and NIR radiation by shaded leaves per layer
*     [J m-2 s-1]
      PAR_SHD = PAR_DF + PARDIR_T - PARDIR_D
      NIR_SHD = NIR_DF + NIRDIR_T - NIRDIR_D

*     Transpiration of shaded leaves per layer [mg m-2 s-1]
      CALL LFTRAN( PAR_SHD, NIR_SHD, HFPC_L, HFSC_L, HFCR_L,
     &             TEMPAIR, RRAD_TOP, RRAD_BOT, VPDair, CO2air,
     &             LAIC,
     &             GS_PROF(1,I), GL_SHD,
```

```
     &              TL_PROF(1,I), VPDS_PROF(1,I), VPDLA_PROF(1,I),
     &              E_SHD, CONVH_SHD
     &              )

      TL_SHD = TL_PROF(1,I)
      GS_SHD = GS_PROF(1,I)
      VPDS_SHD = VPDS_PROF(1,I)
      VPDLA_SHD = VPDLA_PROF(1,I)

*     Leaf photosynthesis [mg m-2 s-1]
      CALL FARPHOT( PAR_SHD, CO2AIR, TL_SHD, GS_SHD,
     &             KC25, KO25, VCMAX25, JMAX25, RD25, THETA, LGHTCON,
     &                        CO2i, PGR_SHD, PN )

*---- Sunlit leaves

      E_SUN = 0.
      TL_SUN = 0.
      PAR_SUN = 0.
      NIR_SUN = 0.
      GS_SUN = 0.
      GL_SUN = 0.
      CONVH_SUN = 0.

      PGR_SUN = 0.

*     Gaussian integration over leaf angles by selecting IGAUSS
*     (three) different angles at a specified LAI level and computing
*     absorption of radiation, assimilation, transpiration and leaf
*     temperature at these leaf angles.
      DO 300 ISN = 1, IGAUSS
*        Absorption of PAR and NIR radiation per leaf angle [W.m-2]
         PAR_S = PAR_SHD + ( OAV + RNG * (XGAUSS(ISN)-0.5) ) * SUNPER
         NIR_S = NIR_SHD + ( OAV + RNG * (XGAUSS(ISN)-0.5) ) * NSUNPER

*        Transpiration of sunlit leaves per layer [mg m-2 s-1]
         CALL LFTRAN( PAR_S, NIR_S, HFPC_L, HFSC_L, HFCR_L,
     &     TEMPair, RRAD_TOP, RRAD_BOT, VPDair, CO2air,
     &     LAIC,
     &     GS_PROF(1+ISN,I), GL_S,
     &     TL_PROF(1+ISN,I), VPDS_PROF(1+ISN,I), VPDLA_PROF(1+ISN,I),
     &     E_S, CONVH_S
     &     )

         TL_S = TL_PROF(1+ISN,I)
         GS_S = GS_PROF(1+ISN,I)
         VPD_S = VPDS_PROF(1+ISN,I)
         VPDLA_S = VPDLA_PROF(1+ISN,I)

*        Leaf photosynthesis [mg m-2 s-1]
         CALL FARPHOT( PAR_S, CO2AIR, TL_S, GS_S,
     &                KC25, KO25, VCMAX25, JMAX25, RD25, THETA, LGHTCON,
     &                CO2i, PGR_S, PN )

*        Calculate mean values over leaf angle distribution
         E_SUN = E_SUN + E_S * WGAUSS(ISN)
         TL_SUN = TL_SUN + TL_S * WGAUSS(ISN)
         PAR_SUN = PAR_SUN + PAR_S * WGAUSS(ISN)
         NIR_SUN = NIR_SUN + NIR_S * WGAUSS(ISN)
         CONVH_SUN = CONVH_SUN + CONVH_S * WGAUSS(ISN)
*        Conductivity of sunlit leaves
         GS_SUN = GS_SUN + GS_S * WGAUSS(ISN)
         GL_SUN = GL_SUN + GL_S * WGAUSS(ISN)

         PGR_SUN = PGR_SUN + PGR_S * WGAUSS( ISN )

300   CONTINUE

*     Totals of shaded and sunlit leaves per leaf layer
      E_L = SLLA * E_SUN + (1.0-SLLA) * E_SHD
      TL_L = SLLA * TL_SUN + (1.0-SLLA) * TL_SHD
      PAR_L = SLLA * PAR_SUN + (1.0-SLLA) * PAR_SHD
      NIR_L = SLLA * NIR_SUN + (1.0-SLLA) * NIR_SHD
      GS_L = SLLA * GS_SUN + (1.-SLLA) * GS_SHD
      GL_L = SLLA * GL_SUN + (1.-SLLA) * GL_SHD
      CONVH_L = SLLA * CONVH_SUN + (1.-SLLA) * CONVH_SHD
```

```
      PG_L = SLLA * PGR_SUN +(1.0-SLLA) * PGR_SHD

*       WRITE( 99, '(5F9.4)')
*     &  LAIC, PAR_SUN, GS_SUN, PGR_SUN, E_SUN

*---- Calculate mean values over all leaf layers
      E      = E + E_L * WGAUSS(I)
      TCANOP = TCANOP + TL_L * WGAUSS(I)
      GSTOT  = GSTOT + GS_L * WGAUSS(I)
      GLTOT  = GLTOT + GL_L * WGAUSS(I)
      PARABS  = PARABS + PAR_L * WGAUSS(I)
      NIRABS  = NIRABS + NIR_L * WGAUSS(I)
      CONVH  = CONVH + CONVH_L * WGAUSS(I)

      PGROS  = PGROS + PG_L * WGAUSS(I)
100   CONTINUE
*     End LAI loop

*     Absorbed UV radiation
      UVDIFAB = UVDIF * (1.-REFHP) * ( 1. - EXP( -KDIF * LAI ))
     &           + UV_REF * (1. - EXP( - KDIF * LAI ))
      UVDIRAB = UVDIR *   (1.-REFHPD) * ( 1. - EXP(  -KDIR * LAI ))
      UVABS = UVDIFAB + UVDIRAB

      GOTO 999


800   CONTINUE
*-------------------------------------------------------------------
*     Night period
*-------------------------------------------------------------------

*---- Gaussian integration over depth  of canopy by selecting IGAUSS
*     (three) different LAI's and computing absorption of radiation,
*     assimilation, transpiration and leaf temperature at these LAI
*     levels.

      DO 900 IN = 1, IGAUSS

*     Selecting of depth of canopy
      LAIC = LAI * XGAUSS(IN)

*     Thermal resistance for leaf layer
      RRAD_TOP = RTHRAD / EXP( -KDIFBL * LAIC )
      RRAD_BOT = RTHRAD / EXP( -KDIFBL * (LAI - LAIC) )

*     Thermal radiation per Layer from Canopy towards Roof [J m-2 s-1]
      HFCR_L= HF_CR * KDIFBL * EXP(-KDIFBL*LAIC)

*     Absorption of thermal radiation (Heat Flow) per leaf Layer
*     from heating Pipes towards Canopy [J m-2 s-1]
      HFPC_L = HF_PC_T * KDIFBL * EXP(-KDIFBL*(LAIC))
     &         + HF_PC_B * KDIFBL * EXP(-KDIFBL*(LAI-LAIC))

*     Absorption of thermal radiation (Heat Flow) per leaf Layer
*     from Soil towards Canopy [J m-2 s-1]
      HFSC_L = HF_SC * KDIFBL * EXP(-KDIFBL*(LAI-LAIC))

      CALL LFTRAN( PAR_L, NIR_L, HFPC_L, HFSC_L, HFCR_L, TEMPAIR,
     &            RRAD_TOP, RRAD_BOT, VPDair, CO2air,
     &            LAIC,
     &            GS_PROF(5,IN), GL_L,
     &            T1_PROF(5,IN), VPDS_PROF(5,IN), VPDLA_PROF(5,IN),
     &            E_L,
     &            CONVH_L
     & )

         TL_L = TL_PROF(5,IN)
         GS_L = GS_PROF(5,IN)

*        Calculate mean values over all leaf layers
         E      = E + E_L * WGAUSS(IN)
         GSTOT  = GSTOT + GS_L * WGAUSS(IN)
         GLTOT  = GLTOT + GL_L * WGAUSS(IN)
         TCANOP  = TCANOP + TL_L * WGAUSS(IN)
         CONVH  = CONVH + CONVH_L * WGAUSS(IN)
```

```
900   CONTINUE
*     End night loop

999   CONTINUE

*---- Multiplication of values per layer by LAI to obtain total
*     values of all layers (the canopy)
      TRANSP = E          * LAI
      GSTOT = GSTOT       * LAI
      GLTOT = GLTOT       * LAI
      PARABS = PARABS     * LAI
      NIRABS = NIRABS     * LAI
      CONVH = CONVH       * LAI

      PGROS = PGROS * LAI

*---- Absorbed radiation from greenhouse cover, ground and pipes
*     Actual heat fluxes, taking account of canopy temperature
      HF_PCTX = LONGRAD( SSPT, TPIPE, TCANOP )
      HF_PCBX = LONGRAD( SSPB, TPIPE, TCANOP )
      HF_CRX = LONGRAD( 1., TCANOP, TROOF )
      HF_SCX = LONGRAD( 1., TGROUND, TCANOP )
      HFPCTOT = (1.-EXP( -LAI * KDIFBL)) * HF_PCTX
     &         + (1.-EXP( -LAI * KDIFBL)) * HF_PCBX
      HFSCTOT = (1.-EXP( -LAI * KDIFBL)) * HF_SCX
      HFCRTOT = (1.-EXP( -LAI * KDIFBL)) * HF_CRX

      NETRAD = HFPCTOT + HFSCTOT - HFCRTOT + PARABS + NIRABS + UVABS

      RETURN
      END
**************************************************************************
* SUBPROGRAM: DMPROD2
* Type: SUBROUTINE
* Date: June 1994
* Author: H. Gijzen
* Purpose:
*     calculation of dry matter production of leaves, stems, roots and
*     fruits of greenhouse crop from daily total of gross assimilation
* Description:
*    Maintenance respiration is subtracted from daily gross
*    photosynthesis, and resulting net assimilates are converted to
*    dry matter. Coefficients for dry matter partitioning are used
*    to calculate dry matter production of individual organs
*
* Origin: SUCROS87 by Spitters et al. (1989)
*
* Control variables: ITASK, TERMNL
* Init variables: ITASK
* Timer variables: DAYNR, DELT
*
* Input:
*  ITASK      : (I4)  control variable for initialization (ITASK=1),
*                     rate calculation (2) and integration (3)       [-]
*  DAYNR      : (R4)  day number (Jan 1st = 1)                    [-]
*  DELT       : (R4)  time step                                   [d]
*  WLVI       : (R4)  initial leaf dry weight of crop             [g m-2]
*  WSTI       : (R4)  initial stem dry weight of crop             [g m-2]
*  WRTI       : (R4)  initial root dry weight of crop             [g m-2]
*  WSOI       : (R4)  initial dry weight of storage organs        [g m-2]
*  FLV        : (R4)  dry matter partitioning to leaves           [-]
*  FST        : (R4)  dry matter partitioning to stems            [-]
*  FRT        : (R4)  dry matter partitioning to roots            [-]
*  FSO        : (R4)  dry matter partitioning to storage organs [-]
*  ASRQLV     : (R4)  assimilate requirement leaves        [g CH2O g dm-1]
*  ASRQST     : (R4)  assimilate requirement stems         [g CH2O g dm-1]
*  ASRQRT     : (R4)  assimilate requirement roots         [g CH2O g dm-1]
*  ASRQSO     : (R4)  assimilate requirement storage org.  [g CH2O g dm-1]
*  DTGA       : (R4)  daily total gross assimilation        [g CO2 m2 d-1]
*  DMAINT     : (R4)  daily total of maint. costs           [g CH2O m2 d-1]
*
* Output:
*  GLV        : (R4)  rate of DM increase of leaves          [g m-2 d-1]
*  GST        : (R4)  rate of DM increase of stems           [g m-2 d-1]
*  GRT        : (R4)  rate of DM increase of roots           [g m-2 d-1]
*  GSO        : (R4)  rate of DM increase of stor. org.      [g m-2 d-1]
*  GTW        : (R4)  rate of DM increase of crop            [g m-2 d-1]
```

```
*  WLV       : (R4)  dry weight of leaves                     [g DM m-2]
*  WST       : (R4)  dry weight of stems                      [g DM m-2]
*  WRT       : (R4)  dry weight of roots                      [g DM m-2]
*  WSO       : (R4)  dry weight of storage organs             [g DM m-2]
*  CWLV      : (R4)  cumulative dry weight of leaves          [g DM m-2]
*  CWST      : (R4)  cumulative dry weight of stems           [g DM m-2]
*  CWRT      : (R4)  cumulative dry weight of roots           [g DM m-2]
*  CWSO      : (R4)  cumulative dry weight of storage organs  [g DM m-2]
*  CTWT      : (R4)  cumulative dry weight of crop            [g DM m-2]
**********************************************************************
      SUBROUTINE DMPROD2( ITASK, DAYNR, DELT,
     &            WLVI, WSTI, WRTI, WSOI,
     &            FLV, FST, FRT, FSO,
     &            ASRQLV, ASRQST, ASRQRT, ASRQSO,
     &            DTGA, DMAINT,
     &            GLV, GST, GRT, GSO, GTW,
     &            WLV, WST, WRT, WSO, TWT,
     &            CWLV, CWST, CWRT, CWSO, CTWT )
      IMPLICIT REAL(A-Z)
      INTEGER ITASK, ITOLD

      SAVE

      DATA ITOLD /4/

*     The task that the subprogram should do (ITASK) is compared with
*     the task done during the previous call (ITOLD)
*     Only certain combinations are allowed:
*
*     New task                 Old task
*     --------                 --------
*     initialization           terminal
*     integration              rate calculation
*     rate calculation         initialization, integration
*     terminal                 any task
*
*    Note: integration after initialization is strictly correct,
*    but will not result in any calculations

      IF( ITOLD.EQ.1 .AND. ITASK.EQ.3 ) THEN
         ITOLD = ITASK
         RETURN
      ENDIF

      IF( ITASK .EQ. 1 ) THEN

*=====================
*       initialization
*=====================

         WLV = WLVI
         WST = WSTI
         WRT = WRTI
         WSO = WSOI
         TWT = WLV + WST + WRT + WSO

*        Cumulative weights
         CWLV = WLVI
         CWST = WSTI
         CWRT = WRTI
         CWSO = WSOI
         CTWT = CWLV + CWST + CWRT + CWSO

      ELSEIF( ITASK .EQ. 2 ) THEN
*==============================
*        rate calculation section
*==============================

*---- Daily assimilates, conversion of CO2 to sugars [g CH2O m-2 day-1]
      DTASS = DTGA * 30./44.

*---- Assimilate requirements for dry matter conversion
*       [g CH2O/g dry matter]
      ASRQ = FLV*ASRQLV + FST*ASRQST + FSO*ASRQSO + FRT*ASRQRT

*---- Rate of growth  [g DM m-2 day-1]
*     take care of assimilates needed in following days (negative
```

```
*     assimilate reserves)
      NETASM = DTASS - DMAINT - RESERV
      IF( NETASM .LT. 0. ) THEN
         RESERV = - NETASM
         NETASM = 0.
      ELSE
         RESERV = 0.
      ENDIF
      GTW = NETASM / ASRQ
      GLV = GTW * FLV
      GST = GTW * FST
      GSO = GTW * FSO
      GRT = GTW * FRT

      ELSEIF (ITASK .EQ. 3) THEN

*===========================
*         integration section
*===========================

*---- Dry weights of leaves stems, storage organs, roots
*     and total biomass (g DM m-2) as integrals of growth rates.
*     Note that no biomass is removed.
      WLV    = WLV + GLV * DELT
      WST    = WST + GST * DELT
      WRT    = WRT + GRT * DELT
      WSO    = WSO + GSO * DELT
      TWT    = WLV + WST + WRT + WSO

      CWLV    = CWLV + GLV * DELT
      CWST    = CWST + GST * DELT
      CWRT    = CWRT + GRT * DELT
      CWSO    = CWSO + GSO * DELT
      CTWT    = CWLV + CWST + CWRT + CWSO

      ENDIF

      ITOLD = ITASK

      RETURN
      END
**********************************************************************
* Subprogram: FDIF_10M
* Type: REAL FUNCTION
* Purpose: calculation of fraction diffuse in global radiation from
*    atmospheric transmission, for 10 min-intervals
* Description: relation between fraction diffuse global radiation and
*    atmospheric transmission is that from De Jong. Parameters used
*    are obtained from fitting to 10 min. data of Naaldwijk in years
*    1990 and 1991.
*
* Input:
*  SOLARC   : (R4)  corrected solar constant                 [J m-2 s-1]
*  GLRADO   : (R4)  global radiation outside greenhouse      [J m-2 s-1]
*  SINELV   : (R4)  sine of solar elevation                  [-]
*
* Output:
*  FDIF_10M : (R4) fraction diffuse in global radiation      [-]
**********************************************************************
      REAL FUNCTION FDIF_10M( SOLARC, GLRADO, SINELV )
      IMPLICIT REAL (A-Z)

*      COMMON /PAR10MIN/ a,b,c,d

      a = 6.027
      b = 0.2756
      c = 0.4304
      D = 0.1384

      S0 = SOLARC * SINELV
      ATMTR = GLRADO / S0

      IF( ATMTR .LT. b ) THEN
          G1 = 0.
      ELSE
          G1 = (ATMTR-b) * (ATMTR-b)
      ENDIF
```

```
      IF( ATMTR .LT. c ) THEN
         G2 = 0.
      ELSE
         G2 = (ATMTR-c) * (ATMTR-c)
      ENDIF
      G = 1. - a * (G1-G2)
      H = d + (1.-d) * (1. - EXP(-0.1 / SINELV) )

      FDIF_10M = AMAX1( G, H )

      RETURN
      END
***********************************************************************
* Subprogram: LENER
*
* Purpose:
*  Calculation of leaf energy balance based on absorbed shortwave and
*  thermal radiation and stomatal conductance
*
* Description: leaf energy balance is calculated with Penman-Monteith
*   combination equation. The energy balance is calculated twice to
*   find the approximate equilibrium leaf temperature.
*
* Input:
*  PARABS   : (R4)  absorbed PAR                              [J m-2 s-1]
*  NIRABS   : (R4)  absorbed NIR                              [J m-2 s-1]
*  HFPC     : (R4)  thermal rad. from heating pipes           [J m-2 s-1]
*  HFSC     : (R4)  thermal rad. from ground                  [J m-2 s-1]
*  HFCR     : (R4)  thermal rad. to greenhouse cover          [J m-2 s-1]
*  RRAD_TOP: (R4)  resistance for thermal radiation coming
*                       from above canopy                         [s m-1]
*  RRAD_BOT: (R4)  resistance for thermal radiation coming
*                       from below canopy                         [s m-1]
*  TEMPair : (R4)  temperature of greenhouse air             [oC]
*  VPDair  : (R4)  vapour pressure deficit of greenhouse air  [kPa]
*  GS      : (R4)  stomatal conductance                      [m s-1]
*  RB      : (R4)  boundary layer resistance for vapour      [s m-1]
*
* Output:
*  TRANLEAF : (R4)  leaf transpiration                     [mg H2O m-2 s-1]
*  TLEAF    : (R4)  leaf temperature                       [oC]
*  CONV     : (R4)  convective heat loss from leaf         [J m-2 s-1]
***********************************************************************
      SUBROUTINE LENER( PARABS, NIRABS, HFPC, HFSC, HFCR, TEMPair,
     &                  GL, RB, RRAD_TOP, RRAD_BOT, VPDair,
     &                  TRANLEAF, TLeaf, CONV )
      IMPLICIT REAL(A-Z)

*     Evaporation energy of 1 mg of water [J mg-1]
      LABDA = 2.5

*     Leaf conductance (GL) and Resistance (RL) for water vapour
      RL = 1. / GL
*     Boundary layer Resistance for Heat
      RBH = RB / 0.93

*     Total heat conductance
      GTH = 1. / RBH + 1. / RRAD_TOP + 1. / RRAD_BOT

      RRAD = 1. / ( 1./RRAD_TOP + 1./RRAD_BOT )

*     Total Heat Resistance
      RBTH = 1. / GTH

*     Water vapour in air [kPa]
      ES = .6107*EXP(17.4*TEMPair/(TEMPair+239.))
*     To determine slope of ES-curve 1 oC higher
      ES1 = .6107*EXP(17.4*(TEMPair+1)/(TEMPair+1.+239.))
*     SLOPE of ES-curve [kPa oC-1]
      SLOPE = ES1-ES
*   Penman method to estimate transpiration
*     Volumetric heat capacity of air [J m-3 oC-1]
      RHOCP = 1200.
*     DRYing Power [kPa J m-2 s-1 oC-1]
      DRYP = VPDair * RHOCP / RBTH
*     PSYCHRometric constant [kPa oC-1]
      PSYCHR = 0.067
```

```
*     Auxiliary variable [kPa oC-1]
      GAMMAST = PSYCHR*(RB+RL) / RBTH
*     Energy for transpiration [J m-2 s-1]
      LE = ( SLOPE * (PARABS+NIRABS+HFPC+HFSC-HFCR) + DRYP )
     &     / (SLOPE + GAMMAST)

*     Thermal convection and radiation of a leaf [J m-2 s-1]
      THRAD = PARABS + NIRABS + HFPC + HFSC - HFCR - LE
*     Leaf Temperature [oC]
      Tleaf = TEMPair + THRAD * RBTH / RHOCP

*     Iteration

*     Water vapour in at leaf temperature [kPa]
      ES= .6107*EXP(17.4*Tleaf/(Tleaf+239.))
*     To determine slope of ES-curve 1 oC higher
      ES1= .6107*EXP(17.4*(Tleaf+1.)/(Tleaf+1.+239.))
*     SLOPE of ES-curve [kPa oC-1]
      SLOPE2 = ES1-ES
*     Mean SLOPE of ES-curve [kPa oC-1]
      SLOPE = ( SLOPE + SLOPE2 ) / 2.

*     Energy for transpiration [J m-2 s-1]
      LE = (SLOPE * (PARABS+NIRABS+HFPC+HFSC-HFCR)+DRYP) /
     &     (SLOPE + GAMMAST)

*     Transpiration in mg water m-2 s-1
      TRANLEAF = LE / LABDA

*     Thermal convection and radiation of a leaf [J m-2 s-1]
      THRAD = PARABS + NIRABS + HFPC + HFSC - HFCR - LE

*     Leaf Temperature [oC]
      Tleaf = TEMPair + THRAD * RBTH / RHOCP

*     Convective heat loss
      CONV = (Tleaf - TEMPair) * RHOCP / RBH

      RETURN
      END
*****************************************************************
* Subprogram: LFTRAN
* Purpose:
*  Calculation of leaf energy balance and transpiration
*
* Input:
*  PARABS    : (R4)  absorbed PAR energy flux                [J m-2 s-1]
*  NIRABS    : (R4)  absorbed NIR energy flux                [J m-2 s-1]
*  HFPC      : (R4)  thermal radiation pipe to leaf          [J m-2 s-1]
*  HFSC      : (R4)  thermal radiation ground to leaf        [J m-2 s-1]
*  HFCR      : (R4)  thermal radiation leaf to roof          [J m-2 s-1]
*  RRAD_TOP  : (R4)  resistance for thermal radiation coming
*                       from above canopy                    [s m-1]
*  RRAD_BOT  : (R4)  resistance for thermal radiation coming
*                       from below canopy                    [s m-1]
*  TEMPAIR   : (R4)  temperature of air                      [oC]
*  CO2AIR    : (R4)  CO2 concentration                       [mumol mol-1]
*  VPDAIR    : (R4)  Vapour Pressure Deficit of air          [kPa]
*  RB        : (R4)  boundary layer resistance for vapour    [s m-1]
*  RCUT      : (R4)  cuticula resistance for vapour          [s m-1]
*  GMAXD     : (R4)  maximal leaf conductance at night       [m s-1]
*  GNVPD     : (R4)  parameter for leaf surface VPD response
*                    of GMAXD                                [kPa-1]
*
* Output:
*  TRANLEAF  : (R4)  leaf transpiration                   [mg H2O m-2 s-1]
*  TLEAF     : (R4)  leaf temperature                        [oC]
*  CONV      : (R4)  convective heat loss from leaf          [J m-2 s-1]
*  GLEAF     : (R4)  leaf conductance                        [m s-1]
*  GS        : (R4)  stomatal conductance                    [m s-1]
*  VPDsurf   : (R4)  VPD at leaf surface                     [kPa]
*  VPDla     : (R4)  leaf-air VPD                            [kPa]
*
*
* Subprograms called: LENER, STOMRESP
*
* Comment:
```

```
*      LAIC is dummy input variable
*******************************************************************
      SUBROUTINE LFTRAN(
     &              PARabs, NIRabs, HFPC, HFSC, HFCR,
     &              TEMPair,
     &              RRAD_TOP, RRAD_BOT, VPDair, CO2air,
     &              LAIC,
     &              GS, Gleaf,
     &              Tleaf, VPDsurf, VPDla, TRANleaf, CONVH
     & )
      IMPLICIT REAL(A-Z)

      INTEGER I, NITER

      COMMON /LEAFPAR/ Rcut, Rb
      COMMON /LEAFPAR2/ Gmaxd, GNVPD
      COMMON /LEAFPAR3/ Gmaxda, GNVPDa

      COMMON /CLIMHUM/ VPair

      LOGICAL LIGHT
      COMMON /GENCOM/ LIGHT

      LOGICAL DO_AIR
      LOGICAL DO_LFAIR
      COMMON /PARSTOM2/ DO_AIR, DO_LFAIR

      DATA NITER, VPD_EPS /10, 0.2/

* ==========================================================================
      IF( .NOT. LIGHT ) THEN

         IF( DO_AIR ) THEN
*           Response of leaf conductance in the dark to air VPD
            Gleafd = Gmaxda * exp( - GNVPDa * VPDair )

*-----      Energy balance
            CALL LENER( PARabs, NIRabs, HFPC, HFSC, HFCR, TEMPair,
     &                 Gleafd, Rb, RRAD_TOP, RRAD_BOT, VPDair,
     &                 TRANleaf, TLeaf, CONVH )
            TRANleaf = AMAX1( 0.001, TRANleaf )

            VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )

*           Humidity at leaf surface
            VPDsurf = AMAX1( 0.01,
     &           (1./Gleafd) / (1./Gleafd + Rb ) * (VPleaf - VPair ) )

         ELSE
*           Response to leaf surface VPD
            DO I = 1, NITER

              VPDsurfi = VPDsurf

*           Response of leaf conductance in the dark to leaf surface VPD
              Gleafd = Gmaxd * exp( - GNVPD * VPDsurf )

*----         Energy balance
              CALL LENER( PARabs, NIRabs, HFPC, HFSC, HFCR, TEMPair,
     &                   Gleafd, Rb, RTHRAD, VPDair,
     &                   TRANleaf, TLeaf, CONVH )
              TRANleaf = AMAX1( 0.001, TRANleaf )

              VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )

*             Humidity at leaf surface
              VPDsurf = AMAX1( 0.01,
     &            (1./Gleafd) / (1./Gleafd + Rb ) * (VPleaf - VPair ) )

*             Exit loop when difference with previous value too small
              IF( ABS( VPDsurf - VPDsurfi ) .LT. VPD_eps ) THEN
                 GOTO 10
              ENDIF
            END DO
10          CONTINUE

      ENDIF
```

```
*      of DO_AIR
         Gleaf = Gleafd

       ELSE
* =====================================================================
*        LIGHT is TRUE

         Gcut = 1. / Rcut

         IF( DO_AIR ) THEN

*----       conductance in the dark, response to air-VPD
            Gleafd = Gmaxda * exp( - GNVPDa * VPDair )

*----       stomatal response
            CALL STOMRESP( PARabs, VPDair, CO2air, TEMPair,
     &                     Gcut, Gs, Gleaf )

            Gleaf = AMAX1( Gleafd, Gleaf )

*----       Energy balance
            CALL LENER( PARabs, NIRabs, HFPC, HFSC, HFCR, TEMPair,
     &                  Gleaf, Rb, RRAD_TOP, RRAD_BOT, VPDair,
     &                  TRANleaf, TLeaf, CONVH )
            TRANleaf = AMAX1( 0.001, TRANleaf )

*        WRITE( 99, '(20X, 5F9.4)') PARABS, NIRABS, GS, GLEAF, TRANLEAF

            VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )

*----      Humidity at leaf surface
            VPDsurf = AMAX1( 0.01,
     &            (1./Gleaf) / (1./Gleaf + Rb ) * (VPleaf - VPair ) )

         ELSE

            VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )

*           leaf-air VPD
            VPDla = VPleaf - VPair

            DO I = 1, NITER

              IF( DO_LFAIR ) THEN
                 VPD = VPDla
              ELSE
                 VPD = VPDsurf
              ENDIF

              VPDi = VPD

*----         conductance in the dark, response to leaf surface VPD
              Gleafd = Gmaxd * exp( - GNVPD * VPDsurf )

*----         stomatal response
              CALL STOMRESP( PARabs, VPDair, CO2air, TEMPair,
     &                     Gcut, Gs, Gleaf )

              Gleaf = AMAX1( Gleafd, Gleaf )

*-----        Energy balance
              CALL LENER( PARabs, NIRabs, HFPC, HFSC, HFCR, TEMPair,
     &                  Gleaf, Rb, RRAD_TOP, RRAD_BOT, VPDair,
     &                  TRANleaf, TLeaf, CONVH )
              TRANleaf = AMAX1( 0.001, TRANleaf )

              VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )

*             Leaf-air VPD
              VPDla = VPleaf - VPair

*             Humidity at leaf surface
              VPDsurf = AMAX1( 0.01,
     &            (1./Gleaf) / (1./Gleaf + Rb ) * (VPleaf - VPair ) )

              IF( DO_LFAIR ) THEN
                 VPD = VPDla
```

```
            ELSE
               VPD = VPDsurf
            ENDIF

*           Exit loop when difference with previous value too small
            IF( ABS( VPD - VPDi ) .LT. VPD_eps ) THEN
               GOTO 40
            ENDIF

           END DO
40         CONTINUE
        ENDIF
*       of DO_AIR
      ENDIF
*     of LIGHT
* ======================================================================

      RETURN
      END
************************************************************************
* SUBPROGRAM: MNRESP
*
* Input:
*  ITASK     : (I4)  control variable for initialization (ITASK=1),
*                    rate calculation (2), integration (3)
*                    terminal calculations (4) and resetting (5)      [-]
*  DELT      : (R4)  time step                                       [h]
*  WLV       : (R4)  dry weight of leaves                      [g DM m-2]
*  WST       : (R4)  dry weight of stems                       [g DM m-2]
*  WRT       : (R4)  dry weight of roots                       [g DM m-2]
*  WSO       : (R4)  dry weight of storage organs              [g DM m-2]
*  MAINLV    : (R4)  maint. costs leaves at 25 oC       [g CH2O g dm-1 d-1]
*  MAINST    : (R4)  maint. costs stems at 25 oC        [g CH2O g dm-1 d-1]
*  MAINRT    : (R4)  maint. costs roots at 25 oC        [g CH2O g dm-1 d-1]
*  MAINSO    : (R4)  maint. costs storage org. at 25 oC [g CH2O g dm-1 d-1]
*  Q10MN     : (R4)  Q10 maintenance respiration                 [-]
*  REFTMP    : (R4)  reference temperature maint. resp.          [oC]
*  TEMPAIR   : (R4)  temperature greenhouse air                  [oC]
*
* Output:
*  DMAINT    : (R4)  daily total of maint. costs          [g CH2O m-2 d-1]
************************************************************************
      SUBROUTINE MNRESP( ITASK, DELT, Q10MN, REFTMP,
     &           WLV, WST, WRT, WSO,
     &       MAINLV, MAINST, MAINRT, MAINSO,
     &       TEMPAIR, DMAINT )
      IMPLICIT REAL(A-Z)

      INTEGER ITASK

*     Initialization or resetting
      IF( ITASK .EQ. 1 .OR. ITASK .EQ. 5) THEN

         DMAINT = 0.

      ELSEIF( ITASK .EQ. 2 ) THEN

*----    Maintenance respiration [mg CH2O m-2 s-1]
*        86.4 converts g d-1 to mg s-1
         MAINTS = ( WLV*MAINLV + WST*MAINST + WSO*MAINSO + WRT*MAINRT )
     &            / 86.4
         TEFF  = Q10MN**( ((TEMPAIR-REFTMP)/10.) )
         MAINT = MAINTS * TEFF

      ELSEIF (ITASK .EQ. 3) THEN

*        3.6 converts mg s-1 to g h-1
         DMAINT = DMAINT + MAINT * DELT * 3.6

      ENDIF

      RETURN
      END
************************************************************************
* Subprogram: LENER
*
* Purpose:
```

```
*  Calculation of leaf energy balance based on absorbed shortwave and
*  thermal radiation and stomatal conductance
*
* Description: leaf energy balance is calculated with Penman-Monteith
*   combination equation. The energy balance is calculated twice to
*   find the approximate equilibrium leaf temperature.
*
* Input:
*  PARABS   : (R4)  absorbed PAR                            [J m-2 s-1]
*  NIRABS   : (R4)  absorbed NIR                            [J m-2 s-1]
*  HFPC     : (R4)  thermal rad. from heating pipes         [J m-2 s-1]
*  HFSC     : (R4)  thermal rad. from ground                [J m-2 s-1]
*  HFCR     : (R4)  thermal rad. to greenhouse cover        [J m-2 s-1]
*  RRAD_TOP: (R4)  resistance for thermal radiation coming
*                       from above canopy                       [s m-1]
*  RRAD_BOT: (R4)  resistance for thermal radiation coming
*                       from below canopy                       [s m-1]
*  TEMPair  : (R4)  temperature of greenhouse air           [oC]
*  VPDair   : (R4)  vapour pressure deficit of greenhouse air  [kPa]
*  GS       : (R4)  stomatal conductance                    [m s-1]
*  RB       : (R4)  boundary layer resistance for vapour    [s m-1]
*
* Output:
*  TRANLEAF : (R4)  leaf transpiration                     [mg H2O m-2 s-1]
*  TLEAF    : (R4)  leaf temperature                       [oC]
*  CONV     : (R4)  convective heat loss from leaf         [J m-2 s-1]
**********************************************************************
      SUBROUTINE LENER( PARABS, NIRABS, HFPC, HFSC, HFCR, TEMPair,
     &                  GL, RB, RRAD_TOP, RRAD_BOT, VPDair,
     &                  TRANLEAF, TLeaf, CONV )
      IMPLICIT REAL(A-Z)

*     Evaporation energy of 1 mg of water [J mg-1]
      LABDA = 2.5

*     Leaf conductance (GL) and Resistance (RL) for water vapour
      RL = 1. / GL
*     Boundary layer Resistance for Heat
      RBH = RB / 0.93

*     Total heat conductance
      GTH = 1. / RBH + 1. / RRAD_TOP + 1. / RRAD_BOT

      RRAD = 1. / ( 1./RRAD_TOP + 1./RRAD_BOT )

*     Total Heat Resistance
      RBTH = 1. / GTH

*     Water vapour in air [kPa]
      ES = .6107*EXP(17.4*TEMPair/(TEMPair+239.))
*     To determine slope of ES-curve 1 oC higher
      ES1 = .6107*EXP(17.4*(TEMPair+1)/(TEMPair+1.+239.))
*     SLOPE of ES-curve [kPa oC-1]
      SLOPE = ES1-ES
*   Penman method to estimate transpiration
*     Volumetric heat capacity of air [J m-3 oC-1]
      RHOCP = 1200.
*     DRYing Power [kPa J m-2 s-1 oC-1]
      DRYP = VPDair * RHOCP / RBTH
*     PSYCHRometric constant [kPa oC-1]
      PSYCHR = 0.067
*     Auxiliary variable [kPa oC-1]
      GAMMAST = PSYCHR*(RB+RL) / RBTH
*     Energy for transpiration [J m-2 s-1]
      LE = ( SLOPE * (PARABS+NIRABS+HFPC+HFSC-HFCR) + DRYP )
     &     / (SLOPE + GAMMAST)

*     Thermal convection and radiation of a leaf [J m-2 s-1]
      THRAD = PARABS + NIRABS + HFPC + HFSC - HFCR - LE
*     Leaf Temperature [oC]
      Tleaf = TEMPair + THRAD * RBTH / RHOCP

*     Iteration

*     Water vapour in at leaf temperature [kPa]
      ES= .6107*EXP(17.4*Tleaf/(Tleaf+239.))
*     To determine slope of ES-curve 1 oC higher
```

```
      ES1= .6107*EXP(17.4*(Tleaf+1.)/(Tleaf+1.+239.))
*     SLOPE of ES-curve [kPa oC-1]
      SLOPE2 = ES1-ES
*     Mean SLOPE of ES-curve [kPa oC-1]
      SLOPE = ( SLOPE + SLOPE2 ) / 2.

*     Energy for transpiration [J m-2 s-1]
      LE = (SLOPE * (PARABS+NIRABS+HFPC+HFSC-HFCR)+DRYP) /
     &     (SLOPE + GAMMAST)

*     Transpiration in mg water m-2 s-1
      TRANLEAF = LE / LABDA

*     Thermal convection and radiation of a leaf [J m-2 s-1]
      THRAD = PARABS + NIRABS + HFPC + HFSC - HFCR - LE

*     Leaf Temperature [oC]
      Tleaf = TEMPair + THRAD * RBTH / RHOCP

*     Convective heat loss
      CONV = (Tleaf - TEMPair) * RHOCP / RBH

      RETURN
      END
*****************************************************************
* Subprogram: LFTRAN
* Purpose:
*  Calculation of leaf energy balance and transpiration
*
* Input:
*  PARABS    : (R4)  absorbed PAR energy flux                [J m-2 s-1]
*  NIRABS    : (R4)  absorbed NIR energy flux                [J m-2 s-1]
*  HFPC      : (R4)  thermal radiation pipe to leaf          [J m-2 s-1]
*  HFSC      : (R4)  thermal radiation ground to leaf        [J m-2 s-1]
*  HFCR      : (R4)  thermal radiation leaf to roof          [J m-2 s-1]
*  RRAD_TOP  : (R4)  resistance for thermal radiation coming
*                       from above canopy                    [s m-1]
*  RRAD_BOT  : (R4)  resistance for thermal radiation coming
*                       from below canopy                    [s m-1]
*  TEMPAIR   : (R4)  temperature of air                      [oC]
*  CO2AIR    : (R4)  CO2 concentration                       [mumol mol-1]
*  VPDAIR    : (R4)  Vapour Pressure Deficit of air          [kPa]
*  RB        : (R4)  boundary layer resistance for vapour    [s m-1]
*  RCUT      : (R4)  cuticula resistance for vapour          [s m-1]
*  GMAXD     : (R4)  maximal leaf conductance at night       [m s-1]
*  GNVPD     : (R4)  parameter for leaf surface VPD response
*                    of GMAXD                                [kPa-1]
*
* Output:
*  TRANLEAF  : (R4)  leaf transpiration                   [mg H2O m-2 s-1]
*  TLEAF     : (R4)  leaf temperature                        [oC]
*  CONV      : (R4)  convective heat loss from leaf          [J m-2 s-1]
*  GLEAF     : (R4)  leaf conductance                        [m s-1]
*  GS        : (R4)  stomatal conductance                    [m s-1]
*  VPDsurf   : (R4)  VPD at leaf surface                     [kPa]
*  VPDla     : (R4)  leaf-air VPD                            [kPa]
*
*
* Subprograms called: LENER, STOMRESP
*
* Comment:
*      LAIC is dummy input variable
*****************************************************************
      SUBROUTINE LFTRAN(
     &             PARabs, NIRabs, HFPC, HFSC, HFCR,
     &             TEMPair,
     &             RRAD_TOP, RRAD_BOT, VPDair, CO2air,
     &             LAIC,
     &             GS, Gleaf,
     &             Tleaf, VPDsurf, VPDla, TRANleaf, CONVH
     & )
      IMPLICIT REAL(A-Z)

      INTEGER I, NITER

      COMMON /LEAFPAR/ Rcut, Rb
      COMMON /LEAFPAR2/ Gmaxd, GNVPD
```

```
      COMMON /LEAFPAR3/ Gmaxda, GNVPDa

      COMMON /CLIMHUM/ VPair

      LOGICAL LIGHT
      COMMON /GENCOM/ LIGHT

      LOGICAL DO_AIR
      LOGICAL DO_LFAIR
      COMMON /PARSTOM2/ DO_AIR, DO_LFAIR

      DATA NITER, VPD_EPS /10, 0.2/

* ======================================================================
      IF( .NOT. LIGHT ) THEN

         IF( DO_AIR ) THEN
*           Response of leaf conductance in the dark to air VPD
            Gleafd = Gmaxda * exp( - GNVPDa * VPDair )

*-----      Energy balance
            CALL LENER( PARabs, NIRabs, HFPC, HFSC, HFCR, TEMPair,
     &                     Gleafd, Rb, RRAD_TOP, RRAD_BOT, VPDair,
     &                     TRANleaf, TLeaf, CONVH )
            TRANleaf = AMAX1( 0.001, TRANleaf )

            VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )

*           Humidity at leaf surface
            VPDsurf = AMAX1( 0.01,
     &            (1./Gleafd) / (1./Gleafd + Rb ) * (VPleaf - VPair ) )

         ELSE
*           Response to leaf surface VPD
            DO I = 1, NITER

              VPDsurfi = VPDsurf

*           Response of leaf conductance in the dark to leaf surface VPD
              Gleafd = Gmaxd * exp( - GNVPD * VPDsurf )

*----         Energy balance
              CALL LENER( PARabs, NIRabs, HFPC, HFSC, HFCR, TEMPair,
     &                     Gleafd, Rb, RTHRAD, VPDair,
     &                     TRANleaf, TLeaf, CONVH )
              TRANleaf = AMAX1( 0.001, TRANleaf )

              VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )

*             Humidity at leaf surface
              VPDsurf = AMAX1( 0.01,
     &            (1./Gleafd) / (1./Gleafd + Rb ) * (VPleaf - VPair ) )

*             Exit loop when difference with previous value too small
              IF( ABS( VPDsurf - VPDsurfi ) .LT. VPD_eps ) THEN
                 GOTO 10
              ENDIF
            END DO
10          CONTINUE

       ENDIF
*      of DO_AIR
         Gleaf = Gleafd

      ELSE
* ======================================================================
*        LIGHT is TRUE

         Gcut = 1. / Rcut

         IF( DO_AIR ) THEN

*----       conductance in the dark, response to air-VPD
            Gleafd = Gmaxda * exp( - GNVPDa * VPDair )

*----       stomatal response
            CALL STOMRESP( PARabs, VPDair, CO2air, TEMPair,
```

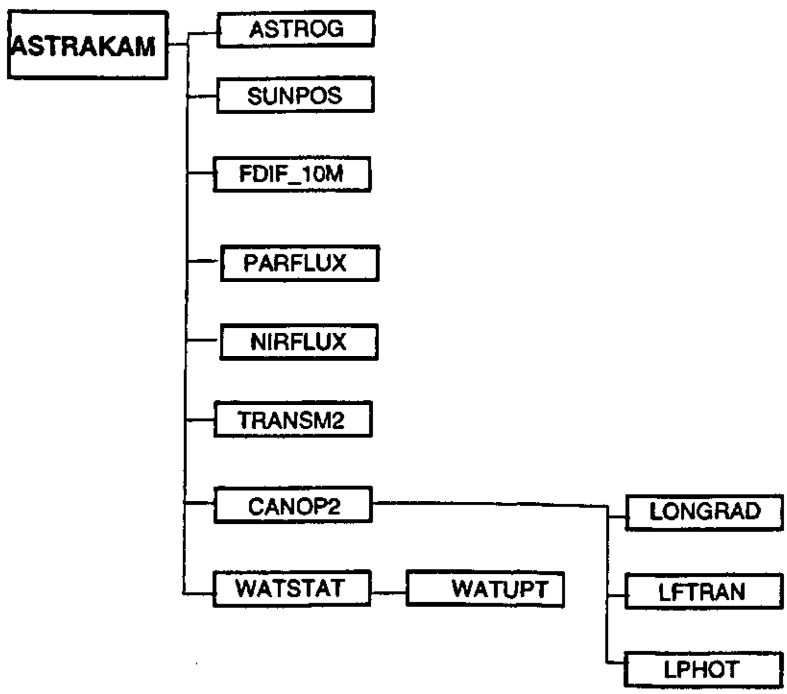
```
     &                      Gcut, Gs, Gleaf )

            Gleaf = AMAX1( Gleafd, Gleaf )

*----       Energy balance
            CALL LENER( PARabs, NIRabs, HFPC, HFSC, HFCR, TEMPair,
     &                      Gleaf, Rb, RRAD_TOP, RRAD_BOT, VPDair,
     &                      TRANleaf, TLeaf, CONVH )
            TRANleaf = AMAX1( 0.001, TRANleaf )

*         WRITE( 99, '(20X, 5F9.4)') PARABS, NIRABS, GS, GLEAF, TRANLEAF

            VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )

*----      Humidity at leaf surface
            VPDsurf = AMAX1( 0.01,
     &            (1./Gleaf) / (1./Gleaf + Rb ) * (VPleaf - VPair ) )

         ELSE

            VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )

*           leaf-air VPD
            VPDla = VPleaf - VPair

            DO I = 1, NITER

              IF( DO_LFAIR ) THEN
                 VPD = VPDla
              ELSE
                 VPD = VPDsurf
              ENDIF

              VPDi = VPD

*----         conductance in the dark, response to leaf surface VPD
              Gleafd = Gmaxd * exp( - GNVPD * VPDsurf )

*----         stomatal response
              CALL STOMRESP( PARabs, VPDair, CO2air, TEMPair,
     &                        Gcut, Gs, Gleaf )

              Gleaf = AMAX1( Gleafd, Gleaf )

*-----        Energy balance
              CALL LENER( PARabs, NIRabs, HFPC, HFSC, HFCR, TEMPair,
     &                      Gleaf, Rb, RRAD_TOP, RRAD_BOT, VPDair,
     &                      TRANleaf, TLeaf, CONVH )
              TRANleaf = AMAX1( 0.001, TRANleaf )

              VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )

*             Leaf-air VPD
              VPDla = VPleaf - VPair

*             Humidity at leaf surface
              VPDsurf = AMAX1( 0.01,
     &            (1./Gleaf) / (1./Gleaf + Rb ) * (VPleaf - VPair ) )

              IF( DO_LFAIR ) THEN
                 VPD = VPDla
              ELSE
                 VPD = VPDsurf
              ENDIF

*             Exit loop when difference with previous value too small
              IF( ABS( VPD - VPDi ) .LT. VPD_eps ) THEN
                 GOTO 40
              ENDIF

            END DO
40          CONTINUE
         ENDIF
*        of DO_AIR
      ENDIF
*     of LIGHT
* ==================================================================
```

```
      RETURN
      END
*****************************************************************
* SUBPROGRAM: MNRESP
*
* Input:
*  ITASK      : (I4)  control variable for initialization (ITASK=1),
*                     rate calculation (2), integration (3)
*                     terminal calculations (4) and resetting (5)     [-]
*  DELT       : (R4)  time step                                      [h]
*  WLV        : (R4)  dry weight of leaves                     [g DM m-2]
*  WST        : (R4)  dry weight of stems                      [g DM m-2]
*  WRT        : (R4)  dry weight of roots                      [g DM m-2]
*  WSO        : (R4)  dry weight of storage organs             [g DM m-2]
*  MAINLV     : (R4)  maint. costs leaves at 25 oC       [g CH2O g dm-1 d-1]
*  MAINST     : (R4)  maint. costs stems at 25 oC        [g CH2O g dm-1 d-1]
*  MAINRT     : (R4)  maint. costs roots at 25 oC        [g CH2O g dm-1 d-1]
*  MAINSO     : (R4)  maint. costs storage org. at 25 oC [g CH2O g dm-1 d-1]
*  Q10MN      : (R4)  Q10 maintenance respiration             [-]
*  REFTMP     : (R4)  reference temperature maint. resp.      [oC]
*  TEMPAIR    : (R4)  temperature greenhouse air              [oC]
*
* Output:
*  DMAINT     : (R4)  daily total of maint. costs        [g CH2O m-2 d-1]
*****************************************************************
      SUBROUTINE MNRESP( ITASK, DELT, Q10MN, REFTMP,
     &             WLV, WST, WRT, WSO,
     &        MAINLV, MAINST, MAINRT, MAINSO,
     &        TEMPAIR, DMAINT )
      IMPLICIT REAL(A-Z)

      INTEGER ITASK

*     Initialization or resetting
      IF( ITASK .EQ. 1 .OR. ITASK .EQ. 5) THEN

         DMAINT = 0.

      ELSEIF( ITASK .EQ. 2 ) THEN

*----    Maintenance respiration [mg CH2O m-2 s-1]
*        86.4 converts g d-1 to mg s-1
         MAINTS = ( WLV*MAINLV + WST*MAINST + WSO*MAINSO + WRT*MAINRT )
     &            / 86.4
         TEFF  = Q10MN**( ((TEMPAIR-REFTMP)/10.) )
         MAINT = MAINTS * TEFF

      ELSEIF (ITASK .EQ. 3) THEN

*        3.6 converts mg s-1 to g h-1
         DMAINT = DMAINT + MAINT * DELT * 3.6

      ENDIF

      RETURN
      END
*****************************************************************
* SUBPROGRAM: NIRFLUX
* Date: 14-04-1994
*
* Purpose: calculation of intensity of NIR flux and UV flux
*  and the diffuse and direct components of these fluxes for
*  10 minute intervals
*
* Input:
*  ATMTR        : (R4) atmospheric transmission                  [-]
*  GLOBRADDIF : (R4) diffuse global radiation              [J m-2 s-1]
*  GLOBRADDIR : (R4) direct global radiation               [J m-2 s-1]
*  PARDIF       : (R4) diffuse PAR                         [J m-2 s-1]
*  PARDIR       : (R4) direct PAR                          [J m-2 s-1]
*
* Output:
*  NIRDIF       : (R4) diffuse NIR                         [J m-2 s-1]
*  NIRDIR       : (R4) direct NIR                          [J m-2 s-1]
*  UVDIF        : (R4) diffuse UV                          [J m-2 s-1]
*  UVDIR        : (R4) direct UV                           [J m-2 s-1]
```

```
*
* Comment:
*  Note that all fluxes are outside greenhouse
**********************************************************************
      SUBROUTINE NIRFLUX( ATMTR, GLOBRADDIF, GLOBRADDIR,
     &             PARDIF, PARDIR, NIRDIF, NIRDIR, UVDIF, UVDIR )
      IMPLICIT REAL(A-Z)

*     Ratio of UV to global radiation
      UVdivGLOB = 0.05

*     Apparent fraction clear
      IF( ATMTR .GT. 0.8 ) THEN
         FCLEAR = 1.
      ELSEIF( ATMTR .LT. 0.3 ) THEN
         FCLEAR = 0.
      ELSE
         FCLEAR = (ATMTR - 0.3) / (0.8-0.3)
      ENDIF

*     Ratio of diffuse UV to diffuse global radiation
      UVDIFdivGLOBDIF = 0.05 + FCLEAR * 0.07

      UV = UVdivGLOB * ( GLOBRADDIF + GLOBRADDIR )
      UVDIF = AMIN1( UV, UVDIFdivGLOBDIF * GLOBRADDIF )
      UVDIR = AMAX1( 0., UV - UVDIF )

*     Diffuse and direct NIR are found by subtracting diffuse and
*     direct PAR energy and UV fluxes from diffuse and direct global
*     radiation
      NIRDIF = GLOBRADDIF - PARDIF - UVDIF
      NIRDIR = GLOBRADDIR - PARDIR - UVDIR

      RETURN
      END
**********************************************************************
* SUBPROGRAM: PARFLUX
* Date: 11-04-1994
* Purpose: calculation of the intensity of the PAR energy flux
*  and the fraction
*  diffuse in PAR from global radiation and atmospheric transmission
*
* Input:
*  ATMTR    : (R4)  atmospheric transmission                 [-]
*  GLRADO   : (R4)  global radiation outside greenhouse      [J m-2 s-1]
*  ELEVN    : (R4) elevation of sun                          [radians]
*  FDIFGLOB : (R4) fraction diffuse in global radiation      [-]
*
* Output:
*  PAROUT   : (R4) PAR outside the greenhouse                [J m-2 s-1]
*  FDIFPAR  : (R4) fraction diffuse in PAR                   [-]
**********************************************************************
      SUBROUTINE PARFLUX( ATMTR, GLRADO, ELEVN, FDIFGLOB,
     &                      PAROUT, FDIFPAR )
      IMPLICIT REAL(A-Z)

      A = 2.9
      B = 4.9
      C = 0.51
      E = 0.84
      F = 0.033

*     Fraction PAR energy
      fm = e * EXP( f / SIN( ELEVN ) )

*     Ratio between PAR photon flux and global radiation
      RATIO =
     &    a - fm * ( 1.0 - EXP(- b * ATMTR ** c ) )

      PAROUT = RATIO * GLRADO / 4.57

*     Apparent fraction clear
      IF( ATMTR .GT. 0.8 ) THEN
         FCLEAR = 1.
      ELSEIF( ATMTR .LT. 0.3 ) THEN
         FCLEAR = 0.
      ELSE
```

```
           FCLEAR = (ATMTR - 0.3) / 0.5
        ENDIF

*       Fraction diffuse in PAR
        FDIFPAR = AMIN1( 1., FDIFGLOB * ( 1. + FCLEAR * 0.35 ) )

        RETURN
        END
**************************************************************************
* SUBPROGRAM: STOMRESP
* Purpose: calculation of stomatal conductivity
* Description:
*  Negative exponential response to
*       - absorbed PAR
*       - VPD (air, leaf-air, or leaf surface )
*       - CO2 concentration
*       (after Nederhoff et al., 1992 )
*   Optimum response to temperature
*       (after Stanghellini, 1987)
* Input:
*   PARABS    : (R4)  absorbed PAR energy flux                  [J m-2 s-1]
*   VPD       : (R4)  Vapour Pressure Deficit                   [kPa]
*   CO2AIR    : (R4)  CO2 concentration                         [mumol mol-1]
*   TEMP      : (R4)  leaf temperature                          [oC]
*   GCUT      : (R4)  cuticular conductance to H2O              [m s-1]
* Output:
*  GLEAF      : (R4)  leaf conductance                           [m s-1]
*  GS         : (R4)  stomatal conductance                       [m s-1]
**************************************************************************
        SUBROUTINE STOMRESP( PARabs, VPD, CO2air, TEMP,
     &    Gcut, Gs, Gleaf )
        IMPLICIT REAL(A-Z)

        COMMON /PARSTOM/ Gsmax, CD1, CD2, CD3, CD4, CD5, CD6

        Gs = AMAX1( 0.0001,
     &              Gsmax
     &              * ( 1. - CD1 * EXP( -CD2 * PARabs ) )
     &              * EXP( - CD3 * VPD )
     &              * EXP( - CD4 * CO2air )
     &              / ( 1. + CD5 * (TEMP-CD6)**2 )
     &       )

        Gleaf = Gcut + Gs

        RETURN
        END
```

Appendix IV:
Listing of model ASTRAKAM and additional routines



Routine	Calculation of
ASTROG	astronomical variables
SUNPOS	sun position
FDIF_10M	fraction diffuse in global radiation
PARFLUX	fluxes total, diffuse and direct PAR
NIRFLUX	fluxes total, diffuse and direct NIR
RANSM2	transmissivity greenhouse cover
CANOP2	canopy transpiration and energy balance, similar to CANOPF (see INTKAM)
LONGRAD	thermal radiation
LFTRAN	leaf transpiration, energy balance and stom. conductance
LENER	leaf energy balance
STOMRESP	stomatal response
LPHOT	leaf gross photosynthesis (summary leaf phot. model)
WATSTAT	crop water content
WATUPT	water uptake of crop

```
***************************************************************************
* Program: ASTRAKAM
* Author: H. Gijzen, AB-DLO, Wageningen
* Version: 1.0
* Date: May 1994
* Purpose: Calculation of crop transpiration, crop energy balance
*      and crop gross photosynthesis
* Description:
*  The canopy is taken to be multi-layered. Of each leaf layer the energy
*  balance is calculated. From the energy balance leaf transpiration and
*  leaf temperature are calculated.
*  Leaf conductances are calculated
*  independent from rate of leaf photosynthesis. Leaf photosynthesis is
*  calculated with the summary model of leaf photosynthesis

* Subroutines called:
*    (simulation)
*      ASTROG     - astronomical variables
*      CANOP2     - canopy transpiration and energy balance
*      FDIF_10M   - fraction diffuse in global radiation
*      NIRFLUX    - diffuse and direct NIR and UV outside greenhouse
*      PARFLUX    - diffuse and direct PAR
*      SUNPOS     - sun position
*      TRANSM2    - transmissivity greenhouse cover
*      WATSTAT    - water content of crop
*
*    (general)
*      ENVASTRA   - obtaining data and parameters from files and user
*      HTIMER     - timer variables
*
* Input:
*      data file            (unit IUDAT)
*      data info file       (unit IUDATIF)
*      timer file           (unit IUTIM)
*      parameter file       (unit IUPAR)
*      transmissivity file  (unit IUTRAN)
*
* Output:
*      file with instantaneous values of parameters     (unit IUOUT)
*      file with energy fluxes                          (unit IUOUTE)
*      file with daily totals                           (unit IUOUTS)
*
*      Names of output files are derived from data file:
*        E.g. data file 'K1151A.DAT'  (name maximal 6 alphanum. characters)
*       -> file name instant. values:  'K1151A' + runstring +  '.CSV'
*       -> file name energy fluxes:    'K1151A' + runstring + 'E.CSV'
*       -> file name cumulative values:  'K1151A' + runstring +  '.SUM'
*       where 'runstring' is an alphanumeric character
*
* Comments:
*   Simulation is done for a single day.
*   Time control:
*       - program increments time counter (DAYMIN = DAYMIN + DELTMIN)
*         time steps in IDELTMIN minutes
*       - start time is minimum of start time of data file and STARTTIM
*         in timer file
*       - finish of simulation when end-of-file is encountered or
*         when finish time is reached
***************************************************************************
      PROGRAM ASTRAKAM
      IMPLICIT REAL(A-Z)

* Logicals for simulation control
      LOGICAL LIGHT
      COMMON /GENCOM/ LIGHT

      CHARACTER*5 EXPRNT

*     File I/O
      INTEGER IUOUT, IUOUTE, IUOUTS
      COMMON /IO_UNIT_OUT/ IUOUT, IUOUTE, IUOUTS
      CHARACTER*40 FILOUT, FILOUTE, FILOUTS
      COMMON /IO_NAME_OUT/ FILOUT, FILOUTE, FILOUTS
```

```
      INTEGER IUDAT, IUDATIF, IUTIM, IUPAR, IUTRAN
      CHARACTER*40 FILTRN
      CHARACTER*40 DATAFIL, INFOFIL, PARFIL
      CHARACTER*40 TIMFIL
      COMMON /IO_UNIT_IN/ IUDAT, IUDATIF, IUTIM, IUPAR, IUTRAN
      COMMON /IO_NAME_IN/ DATAFIL, INFOFIL, TIMFIL, PARFIL, FILTRN

      INTEGER ITASK
      INTEGER IOPHASE

      INTEGER I, I3, IVAL, NVALS
      INTEGER IDUM

      INTEGER IUNERR
      LOGICAL INI, RESET
      LOGICAL INI_CANOP

      INTEGER ICOM

      COMMON /GENCOM2/ SOLHR

      COMMON /ENERGY_EXCH1/ TCANOP
      COMMON /ENERGY_EXCH2/ HF_CR, HF_SC, HF_PC, CONVH
      COMMON /ENERGY_EXCH3/ HFCRTOT, HFSCTOT, HFPCTOT

* Timer variables
      INTEGER SIM_DAY_MIN, DATA_DAY_MIN
      INTEGER SIM_DAY_MIN_START
      INTEGER DELTMIN, NDELT, OUTDELMIN

* General simulation control
      LOGICAL INI_SIM, TERMNL
      LOGICAL OUTPUT
      LOGICAL FILE_END

      LOGICAL command
      COMMON /IO_0/ command

* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
* Parameters
      COMMON /LEAFPAR/ Rcut, Rb
      COMMON /LEAFPAR_mol/ Rcut_mol, Rb_mol
      COMMON /LEAFPAR2/ Gmaxd, GNVPD
      COMMON /LEAFPAR3/ Gmaxda, GNVPDa

      DATA IUDAT, IUDATIF, IUTIM, IUPAR, IUTRAN / 11, 12, 13, 14, 15 /
      DATA IUOUT, IUOUTE, IUOUTS / 20, 21, 22 /
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

      PI = 3.1415926
      RADN = PI/180.

*---- Initialization: reading of parameter values and opening data file
      ITASK = 1

      IOPHASE = 1
      CALL ENVASTRA( IOPHASE, EXPRNT,
     &  DAYNR, SOLHR, HOUR, DATA_DAY_MIN, SIM_DAY_MIN, SINELV,
     &  STARTTIM, FINTIM, DELTMIN, OUTDELMIN,
     &  LAT, TIMCOR, SUNRISE, SUNSET, DAYL, REFGR,
     &  AZIMGR, TRDIF, TRCOR_UV,
     &  LAI, KDIF, KDIFBL, SCP, SCN, GB, GCUT,
     &  GLRADO, GLOBDIFO,
     &  CO2AIR, TEMPAIR, VPDAIR,
     &  TRAN_MEAS,
     &  SSPT, SSPB, TPIPE, TROOF, TFLOOR,
     &  TEMPAIR_OUT, SCREEN,
     &  WATCONI, WATCONMAX, RWATCONI, PSIWIL, PSIROOTM,
     &  RESWAT, RIONUPT,
     &  FILE_END )

*     Reading transmissivity properties greenhouse
      CALL TRANSM2( ITASK, IUTRAN, FILTRN,
     &             AZIMGR, AZIMS, ELEVN,
     &             TRDIF, TRCOR_UV, TRCON, TRGLAS )

*     Initialization of water status
```

```
      CALL WATSTAT( ITASK,
     &    WATCONI, WATCONMAX, RWATCONWI, PSIWIL, PSIROOTM, RESWAT,
     &    RIONUPT,
     &    DELT, HOUR, TEMPAIR, TRANSP, RWUPT, WATCON, PSIPL )

*---- Get day number and start time from data file
      IOPHASE = 2
      CALL ENVASTRA( IOPHASE, EXPRNT,
     & DAYNR, SOLHR, HOUR, DATA_DAY_MIN, SIM_DAY_MIN, SINELV,
     & STARTTIM, FINTIM, DELTMIN, OUTDELMIN,
     & LAT, TIMCOR, SUNRISE, SUNSET, DAYL, REFGR,
     & AZIMGR, TRDIF, TRCOR_UV,
     & LAI, KDIF, KDIFBL, SCP, SCN, GB, GCUT,
     & GLRADO, GLOBDIFO,
     & CO2AIR, TEMPAIR, VPDAIR,
     & TRAN_MEAS,
     & SSPT, SSPB, TPIPE, TROOF, TFLOOR,
     & TEMPAIR_OUT, SCREEN,
     & WATCONI, WATCONMAX, RWATCONI, PSIWIL, PSIROOTM,
     & RESWAT, RIONUPT,
     & FILE_END )

*---- user interaction, opening output files, writing info in headers
      IOPHASE = 3
      CALL ENVASTRA( IOPHASE, EXPRNT,
     & DAYNR, SOLHR, HOUR, DATA_DAY_MIN, SIM_DAY_MIN, SINELV,
     & STARTTIM, FINTIM, DELTMIN, OUTDELMIN,
     & LAT, TIMCOR, SUNRISE, SUNSET, DAYL, REFGR,
     & AZIMGR, TRDIF, TRCOR_UV,
     & LAI, KDIF, KDIFBL, SCP, SCN, GB, GCUT,
     & GLRADO, GLOBDIFO,
     & CO2AIR, TEMPAIR, VPDAIR,
     & TRAN_MEAS,
     & SSPT, SSPB, TPIPE, TROOF, TFLOOR,
     & TEMPAIR_OUT, SCREEN,
     & WATCONI, WATCONMAX, RWATCONI, PSIWIL, PSIROOTM,
     & RESWAT, RIONUPT,
     & FILE_END )

*     Variable column names
      WRITE(IUOUT,'(A,A,A,A)')
     &    ' COL: Hour, GLRADO, TRDIR, GLOBRAD, NETRAD, RADABS, PAR, ',
     &    ' CO2air, TEMPAIR, ',
     &    ' VPDair, ',
     &    ' TRAN_MEAS, TRAN_SIM, Tpipe, Troof, Tground, GLtot,',
     &    ' RWUPT, WATCON '

      WRITE(IUOUTE,'(A,A,A,A,A,A)')
     & ' HOUR, TEMPAIR, CROPTEMP, ',
     & ' GLOBRAD, GLRADABS, TRAN_ENER,  HF_PC, HF_CR, HF_SC,',
     & ' CONVH '

*---- Initial calculations
*     Conversion of degrees into radians
      LAT = LAT * RADN

      RB = 1. / GB
      RCUT = 1. / GCUT

*     Resistance for thermal radiation
      RTHRAD = 200. / KDIFBL

      Rb_mol = Rb / 40.
      Rcut_mol = Rcut / 40.

* Initial value for stomatal conductance
      gsin = 2. * GMAXDA

      SQP = SQRT( 1.-SCP )
      AZIMGR = AZIMGR * RADN

*     Daily calculations
      CALL ASTROG( DAYNR, LAT, SOLARC, SINLD, COSLD, DECL,
     &             DAYL, DSINBE )

      SUNRISE = 12. - 0.5 * DAYL
      SUNSET = 12. + 0.5 * DAYL
```

```
*---- Timer
      INI_SIM = .TRUE.
      CALL HTIMER( INI_SIM, DAYMIN, DATA_DAYMIN,
     &  STARTTIM, FINTIM, DELTMIN, DELT,
     &  OUTDELMIN, DAYMIN_START, HOUR, OUTPUT, TERMNL )

*     From standard time to solar time
      SOLHR = HOUR + TIMCOR

      TSIMTRANS_D = 0.
      TSIMTRANS_N = 0.
      TMEASTRANS_D = 0.
      TMEASTRANS_N = 0.
      TGLRADO = 0.
      TGLOBDIRO = 0.
      TGLOBDIFO = 0.
      TGLOBRADIN = 0.
      TGLOBDIR = 0.
      TGLOBDIF = 0.
      TPAR = 0.
      TRADABS = 0.
      TPARABS = 0.
      TPARDIRTO = 0.
      TNIRABS = 0.
      TNETRAD_D = 0.
      TNETRAD_N = 0.
      THFCRTOT_D = 0.
      THFCRTOT_N = 0.
      THFSCTOT_D = 0.
      THFSCTOT_N = 0.
      THFPCTOT_D = 0.
      THFPCTOT_N = 0.

      DO WHILE( .NOT. TERMNL )

* ==============================================================
*         Integration
* ==============================================================
         ITASK = 3
*     Calculation of daily total
         IF( LIGHT ) THEN
            TSIMTRANS_D = TSIMTRANS_D + TRAN_SIM * DELT
            TMEASTRANS_D = TMEASTRANS_D + TRAN_MEAS * DELT
            TNETRAD_D = TNETRAD_D + NETRAD * DELT
            THFCRTOT_D = THFCRTOT_D + HFCRTOT * DELT
            THFSCTOT_D = THFSCTOT_D + HFSCTOT * DELT
            THFPCTOT_D = THFPCTOT_D + HFPCTOT * DELT
            TGLRADO = TGLRADO + GLRADO * DELT
            TGLOBDIRO = TGLOBDIRO + GLOBDIRO * DELT
            TGLOBDIFO = TGLOBDIFO + GLOBDIFO * DELT
            TGLOBRADIN = TGLOBRADIN + GLOBRADIN * DELT
            TGLOBDIR = TGLOBDIR + GLOBDIR * DELT
            TGLOBDIF = TGLOBDIF + GLOBDIF * DELT
            TPAR = TPAR + PAR * DELT
            TRADABS = TRADABS + RADABS * DELT
            TPARABS = TPARABS + PARABS * DELT
            TPARDIRTO = TPARDIRTO + PARDIRTO * DELT
            TNIRABS = TNIRABS + NIRABS * DELT
         ELSE
            TSIMTRANS_N = TSIMTRANS_N + TRAN_SIM * DELT
            TMEASTRANS_N = TMEASTRANS_N + TRAN_MEAS * DELT
            TNETRAD_N = TNETRAD_N + NETRAD * DELT
            THFCRTOT_N = THFCRTOT_N + HFCRTOT * DELT
            THFSCTOT_N = THFSCTOT_N + HFSCTOT * DELT
            THFPCTOT_N = THFPCTOT_N + HFPCTOT * DELT
         ENDIF

         CALL WATSTAT( ITASK,
     &     WATCONI, WATCONMAX, RWATCONWI, PSIWIL, PSIROOTM, RESWAT,
     &     RIONUPT,
     &     DELT, HOUR, TEMPAIR, TRANSP, RWUPT, WATCON, PSIPL )
* ==============================================================
*         Calculations driving variables
* ==============================================================
         ITASK = 2
```

```
*         Solar position; SINELV is sine of solar elevation,
*         AZIMS is azimuth of sun
          CALL SUNPOS (LAT, SINLD, COSLD, DECL, SOLHR,
     &                    ELEVN, AZIMS, SINELV )

          IOPHASE = 4
          CALL ENVASTRA( IOPHASE, EXPRNT,
     &      DAYNR, SOLHR, HOUR, DATA_DAY_MIN, SIM_DAY_MIN, SINELV,
     &      STARTTIM, FINTIM, DELTMIN, OUTDELMIN,
     &      LAT, TIMCOR, SUNRISE, SUNSET, DAYL, REFGR,
     &      AZIMGR, TRDIF, TRCOR_UV,
     &      LAI, KDIF, KDIFBL, SCP, SCN, GB, GCUT,
     &      GLRADO, GLOBDIFO,
     &      CO2AIR, TEMPAIR, VPDAIR,
     &      TRAN_MEAS,
     &      SSPT, SSPB, TPIPE, TROOF, TFLOOR,
     &      TEMPAIR_OUT, SCREEN,
     &      WATCONI, WATCONMAX, RWATCONI, PSIWIL, PSIROOTM,
     &      RESWAT, RIONUPT,
     &      FILE_END )

*         End of simulations
          IF( FILE_END ) GOTO 99

*         Vapour pressure and saturated vapour pressure of
*         greenhouse air [kPa]
          VPSATair = .6107 * EXP( 17.4 * TEMPair / (TEMPair + 239.) )
          VPair = VPSATair - VPDair

          IF (GLRADO .LT. 0.1 ) THEN
             LIGHT = .FALSE.
             PARDIF = 0.
             PARDIR = 0.
             NIRDIF = 0.
             NIRDIR = 0.
             GLRADO = 0.
             UVDIR = 0.
             UVDIF = 0.
          ELSE
             LIGHT = .TRUE.

             SINELV = AMAX1( .05, SINELV )
             ELEVN = AMAX1( .05, ELEVN )

*            Atmospheric transmission
             ATMTR = GLRADO / (SOLARC * SINELV )

*            Direct and diffuse radiation outside greenhouse
             FRDIF = FDIF_10M( SOLARC, GLRADO, SINELV )
             GLOBDIFO = FRDIF * GLRADO
             GLOBDIRO = GLRADO - GLOBDIFO

*            direct and diffuse PAR outside greenhouse
             CALL PARFLUX( ATMTR, GLRADO, ELEVN,
     &                          FRDIF, PAROUT, FRDIFPAR )

*            direct and diffuse NIR and UV outside greenhouse
             CALL NIRFLUX( ATMTR, GLOBDIFO, GLOBDIRO,
     &                    PARDIFO, PARDIRO, NIRDIFO, NIRDIRO,
     &                    UVDIFO, UVDIRO )

*           Transmission greenhouse
            CALL TRANSM2( ITASK, IUTRAN, FILTRN,
     &             AZIMGR, AZIMS, ELEVN,
     &             TRDIF, TRCOR_UV, TRCON, TRGLAS )

*           Diffuse and direct PAR and NIR inside greenhouse
            PARDIF = PARDIFO * TRDIF
            UVDIF = UVDIFO * TRDIF * TRCOR_UV

            TRDIR = TRCON * TRGLAS
            PARDIR = PARDIRO * TRDIR
            UVDIR = UVDIRO * TRDIR * TRCOR_UV

            NIRDIR = NIRDIRO * TRDIR
            NIRDIF = NIRDIFO * TRDIF
```

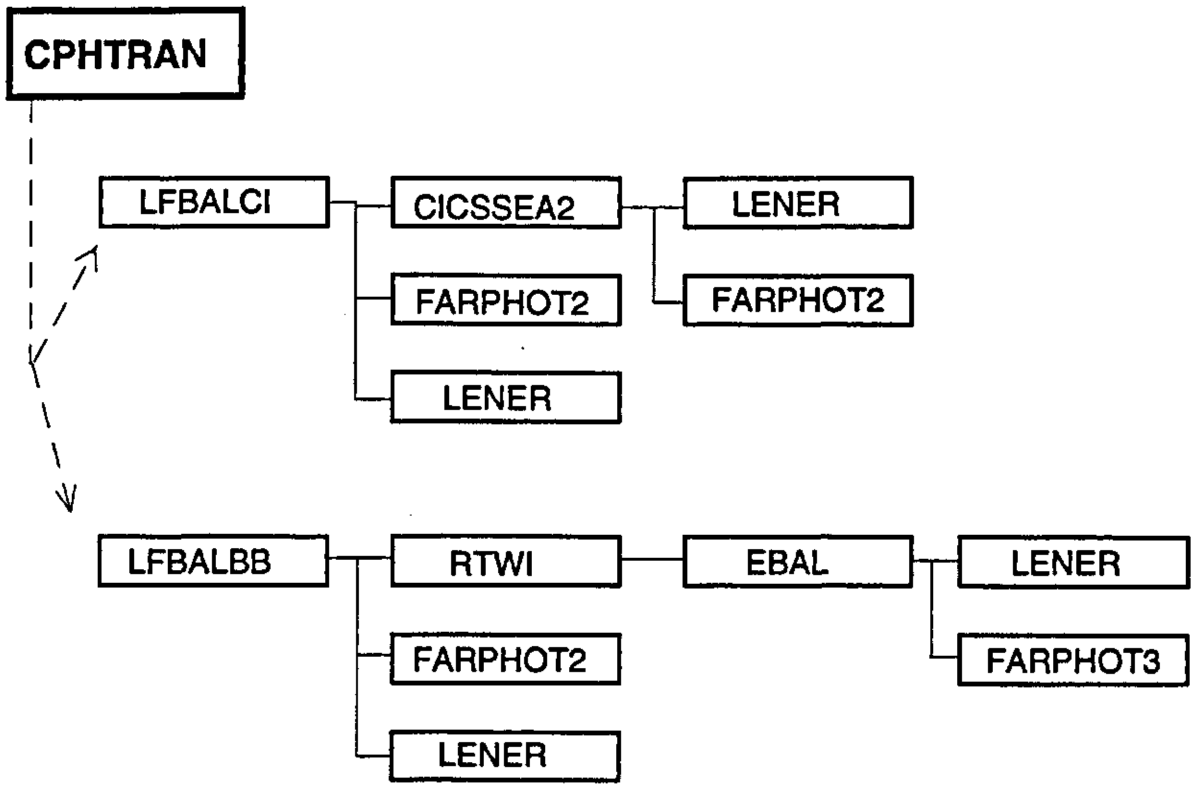
```
         ENDIF
*        of GLRADO .LT. 0

         NIR = NIRDIR + NIRDIF

         PAR = PARDIF + PARDIR
         GLOBDIR = PARDIR + NIRDIR + UVDIR
         UV = UVDIF + UVDIR

         GLOBDIF = PARDIF + NIRDIF + UVDIF
         GLOBRADIN = PAR + NIR + UV

* ==============================================================
*         Rate calculations
* ==============================================================

          CALL CANOP2( INI_SIM, HOUR, PARDIF, PARDIR, NIRDIF,
     &      NIRDIR, UVDIF, UVDIR, ELEVN, SINELV,
     &      LAI, ANDIS, KDIFBL, SCP, SCN, CO2air, TEMPAIR, VPDair,
     &      SSPT, SSPB, TPIPE, TROOF, TGROUND,
     &      GSin, RTHRAD, REFGR,
     &      PARABS, PARDIRTO, NIRABS, UVABS, NETRAD,
     &      GSTOT, GLTOT,
     &      FGROS, TRAN_SIM
     &      )

         RADABS = PARABS + NIRABS + UVABS

*        Energy flux associated with transpiration
         TRAN_ENER_SIM = TRAN_SIM * 2.5
         TRAN_ENER_MEAS = TRAN_MEAS * 2.5

*----    Water status
         CALL WATSTAT( ITASK,
     &     WATCONI, WATCONMAX, RWATCONWI, PSIWIL, PSIROOTM, RESWAT,
     &     RIONUPT,
     &     DELT, HOUR, TEMPAIR, TRANSP, RWUPT, WATCON, PSIPL )

         IF ( OUTPUT ) THEN

         WRITE( IUOUT, 901) HOUR, GLRADO, GLOBRADIN, NETRAD,
     &                      RADABS, PAR,
     &                      CO2air, TEMPair, VPDair,
     &                      TRAN_MEAS, TRAN_SIM,
     &                      Tpipe, Troof, Tground, GLtot,
     &                      RWUPT, WATCON
901      FORMAT( F8.3, ',',F5.0, ',', 4(F5.0,','),
     &           F6.0,',', F6.2, ',', F6.3, ',',
     &           2(F6.1, ','), 3(F5.1,','), F8.3,
     &           F6.1, ',' F6.1 )

           WRITE( IUOUTE, 905 )
     &      HOUR, TEMPAIR, TCANOP,
     &     GLOBRADIN, RADABS, TRAN_ENER_SIM, -HFPCTOT, HFCRTOT,
     &     -HFSCTOT, CONVH
905        FORMAT( F7.3, ',', 2(F5.1, ','),
     &     8( F6.0, ',') )

         ENDIF
*         ENDIF OUTPUT

         WRITE(*, '(A,F7.3)') '+ ', HOUR

*---     Time update
         CALL HTIMER( INI, DAYMIN, DATA_DAYMIN,
     &     STARTTIM, FINTIM, DELTMIN, DELT,
     &     OUTDELMIN, DAYMIN_START, HOUR, OUTPUT, TERMNL )
*        From standard time to solar time
         SOLHR = HOUR + TIMCOR

      END DO
*        of end while

99    CONTINUE

* ==============================================================
*         Terminal
```

```
* ==================================================================
      WRITE (*, '(A,F9.1,A)')
     &     ' Total sim. daytime transpiration  ', TSIMTRANS_D,
     &     ' g H2O m-2 '
      WRITE (*, '(A,F9.1,A)')
     &     ' Total meas. daytime transpiration  ', TMEASTRANS_D,
     &     ' g H2O m-2 '

      TMEASTRANS = TMEASTRANS_D + TMEASTRANS_N
      TSIMTRANS = TSIMTRANS_D + TSIMTRANS_N

      TTRAN_ENER_MEAS_D = TMEASTRANS_D * 2.5 * 1000. * 1.E-6
      TTRAN_ENER_MEAS_N = TMEASTRANS_N * 2.5 * 1000. * 1.E-6
      TTRAN_ENER_MEAS = TMEASTRANS * 2.5 * 1000. * 1.E-6

      TTRAN_ENER_SIM = TSIMTRANS * 2.5 * 1000. * 1.E-6
      TTRAN_ENER_SIM_D = TSIMTRANS_D * 2.5 * 1000. * 1.E-6
      TTRAN_ENER_SIM_N = TSIMTRANS_N * 2.5 * 1000. * 1.E-6

      WRITE( IUOUTS, '(A,A)')
     & '                    ',
     & ' Day         Night         Total '
      WRITE( IUOUTS, 915) ' Transp. measured ',
     &     TMEASTRANS_D, TMEASTRANS_N, TMEASTRANS,
     &     ' g H2O m-2 '
      WRITE( IUOUTS, 915) ' Transp. simulated ',
     &     TSIMTRANS_D, TSIMTRANS_N, TSIMTRANS,
     &     ' g H2O m-2 '

      WRITE( IUOUTS, 915) ' Transp. energy measured ',
     &     TTRAN_ENER_MEAS_D, TTRAN_ENER_MEAS_N, TTRAN_ENER_MEAS,
     &     ' MJ m-2 '

      WRITE( IUOUTS, 915) ' Transp. energy simulated ',
     &     TTRAN_ENER_SIM_D, TTRAN_ENER_SIM_N, TTRAN_ENER_SIM,
     &     ' MJ m-2 '

      CF = 3600. * 1.E-6

      TNETRAD = TNETRAD_D + TNETRAD_N
      THFCRTOT = THFCRTOT_D + THFCRTOT_N
      THFSCTOT = THFSCTOT_D + THFSCTOT_N
      THFPCTOT = THFPCTOT_D + THFPCTOT_N
      WRITE( IUOUTS, 915 ) '  Net radiation   ',
     &            TNETRAD_D * CF, TNETRAD_N * CF, TNETRAD * CF,
     &            ' MJ m-2'
      WRITE( IUOUTS, 915 ) ' Heat flux canopy - roof  ',
     &            THFCRTOT_D * CF, THFCRTOT_N * CF, THFCRTOT * CF,
     &            ' MJ m-2'
      WRITE( IUOUTS, 915 ) ' Heat flux pipe - canopy ',
     &            THFPCTOT_D * CF, THFPCTOT_N * CF, THFPCTOT * CF,
     &            ' MJ m-2'
      WRITE( IUOUTS, 915 ) ' Heat flux soil - canopy ',
     &            THFSCTOT_D * CF, THFSCTOT_N * CF, THFSCTOT * CF,
     &            ' MJ m-2'

      WRITE( IUOUTS, * )

* DELT in hours
      WRITE( IUOUTS, 913 ) ' TGLRADO     ', TGLRADO * CF, ' MJ m-2'
      WRITE( IUOUTS, 913 ) ' TGLOBDIRO   ', TGLOBDIRO * CF, ' MJ m-2'
      WRITE( IUOUTS, 913 ) ' TGLOBDIFO   ', TGLOBDIFO * CF, ' MJ m-2'
      WRITE( IUOUTS, 913 ) ' TGLOBRADIN ', TGLOBRADIN * CF, ' MJ m-2'
      WRITE( IUOUTS, 913 ) ' TGLOBDIR    ', TGLOBDIR * CF, ' MJ m-2'
      WRITE( IUOUTS, 913 ) ' TGLOBDIF    ', TGLOBDIF * CF, ' MJ m-2'
      WRITE( IUOUTS, 913 ) ' TPAR        ', TPAR * CF, ' MJ m-2'
      WRITE( IUOUTS, 913 ) ' TRADABS     ', TRADABS * CF, ' MJ m-2'
      WRITE( IUOUTS, 913 ) ' TPARABS     ', TPARABS * CF, ' MJ m-2'
      WRITE( IUOUTS, 913 ) ' TPARDIRTO   ', TPARDIRTO * CF, ' MJ m-2'
      WRITE( IUOUTS, 913 ) ' TNIRABS     ', TNIRABS * CF, ' MJ m-2'

913   FORMAT( A20, F9.3, A )
914   FORMAT( A20, F10.3, A )
915   FORMAT( A20, 3F10.3, A )
```

```
      WRITE( *, '(A,A)' ) ' Output to : ', FILOUT
      WRITE( *, '(A,A)' ) ' Output to : ', FILOUTE
      WRITE( *, '(A,A)' ) ' Output to : ', FILOUTS

      END
*************************************************************************
* SUBPROGRAM: HTIMER
* Purpose: incrementing time counter; counter is cumulative number of
*   minutes in the current day from 0.0 hour onwards
*************************************************************************
      SUBROUTINE HTIMER( INI, DAYMIN, DATA_DAYMIN,
     &  STARTTIM, FINTIM, DELTMIN, DELT,
     &  OUTDELMIN, DAYMIN_START, HOUR, OUTPUT, TERMNL )
      IMPLICIT REAL(A-Z)

      LOGICAL INI
      LOGICAL OUTPUT, TERMNL

      INTEGER DAYMIN, DATA_DAYMIN
      INTEGER M_START, DAYMIN_START
      INTEGER DELTMIN, OUTDELMIN

      IF( INI ) THEN
         M_START = MAX( INT(60.* STARTTIM), DATA_DAYMIN )

         IF( MOD( M_START, DELTMIN ) .EQ. 0 ) THEN
            DAYMIN_START = M_START
         ELSE
            DAYMIN_START = ( M_START / DELTMIN + 1 ) * DELTMIN
         ENDIF

*        Time step in hours
         DELT = FLOAT( DELTMIN ) / 60.

         DAYMIN = DAYMIN_START
         OUTPUT = .TRUE.

         RETURN
      ENDIF

      DAYMIN = DAYMIN + DELTMIN

      IF ( MOD( DAYMIN, OUTDELMIN ) .EQ. 0 ) THEN
         OUTPUT = .TRUE.
      ELSE
         OUTPUT = .FALSE.
      ENDIF

      IF( DAYMIN .GT. INT( 60. * FINTIM) ) THEN
         TERMNL = .TRUE.
      ENDIF

      HOUR = FLOAT(DAYMIN) / 60.
      RETURN
      END
*
```

Appendix V:

Listing of photosynthesis-based leaf transpiration routines



Routine	Calculation of
CPHTRAN	canopy transpiration and energy balance, similar to CANOPF (see INTKAM); calling either LFBALCI or LFBALBB
LFBALCI	leaf energy balance and transpiration, based on Ci/Cs-model
CICSSEA2	calculation of new value of gs from old value
LENER	leaf energy balance from given stomatal conductance
FARPHOT2	leaf net and gross photosynthesis (model Farquhar et al.), from given stomat conductance
LFBALBB	leaf energy balance and transpiration, based on model Ball et al.
RTWI	routine for solving implicit function Ci = f(Ci)
EBAL	calculation of new value of Ci from old value
LENER	leaf energy balance from given stomatal conductance
FARPHOT3	leaf net and gross photosynthesis (model Farquhar et al.), from given Ci
FARPHOT2	leaf net and gross photosynthesis (model Farquhar et al.), from given stomat conductance

(no listing of CPHTRAN is given)

```
************************************************************************
* SUBPROGRAM: LFBALCI
* Date: 16-Jan-1994
* Purpose: Calculation of leaf energy balance, transpiration and
*     gross photosynthesis.
*
* Description: leaf energy balance is calculated with Penman-Monteith
*  equation. Stomatal conductance is calculated from setpoint of
*  internal CO2 concentration. The setpoint is related to the CO2
*  concentration at the leaf surface. Leaf photosynthesis is calculated
*  based on model of Farquhar et al. (1980). An iteration loop
*  is done to find equilibrium leaf conductance and leaf surface
*  conditions.
*
* Input:
*  PARABS  : (R4)  absorbed PAR energy flux                   [J m-2 s-1]
*  CO2AIR  : (R4)  CO2 concentration                          [mumol mol-1]
*  NIRABS  : (R4)  absorbed NIR energy flux                   [J m-2 s-1]
*  HFPC    : (R4)  thermal radiation pipe to leaf             [J m-2 s-1]
*  HFSC    : (R4)  thermal radiation ground to leaf           [J m-2 s-1]
*  HFCR    : (R4)  thermal radiation leaf to roof             [J m-2 s-1]
*  RRAD_TOP  : (R4)  resistance for thermal radiation coming
*                       from above canopy                     [s m-1]
*  RRAD_BOT  : (R4)  resistance for thermal radiation coming
*                       from below canopy                     [s m-1]
*  TEMPAIR : (R4)  temperature of air                         [oC]
*  RB      : (R4)  boundary layer resistance for vapour       [s m-1]
*  RCUT    : (R4)  cuticula resistance for vapour             [s m-1]
*  GMAXD   : (R4)  maximal leaf conductance at night          [m s-1]
*  GNVPD   : (R4)  parameter for leaf surface VPD response
*                    of GMAXD                                 [kPa-1]
*  GS      : (R4)  stomatal conductance to H2O diffusion      [m s-1]
*  VCMAX25 : (R4)  maximal carboxylation velocity at 25 oC
*                                                  [mumol CO2 m-2 s-1]
*  JMAX25  : (R4) maximal rate of electron transport, at 25 oC
*                                                  [mumol e- m-2 s-1]
*  KC25    : (R4) Michaelis Menten constant for CO2
*                 binding to RuBP under standard conditions    [mumol mol-1]
*  KO25    : (R4) Michaelis Menten constant for O2 binding
*                to RuBP under standard conditions          [mmol mol-1]
*  THETA   : (R4) param. for degree of curvature of light response of
*                 electron transport                           [-]
*  RD25    : (R4) dark respiration at 25 oC                 [mg CO2 m-2 s-1]
*  FCICS   : (R4) factor for dependence Ci on Cs               [-]
*  FCVPD   : (R4) param. for response FCICS on leaf surface VPD  [-]
*  GLEAF0  : (R4) leaf conductance at zero leaf gross phot.      [m s-1]
*
* Output:
*  TRANLEAF  : (R4)  leaf transpiration                     [mg H2O m-2 s-1]
*  TLEAF     : (R4)  leaf temperature                          [oC]
*  CONV      : (R4)  convective heat loss from leaf            [J m-2 s-1]
*  VPDsurf   : (R4)  VPD at leaf surface                       [kPa]
*  GLEAF     : (R4)  leaf conductance                          [m s-1]
*  CO2I      : (R4)  internal CO2 concentration                [mumol mol-1]
*  CO2SURF   : (R4)  CO2 concentration at leaf surface         [mumol mol-1]
*  PG        : (R4)  leaf gross photosynthesis                [mumol m-2 s-1]
*  PN        : (R4)  leaf net photosynthesis                  [mumol m-2 s-1]
*
* Subprograms called:
*       CICSSEA, FARPHOT3, LENER
*
* Comments: for minimum searching no special routine is used
*         (as in LFBALBB), but a simple DO-LOOP is applied
************************************************************************
      SUBROUTINE LFBALCI( PARabs, NIRabs, HFPC, HFSC, HFCR,
     &             TEMPair,
     &             RRAD_TOP, RRAD_BOT, VPDair, CO2air,
     &             GS, Gleaf, Tleaf,
     &             VPDsurf, CO2i, CO2surf,
     &             Pg, Pn, TRANleaf,
     &             CONVH )
      IMPLICIT REAL(A-Z)

      INTEGER I, NITER

      COMMON /LEAFPAR/ Rcut, Rb
```

```
      COMMON /LEAFPAR_mol/ Rcut_mol, Rb_mol
      COMMON /LEAFPAR2/ Gmaxd, GNVPD
      COMMON /STOMNI2/ Gleafd

*     Link with CPHTRAN
      COMMON /FARQ_PAR4/ VCMAX25, JMAX25, RD25
      COMMON /FARQ_PAR4B/ KC25, KO25, THETA, LGHTCON

*     Link with FARPHOT2
      COMMON /PHOT_VAR/ RD, GAMMA

      LOGICAL LIGHT
      COMMON /GENCOM/ LIGHT

*     Link with CICSSEA
      COMMON /CICSSEAa/ e_PARabs, e_NIRabs, e_HFPC, e_HFSC, e_HFCR,
     &                  e_RRAD_TOP, e_RRAD_BOT
      COMMON /CICSSEAb/ e_TEMPair, e_CO2air, e_VPDair
      COMMON /CICSSEAc/ e_CO2surf, e_VPDsurf
      COMMON /CICSSEAd/ e_Tleaf

      INTEGER ITCNT,IERR

      SAVE FRACT2
      EXTERNAL CICSSEA

      DATA FRACT2, EPS, ITCNT / 0.72, 0.0001, 50 /
      DATA NITER /8/

      Gcut = 1. / Rcut

      IF( .NOT. LIGHT ) THEN
* -----------------------------------------------------------------------

*----     Loop to find equilibrium value of leaf conductance at night
*         exit loop when VPDsurf changes little

          DO I = 1, NITER
*         Initial value from last call to LFBALCI
          VPDsurfi = VPDsurf
*
          Gleafd = Gmaxd * exp( - GNVPD * VPDsurf )

*----     Energy balance
          CALL LENER( PARabs, NIRabs, HFPC, HFSC, HFCR, TEMPair,
     &                Gleafd, Rb, RRAD_TOP, RRAD_BOT, VPDair,
     &                TRANleaf, TLeaf, CONVH )
          TRANleaf = AMAX1( 0.001, TRANleaf )

          VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )

*----     Relative humidity at leaf surface
          k = 2.17 / (Tleaf + 273.) * 1.E+6
          VPSATsurf = VPleaf
          VPsurf = AMAX1( 0.1, VPSATsurf - TRANleaf / Gleafd / k )
          VPDsurf = VPSATsurf - VPsurf
          IF( ABS( VPDsurf - VPDsurfi ) .LT. .02 ) THEN
             GOTO 5
          ENDIF
      END DO
5     CONTINUE

          Gleaf = Gleafd

*         Relative humidity as a ratio !!
          RHsurf = VPsurf / VPleaf

*         66405. is activation energy (J mol-1)
*           (see routine TEMPCF from Farquhar model)
          PN   = - RD25 * TEMPDEP1( Tleaf, 66405. )

          GAMMA = (42.7 + 1.68 * (Tleaf - 25.) +
     &          0.012 * (Tleaf-25.)**2)

*         CO2 concentration at leaf surface
          CO2surf = CO2air - Pn * 1.37 * Rb_mol
```

```
          Gs = Gleafd - Gcut
          Gs_mol = Gs * 40.
          Rs_mol = 1. / Gs_mol

          CO2i = CO2surf - Pn * 1.6 /  Gs_mol
          CO2i = AMIN1( CO2i, 999. )

          RETURN

       ELSE
* ----------------------------------------------------------------------
*      LIGHT is .TRUE.

       Gs_mol = Gs * 40.
*
       Gleaf = Gs + Gleaf0

       Rs_mol = 1. / Gs_mol

*      Link with CICSSEA
       e_PARabs = PARabs
       e_NIRabs = NIRabs
       e_HFPC = HFPC
       e_HFSC = HFSC
       e_HFCR = HFCR
       e_RRAD_TOP = RRAD_TOP
       e_RRAD_BOT = RRAD_BOT
       e_TEMPair = TEMPair
       e_CO2air = CO2air
       e_VPDair = VPDair
       e_Tleaf = Tleaf
       e_VPDsurf = VPDsurf

*----  Loop to find equilibrium value of leaf conductance
       DO I = 1, NITER
         Gleaf_in = Gleaf
         Gleaf = CICSSEA( Gleaf_in )
         IF( ABS( Gleaf_in - Gleaf ) / Gleaf .LT.  0.002 ) THEN
            GOTO 15
         ENDIF
       ENDDO
15     CONTINUE

*      Leaf conductance at night is used for lower limit of gs
*      VPDsurf is saved from last call to LFBALCI
       Gleafd = Gmaxd * exp( - GNVPD * VPDsurf )
       Gleaf = AMAX1( Gleaf, Gleafd )
       Gs = AMAX1( 0.0002, Gleaf - Gcut )
       Gs_mol = Gs * 40.

       Rs_mol = 1. / Gs_mol
       Rs = 1. / Gs

*----  Calculated net photosynthesis rate (mumol m-2 s-1)
*      FARPHOT2 is same as FARPHOT, but has photosynthesis expressed
*      in mumol CO2 m-2 s-1
       CALL FARPHOT2( PARabs, CO2air, Tleaf, Gs_mol, Rb_mol,
     &                KC25, KO25, VCMAX25, JMAX25,
     &                RD25, THETA, LGHTCON, CO2i, Pg, Pn )
*      CO2 concentration at leaf surface
       CO2surf = CO2air - Pn * 1.37 * Rb_mol

*----  Energy balance
       CALL LENER( PARabs, NIRabs, HFPC, HFSC, HFCR, TEMPair,
     &                     Gleaf, Rb, RRAD_TOP, RRAD_BOT, VPDair,
     &                     TRANleaf, TLeaf, CONVH )
       TRANleaf = AMAX1( 0.001, TRANleaf )

       VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )
       VPSATsurf = VPleaf

*      Relative humidity at leaf surface
       k = 2.17 / (Tleaf + 273.) * 1.E+6
       VPsurf = AMAX1( 0.1, VPSATsurf - TRANleaf / Gleaf / k )
       VPDsurf = VPSATsurf - VPsurf

      ENDIF
```

```
*         of LIGHT
* -----------------------------------------------------------------------

      RETURN
      END
*************************************************************************
* Subprogram: CICSSEA
* Purpose: calculation of new leaf conductance from old leaf conductance
*        based on given relation between internal and external
*        CO2 concentration
*************************************************************************
      REAL FUNCTION CICSSEA( Gleaf_in )
      IMPLICIT REAL(A-Z)

      COMMON /CICSSEAa/ PARabs, NIRabs, HFPC, HFSC, HFCR,
     &                  RRAD_TOP, RRAD_BOT
      COMMON /CICSSEAb/ TEMPair, CO2air, VPDair
      COMMON /CICSSEAc/ CO2surf, VPDsurf
      COMMON /CICSSEAd/ Tleaf

      COMMON /LEAFPAR_mol/ Rcut_mol, Rb_mol
      COMMON /LEAFPAR2/ Gmaxd, GNVPD

      COMMON /STOMNI2/ Gleafd

      COMMON /STOMFMOD2/ FCICS, FCVPD, Gleaf0

      COMMON /FARQ_PAR4/ VCMAX25, JMAX25, RD25
      COMMON /FARQ_PAR4B/ KC25, KO25, THETA, LGHTCON
      COMMON /PHOT_VAR/ RD, GAMMA

      Gs_mol_in = (Gleaf_in - Gleaf0) * 40.
      Gleaf = Gleaf_in

*---- Find new leaf temperature from energy balance
      Rb = Rb_mol * 40.
      CALL LENER( PARabs, NIRabs, HFPC, HFSC, HFCR, TEMPair,
     &                    Gleaf, Rb, RRAD_TOP, RRAD_BOT, VPDair,
     &                    TRANleaf, TLeaf, CONVH )
      TRANleaf = AMAX1( 0.001, TRANleaf )
      VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )
      VPSATsurf = VPleaf

*     Relative humidity at leaf surface
      k = 2.17 / (Tleaf + 273.) * 1.E+6
      VPsurf = AMAX1( 0.1, VPSATsurf - TRANleaf / Gleaf / k )
      VPDsurf = VPSATsurf - VPsurf

*---- Calculated net photosynthesis rate
      CALL FARPHOT2( PARabs, CO2air, Tleaf, Gs_mol_in, Rb_mol,
     &               KC25, KO25, VCMAX25, JMAX25,
     &               RD25, THETA, LGHTCON, CO2i, Pg, Pn )

*     CO2 concentration at leaf surface
      CO2surf = AMIN1( 2000., CO2air - Pn * 1.37 * Rb_mol )

*     Effect VPD at leaf surface on Ci
      COR_VPD = EXP( - FCVPD * VPDsurf )

*     GAMMA from last calculation with FARPHOT2
      CO2i_setp = FCICS * ( CO2surf - GAMMA ) * COR_VPD + GAMMA

*---- Next value of leaf conductance
      Gleaf = Gleaf0 + 1.6 * Pg / ( CO2surf - CO2i_setp ) / 40.

      CICSSEA = Gleaf

      RETURN
      END
*************************************************************************
* SUBPROGRAM: LFBALBB
*
* Purpose: Calculation of leaf energy balance, transpiration and
*     gross photosynthesis.
*
* Description: leaf energy balance is calculated with Penman-Monteith
*  equation. Stomatal conductance is calculated based on model of
```

```
*  Ball et al. (1987). Leaf photosynthesis is calculated
*  based on model of Farquhar et al. (1980). An iteration loop
*  is done to find equilibrium internal CO2 concentration and leaf surface
*  conditions.
*
* Input:
*  PARABS    : (R4)  absorbed PAR energy flux                    [J m-2 s-1]
*  CO2AIR    : (R4)  CO2 concentration                           [mumol mol-1]
*  NIRABS    : (R4)  absorbed NIR energy flux                    [J m-2 s-1]
*  HFPC      : (R4)  thermal radiation pipe to leaf              [J m-2 s-1]
*  HFSC      : (R4)  thermal radiation ground to leaf            [J m-2 s-1]
*  HFCR      : (R4)  thermal radiation leaf to roof              [J m-2 s-1]
*  RRAD_TOP  : (R4)  resistance for thermal radiation coming
*                       from above canopy                        [s m-1]
*  RRAD_BOT  : (R4)  resistance for thermal radiation coming
*                       from below canopy                        [s m-1]
*  TEMPAIR   : (R4)  temperature of air                          [oC]
*  RB        : (R4)  boundary layer resistance for vapour        [s m-1]
*  RCUT      : (R4)  cuticula resistance for vapour              [s m-1]
*  GMAXD     : (R4)  maximal leaf conductance at night           [m s-1]
*  GNVPD     : (R4)  parameter for leaf surface VPD response
*                    of GMAXD                                    [kPa-1]
*  GS        : (R4)  stomatal conductance to H2O diffusion       [m s-1]
*  VCMAX25   : (R4)  maximal carboxylation velocity at 25 oC
*                                                    [mumol CO2 m-2 s-1]
*  JMAX25    : (R4) maximal rate of electron transport, at 25 oC
*                                                    [mumol e- m-2 s-1]
*  KC25      : (R4) Michaelis Menten constant for CO2
*                 binding to RuBP under standard conditions    [mumol mol-1]
*  KO25      : (R4) Michaelis Menten constant for O2 binding
*                  to RuBP under standard conditions           [mmol mol-1]
*  THETA     : (R4) param. for degree of curvature of light response of
*                 electron transport             [-]
*  RD25      : (R4) dark respiration at 25 oC                  [mg CO2 m-2 s-1]
*  m         : (R4) parameter model Ball et al.                [-]
*  b         : (R4) parameter model Ball et al.                [mol m-2 s-1]
*
* Output:
*  TRANLEAF  : (R4)  leaf transpiration                       [mg H2O m-2 s-1]
*  TLEAF     : (R4)  leaf temperature                            [oC]
*  CONV      : (R4)  convective heat loss from leaf              [J m-2 s-1]
*  VPDsurf   : (R4)  VPD at leaf surface                         [kPa]
*  GLEAF     : (R4)  leaf conductance                            [m s-1]
*  CO2I      : (R4)  internal CO2 concentration                  [mumol mol-1]
*  CO2SURF   : (R4)  CO2 concentration at leaf surface           [mumol mol-1]
*  PG        : (R4)  leaf gross photosynthesis                 [mumol m-2 s-1]
*  PN        : (R4)  leaf net photosynthesis                   [mumol m-2 s-1]
*
*  SUBPROGRAMS CALLED:
*                         EBAL
*                         LENER
*                         FARPHOT2
***********************************************************************
      SUBROUTINE LFBALBB( PARabs, NIRabs, HFPC, HFSC, HFCR,
     &            TEMPair,
     &            RRAD_TOP, RRAD_BOT, VPDair, CO2air,
     &            GS, Gleaf, Tleaf,
     &            VPDsurf, CO2i, CO2surf,
     &            Pg, Pn, TRANleaf, CONVH )
      IMPLICIT REAL(A-Z)

      INTEGER I, NITER

      COMMON /TEST_TIME/ DAYNR, HOUR

      COMMON /LEAFPAR/ Rcut, Rb
      COMMON /LEAFPAR_mol/ Rcut_mol, Rb_mol
      COMMON /LEAFPAR2/ Gmaxd, GNVPD
      COMMON /STOMNI2/ Gleafd

*     Link with CPHTRAN
      COMMON /FARQ_PAR4/ VCMAX25, JMAX25, RD25
      COMMON /FARQ_PAR4B/ KC25, KO25, THETA, LGHTCON

      COMMON /PHOT_VAR/ RD, GAMMA

      COMMON /LEAFCON_mol/ Gs_mol
```

```
      COMMON /STOMFMOD1/ b, m

      LOGICAL LIGHT
      COMMON /GENCOM/ LIGHT

* Link with EBAL
      COMMON /EBALa/ e_PARabs, e_NIRabs, e_HFPC, e_HFSC, e_HFCR,
     &       e_RRAD_TOP, e_RRAD_BOT
      COMMON /EBALb/ e_TEMPair, e_CO2air, e_VPDair
      COMMON /EBALc/ e_VPDsurf
      COMMON /EBALd/ e_Gs_mol

      INTEGER ITCNT,IERR

      SAVE FRACT
      EXTERNAL EBAL

      DATA FRACT,EPS,ITCNT / 0.72, 0.01, 50 /
      DATA NITER /10/

      Gcut = 1. / Rcut

      IF( .NOT. LIGHT ) THEN
* ----------------------------------------------------------------------
*----     Loop to find equilibrium value of leaf conductance at night
*         exit loop when VPDsurf changes little
          DO I = 1, NITER

*         Initial value from last call to LFBALBB
          VPDsurfi = VPDsurf
*
          Gleafd = Gmaxd * exp( - GNVPD * VPDsurf )

*         Energy balance
          CALL LENER( PARabs, NIRabs, HFPC, HFSC, HFCR, TEMPair,
     &                      Gleafd, Rb, RRAD_TOP, RRAD_BOT, VPDair,
     &                      TRANleaf, TLeaf, CONVH )
          TRANleaf = AMAX1( 0.001, TRANleaf )

          VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )

*         Relative humidity at leaf surface
          k = 2.17 / (Tleaf + 273.) * 1.E+6
          VPSATsurf = VPleaf
          VPsurf = AMAX1( 0.1, VPSATsurf - TRANleaf / Gleafd / k )
          VPDsurf = VPSATsurf - VPsurf
          IF( ABS( VPDsurf - VPDsurfi ) .LT. .02 ) THEN
             GOTO 5
          ENDIF
      END DO
5     CONTINUE

          Gleaf = Gleafd

*         Relative humidity as a ratio !!
          RHsurf = VPsurf / VPleaf

*         66405. is activation energy (J mol-1)
*           (see routine TEMPCF from Farquhar model)
          PN    = - RD25 * TEMPDEP1( Tleaf, 66405. )

          GAMMA = (42.7 + 1.68 * (Tleaf - 25.) +
     &          0.012 * (Tleaf-25.)**2)

*         CO2 concentration at leaf surface
          CO2surf = CO2air - Pn * 1.37 * Rb_mol

          Gs = Gleafd - Gcut
          Gs_mol = Gs * 40.
          Gs_mol = AMAX1( 0.002, Gs_mol )
          Gs = Gs_mol * .025

          CO2i = CO2surf - Pn * 1.6 / Gs_mol
          CO2i = AMIN1( CO2i, 999. )

          RETURN
```

```
      ELSE
* -------------------------------------------------------------------
*       LIGHT is .TRUE.

        Gs_mol = Gs * 40.

*       Link with EBAL
        e_PARabs = PARabs
        e_NIRabs = NIRabs
        e_HFPC = HFPC
        e_HFSC = HFSC
        e_HFCR = HFCR
        e_RRAD_TOP = RRAD_TOP
        e_RRAD_BOT = RRAD_BOT
        e_TEMPair = TEMPair
        e_CO2air = CO2air
        e_VPDair = VPDair
        e_VPDsurf = VPDsurf

*      Find substomatal CO2 concentration that makes Gs correspond
*      with Pn
*      Initial guess
       CO2IG = CO2air * FRACT
       CALL RTWI( CO2I, VAL, EBAL, CO2IG, EPS, ITCNT, IERR )
       IF( IERR .NE. 0 ) THEN
          CO2IG = 0.7 * CO2air
          CALL RTWI( CO2i, VAL, EBAL, CO2IG, EPS, ITCNT, IERR )
          IF( IERR.NE.0 .AND. Gs_mol .GT. 0.002 ) THEN
             Gleafd = Gmaxd * exp( - GNVPD * VPDsurf )
             WRITE( *, 901 ) ' Error in LFBALBB ',
     &       ' DAYNR - HOUR : ', DAYNR, HOUR,
     &       ' IERR : ', IERR,
     &       ' CO2i : ', CO2i,
     &       ' CO2IG : ', CO2IG,
     &       ' VAL : ', VAL,
     &       ' PARabs : ', PARabs,
     &       ' Tleaf : ', Tleaf,
     &       ' CO2air : ', CO2air,
     &       ' VPDair : ', VPDair,
     &       ' VPDsurf : ', VPDsurf,
     &       ' Gs_mol : ', Gs_mol,
     &       ' Gleafd : ', Gleafd
901         FORMAT( A, A, F6.0, F7.3, /, A, I4, 2(A, F7.1), A, F10.3,/,
     &         5(A, F9.3), /, 2(A, F9.5)  )
             STOP
          ENDIF
       ENDIF

*----  Satisfy other constraints
       Gleafd = Gmaxd * exp( - GNVPD * VPDsurf )
       Gs_mol = AMAX1( Gs_mol, (Gleafd - Gcut) * 40. )
       Gs_mol = AMAX1( 0.005, Gs_mol )

*----  Equilibrium values
       Gs = Gs_mol * .025
       Rs_mol = 1. / Gs_mol
       Rs = 1. / Gs

*----  Calculated net photosynthesis rate (mumol m-2 s-1)
*      FARPHOT2 is same as FARPHOT, but has photosynthesis expressed
*      in mumol CO2 m-2 s-1
       CALL FARPHOT2( PARabs, CO2air, Tleaf, Gs_mol, Rb_mol,
     &                KC25, KO25, VCMAX25, JMAX25,
     &                RD25, THETA, LGHTCON, CO2i, Pg, Pn )

       Gleaf = Gcut + Gs

*----  Energy balance
       CALL LENER( PARabs, NIRabs, HFPC, HFSC, HFCR, TEMPair,
     &                     Gleaf, Rb, RRAD_TOP, RRAD_BOT, VPDair,
     &                     TRANleaf, TLeaf, CONVH )
       TRANleaf = AMAX1( 0.01, TRANleaf )

       VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )
       VPSATsurf = VPleaf

*      Relative humidity at leaf surface
```

```
      k = 2.17 / (Tleaf + 273.) * 1.E+6
      VPsurf = AMAX1( 0.1, VPSATsurf - TRANleaf / Gleaf / k )
      VPDsurf = VPSATsurf - VPsurf

*     Relative humidity as a ratio !!
      RHsurf = VPsurf / VPleaf

*     CO2 concentration at leaf surface
      CO2surf = CO2air - Pn * 1.37 * Rb_mol

      ENDIF
*        of LIGHT
* ------------------------------------------------------------------

      RETURN
      END
**********************************************************************
* Subprogram: EBAL
* Purpose: calculation of new Ci from old Ci
*       based on relation of Ball et al for photosynthesis-based
*       stomatal conductance
*
* Subprograms called: FARPHOT3, LENER
*
**********************************************************************
      REAL FUNCTION EBAL( CO2in )
      IMPLICIT REAL(A-Z)

      COMMON /EBALa/ PARabs, NIRabs, HFPC, HFSC, HFCR,
     &               RRAD_TOP, RRAD_BOT
      COMMON /EBALb/ TEMPair, CO2air, VPDair

      COMMON /LEAFPAR_mol/ Rcut_mol, Rb_mol
      COMMON /LEAFPAR2/ Gmaxd, GNVPD
      COMMON /STOMNI2/ Gleafd

      COMMON /FARQ_PAR4/ VCMAX25, JMAX25, RD25
      COMMON /FARQ_PAR4B/ KC25, KO25, THETA, LGHTCON

      COMMON /PHOT_VAR/ RD, GAMMA

*     Parameters model Ball et al.
      COMMON /STOMFMOD1/ b, m

      COMMON /LEAFCON_mol/ Gs_mol

      LOGICAL LIGHT
      COMMON /GENCOM/ LIGHT

      CO2i = CO2in

      Gleaf = (Gs_mol + 1./ Rcut_mol) * 0.025

*     Energy balance
      Rb = Rb_mol * 40.
      CALL LENER( PARabs, NIRabs, HFPC, HFSC, HFCR, TEMPair,
     &            Gleaf, Rb, RRAD_TOP, RRAD_BOT, VPDair,
     &            TRANleaf, TLeaf, CONVH )
      TRANleaf = AMAX1( 0.01, TRANleaf )

      VPleaf = .6107 * EXP( 17.4 * Tleaf / (Tleaf + 239.) )

*     Relative humidity at leaf surface
      VPSATsurf = VPleaf
      k = 2.17 / (Tleaf + 273.) * 1.E+6
      VPsurf = AMAX1( 0.1, VPSATsurf - TRANleaf / Gleaf / k )

      VPDsurf = VPSATsurf - VPsurf

*     Relative humidity as a ratio !!
      RHsurf = VPsurf / VPleaf

*---- Calculated net photosynthesis rate (mumol CO2 m-2 s-1)
      CALL FARPHOT3( PARabs, CO2air, Tleaf, CO2i,
     &               KC25, KO25, VCMAX25, JMAX25,
     &               RD25, THETA, LGHTCON, Pg, Pn )
```

```
*     CO2 concentration at leaf surface
      CO2surf = CO2air - Pn * 1.37 * Rb_mol

*     Stomatal conductance (mol m-2 s-1)
      Gs_mol = AMAX1( 0.001,
     &            m * Pn * RHsurf / ( CO2surf-GAMMA ) + b )
      Rs_mol = 1. / Gs_mol

      CO2i_next = AMAX1( GAMMA + 1.,
     &          CO2air - Pn * (RS_mol*1.6 + RB_mol*1.37) )

      EBAL = CO2i_next

      RETURN
      END
************************************************************************
* SUBPROGRAM: TEMPDEP1
*
* Pupose: calculation of temperature response for given activation
*       energy
* Input:
*   TEMP    : (R4)  temperature                          [oC]
*   ACTENER : (R4)  activation enery                     [J mol-1]
*
* Output:
*   TEMPDEP1 : (R4)  relative temperature effect
*                    (scaled to 25 oC)                   [-]
************************************************************************
      REAL FUNCTION TEMPDEP1( TEMP, ACTENER )
      IMPLICIT REAL( A-Z )

      TEMPabs = TEMP + 273.

*     4.0335E-4 = 1/R(25+273.2)
      COR1 = 4.0355E-4 * (TEMPabs-298.2) / (TEMPabs + 273.)
      TEMPDEP1 = EXP( ACTENER * COR1 )

      RETURN
      END
*
```

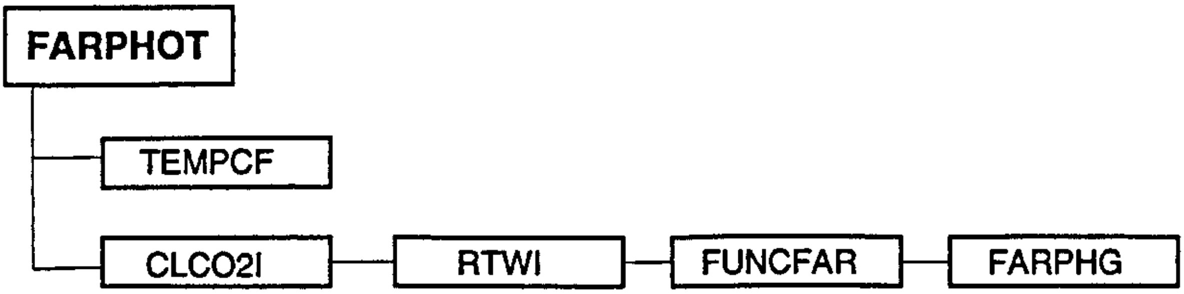
Appendix VI:
Listing of subroutine LPHOT

```
************************************************************************
* SUBPROGRAM: LPHOT
* Type: SUBROUTINE
* Date: 05-Oct-1993
* Author: H. Gijzen
*
* Purpose:
*   Calculation of leaf gross photosynthesis from absorbed PAR energy,
*   CO2 concentration and leaf temperature
*
* Description: descriptive formulae are used to calculated initial slope
*        and light-saturation value of negative-exponential light
*        response curve. Formulae are developed partly based on theory
*        of Farquhar, von Caemmerer and Berry (1980).
*
* Origin: J. Goudriaan (Kollegediktaat, 1989), and Goudriaan et al., 1985
*
* Input:
*  PARABS    : (R4)  absorbed PAR energy flux                  [J m-2 s-1]
*  CO2AIR    : (R4)  CO2 concentration                         [mumol mol-1]
*  TLEAF     : (R4)  leaf temperature                          [oC]
*  GS        : (R4)  stomatal conductance to H2O diffusion     [m s-1]
*  VCMAX25   : (R4)  maximal carboxylation velocity at 25 oC
*                                                        [mumol CO2 m-2 s-1]
*  JMAX25    : (R4) maximal rate of electron transport, at 25 oC
*                                                        [mumol e- m-2 s-1]
*  FCO2CURV  : (R4)param. for degree of curvature of CO2 response of
*                light saturated net photosynthesis              [-]
*  RD25      : (R4) dark respiration at 25 oC                [mg CO2 m-2 s-1]
*
* Output:
*  PGROSL    : (R4)  leaf gross photosynthesis              [mg CO2 m-2 s-1]
*
* SUBPROGRAMS CALLED:  none
*
* COMMENT:
*  Parameters:
*  EFF0     : Potential light use efficiency in absence of
*                 oxygen (mg CO2 (mumol photons)-1 )
*  RB       : boundary layer resistance to H20 diffusion (s m-1)
*  VCMAX25 : maximal carboxylation velocitym at 25 oC  (mumol CO2 m-2 s-1)
*  JMAX25  : maximal rate of electron transport, at 25 oC (mumol e- m-2 s-1)
*  FCO2CURV   : parameter for degree of curvature of CO2 response of
*                light saturated net photosynthesis
*  RD25     : dark respiration at 25 oC (mg CO2 m-2 s-1)
*  KM25     : effective M.M.-constant of Rubisco at 25 oC
*  Q10RD    : Q10 of dark respiration
*  Q10KM    : Q10 of effective M.M. constant Rubisco
*  Q10VCM   : Q10 of carboxylation velocity
************************************************************************
      SUBROUTINE LPHOT( PARABS, CO2AIR, TLEAF, GS,
     &                  VCMAX25, JMAX25, RD25, FCO2CURV,
     &                  PGROSL )
      IMPLICIT REAL(A-Z)

      PARAMETER( EFF0 = 0.0037,  RB=100.,  Q10RD=2.0 )
      PARAMETER( KM25 = 1300., Q10KM = 1.7, Q10VCM = 2.2 )

*     Stomatal resistance to H2O diffusion (s m-1)
      RS = 1. / GS
*     Difference leaf temperature with 25 oC
      TEMPDIF = TLEAF - 25.

*---- Carboxylation resistance; is dependent on temperature
*     Effective M-M constant (mg CO2 m-3) of Rubisco for
*      CO2 at 210 ml O2 l-1
      KM = KM25 * Q10KM ** ( 0.1 * TEMPDIF )
```

```
*     Maximal rate of carboxylation (mg CO2 m-2 s-1)
      VCMAX = VCMAX25 * Q10VCM ** ( 0.1 * TEMPDIF )
*     Carboxylation resistance (s m-1)
      RC = KM / VCMAX

*---  Endogenous photosynthetic capacity PMM (mg CO2 m-2 s-1)
*     is a function of temperature;
*     is approximately scaled to JMAX25 of model Farquhar et al.
*     0.011 converts mumol electrons to mg CO2
      PMM = 0.
      IF( TLEAF .LT. 25. ) THEN
          PMM = JMAX25 * 0.011 * (TLEAF - 5.) / 20.
      ELSEIF( TLEAF .LT. 35. ) THEN
          PMM = JMAX25 * 0.011
      ELSEIF( TLEAF .LT. 45. ) THEN
          PMM = JMAX25 * 0.011 * (1. - (TLEAF - 35.) / 10. )
      ELSE
          PMM = 0.
      ENDIF

*---  CO2 compensation point increases with temperature
*     dependance according to Brooks & Farquhar, 1985
      GAMMA = 42.7 + 1.68 * TEMPDIF + 0.012 * TEMPDIF**2

*---  Reduction of licht use efficiency by photorespiration;
*     affected by CO2 concentration
      CO2 = MAX( CO2AIR, GAMMA )

*     Efficiency in mg CO2 per mumol photons
      EFF = EFF0 * (CO2-GAMMA) / (CO2+2.*GAMMA)

*---  PNC is maximum as determined by CO2 diffusion
*     1.830 mg CO2 per m3 per mul 1-1
*     Stomatal resistance and boundary layer resistance to CO2 are
*     1.6 and 1.37 times larger than to water vapour, respectively
      PNC = (CO2-GAMMA) * 1.830 / (1.37*RB + 1.6*RS + RC)

*---  PNMAX shows saturation with PNC
      IF (PMM .LT. 0.00001) THEN
           PNMAX = 0.0
      ELSE
           PNMAX = ( PNC + PMM
     &          - SQRT( (PNC+PMM)**2 - 4. * FCO2CURV * PNC * PMM ) )
     &     / (2. * FCO2CURV )
      ENDIF

*---  Dark respiration (mg CO2 m-2 s-1)
      RD = RD25 * Q10RD**( 0.1 * TEMPDIF )

*---  PGMAX (mg CO2 m-2 leaf s-1) is determined by
*     maximal net assimilation PNMAX and RD
      PGMAX = PNMAX + RD

*---  Gross leaf photosynthetic CO2 assimilation (mg CO2 m-2 leaf s-1)
      PGROSL = PGMAX * (1. - EXP( -EFF * PARABS / PGMAX ) )

      RETURN
      END
```

Appendix VII:

Listing of the subroutines containing the model of Farquhar et al. (subroutine FARPHOT)



Routine	Calculation of
TEMPCF	temperature dependencies of parameters in model Farquhar et al.
CLCO2I	calculation of internal CO2 concentration (Ci)
RTWI	routine for solving implicit function Ci = f(Ci)
FUNCFAR	calculation of Ci from Ca and leaf net photosynthesis
FARPHG	calculation of leaf gross photosynthesis from given Ci

```
************************************************************************
* SUBPROGRAM: FARPHOT
* Date: 29-Mar-1993
* Purpose:
*  Computation of leaf photosynthesis according to model of
*  Farquhar et al. (1980)
*
* Description:
*  Leaf gross photosynthesis is calculated from Rubisco limited
*  carboxylation rate and RuBP regeneration limited carboxylation rate.
*  No Pi-regeneration limitation is assumed. Internal CO2 concentration
*  is calculated by implicit equation solver (RTWI)
*
* Input:
*  PARABS  : (R4) absorbed PAR                              [J m-2 s-1]
*  TLEAF   : (R4) leaf temperature                          [oC]
*  CO2AIR  : (R4) CO2 concentration                         [mumol mol-1]
*  GS_mol  : (R4) stomatal conductance to H2O diffusion    [mol m-2 s-1]
*  RB_mol  : (R4) boundary layer resistance to H2O          [s m2 mol-1]
*  KC25    : (R4) Michaelis Menten constant for CO2
*                 binding to RuBP under standard conditions  [mumol mol-1]
*  KO25    : (R4) Michaelis Menten constant for O2 binding
*                    to RuBP under standard conditions       [mmol mol-1]
*  VCMAX25 : (R4)  maximal carboxylation velocity at 25 oC
*                                                     [mumol CO2 m-2 s-1]
*  JMAX25  : (R4) maximal rate of electron transport, at 25 oC
*                                                      [mumol e- m-2 s-1]
*  THETA   : (R4) param. for degree of curvature of light response of
*                 electron transport                            [-]
*  RD25    : (R4) dark respiration at 25 oC              [mumol CO2 m-2 s-1]
*
* Output:
*  PGROSL    : (R4) leaf gross photosynthesis              [mg CO2 m-2 s-1]
*  PNETL     : (R4) leaf net assimilation                  [mg CO2 m-2 s-1]
*  CO2IN     : (R4) leaf internal CO2 concentration        [mumol mol-1]
*
* Subroutines called: TEMPCF,  CLCO2I
*   ( other routines are: FUNCFAR, FARPHG and RTWI )
*
* Comment:
* References are:
* F,1980  = A biochemical model of photosynthetic CO2 assimilation
*           in leaves of C3 species.
*           Farquhar G.D., Caemmerer S. von, Berry J.A.
*           Planta 149, 78-90 (1980).
* F,1982  = Modelling of photosynthetic response to environmental
*           conditions.
*           Farquhar G.D., Caemmerer S. von
*           In: Encyclopedia of plant physiology new series vol. 12B
*           pp. 549-582
************************************************************************
      SUBROUTINE FARPHOT( PARABS, CO2AIR, TLEAF, GS_mol,
     &             KC25, KO25, VCMAX25, JMAX25,
     &             RD25, THETA, LGHTCON, CO2in, PGROSL, PNETL )
      IMPLICIT REAL(A-Z)
*
*     Links with FUNCFAR.FOR
      COMMON /FUNCOM/ CO2I, CO2E, GAMMA, VCMAX, KC, KO,
     &         O2, RD, VC, J, VC1, VC2
      COMMON /LEAFPAR_mol/ RCUT_mol, RB_mol
*     Link with FUNCFAR
      COMMON /LEAFRES_mol/ RS_mol

      Rs_mol = 1. / Gs_mol

*     CO2 concentration
      CO2E = CO2air
*     O2 concentration (mmol mol-1)
      O2 = 210.
*
*     Temperature dependent correction factors
      CALL TEMPCF( TLEAF, TKC, TKO, TVCMAX, TRD, TJMAX )

      KC = KC25 * TKC
      KO = KO25 * TKO
```

```
*
*     Maximal carboxylation velocity
      VCMAX = VCMAX25 * TVCMAX
*     Dark respiration
      RD = RD25 * TRD

*     CO2 compensation point increases with temperature
*     according to Brooks & Farquhar, 1985
      GAMMA = (42.7 + 1.68 * (TLEAF - 25.) + 0.012 * (TLEAF-25.)**2)

*     Temperature dependent potential rate of electron transport
*     mu Eq m-2 s-1
*     (16.33) + (16.34): F,1982 see also F,1982 fig 16.7
      JMAX = JMAX25 * TJMAX

*     Calculate potential rate of electron transport (mumol e- m-2 s-1)
*     2 electrons per absorbed photon
*     Conversion of J m-2 s-1 to mumol m-2 s-1 with LGHTCON
*     F is fraction of photons absorbed by non-photosynthetic tissues
      F = .3
      EFFRAD = PARABS * LGHTCON / 2. * (1. - F)
      J = ( JMAX + EFFRAD -
     &     SQRT( (JMAX+EFFRAD)**2-4.*THETA*EFFRAD*JMAX )   )
     &     / (2. * THETA)

*     Use implicit equation solver to calculate CO2I
      CO2I = CLCO2I( CO2E )
      CO2in = CO2i
*
*     Gross photosynthesis  (mumol CO2 m-2 s-1)
      PNETL = ( CO2E-CO2I ) / ( RS_mol * 1.6 + RB_mol * 1.37 )
      PGROSL = PNETL + RD

*     Conversion to mg m-2 s-1
      PGROSL = PGROSL * 0.044
      PNETL = PNETL * 0.044

      RETURN
      END
**********************************************************************
* SUBPROGRAM FARPHG
* Purpose: calculation of leaf gross photosynthesis according to
*   model Farquhar et al. (1980)
*
* Input:
*  CO2I   : (R4) internal CO2 concentration          [mumol mol-1]
*  O2     : (R4) internal O2 concentration           [mmol mol-1]
*  GAMMA  : (R4) CO2 compensation point in absence of
*                photorespiration                    [mumol mol-1]
*  KC     : (R4) M.M. constant for CO2 binding to RuBP [mumol mol-1]
*  KO     : (R4) M.M. constant for O2 binding to RuBP  [mmol mol-1]
*  VC     : (R4) maximal carboxylation rate          [mumol m-2 s-1]
*  J      : (R4) potential electron transport rate   [mumol m-2 s-1]
*
* Output:
*  FARPHG : (R4) Leaf gross photosynthesis        [mumol CO2 m-2 s-1]
*
* Comments: no mesophyll resistance is assumed
**********************************************************************
      REAL FUNCTION FARPHG( CO2I, GAMMA, VCMAX, KC, KO, O2,
     &             VC, J )
      IMPLICIT REAL(A-Z)
*
*     JC:  actual carboxylation electron transport velocity
      JC = J
*
*     Calculate RuP2-saturated rate of carboxylation (16.59) F,1982
*     (is limiting rate of carboxylation)
      VC1 = VCMAX * CO2I / ( CO2I + KC * (1.+O2/KO) )
*
*---  Calculate electron transport/photophosphorylation limited rate of
*     RuP2 regeneration
*        ( division by (4.5*CO2i+10.5*GAMMA)
*            assumes pseudocyclic electron transport )
*      VC2 = JC * CO2I / ( 4.5*CO2I + 10.5*GAMMA )
*
      VC2 = JC / 4. * CO2I / ( CO2I + 2. * GAMMA )
```

```
*
*---- Compute actual carboxylation velocity
      VC = MIN( VC1, VC2 )
*
*     Compute photorespiration (16.3)+(16.18) F,1982
      FRESP = VC * GAMMA / CO2I
*
*---- Leaf gross photosynthesis (mumol CO2 m-2 s-1) (16.57) F,1982
      FARPHG = VC - FRESP

      RETURN
      END
**********************************************************************
* Subprogram: FUNCFAR
*
*     Function to describe dependence between CO2I, CO2E,
*     gross photosynthesis and resistance to CO2
*
*     Input and output are CO2 concentration; this implicit function
*     must be called by implicit function solver
*
**********************************************************************
      REAL FUNCTION FUNCFAR( CO2IN )
      IMPLICIT REAL(A-Z)
*
      COMMON /FUNCOM/ CO2I, CO2E, GAMMA, VCMAX, KC, KO,
     &       O2, RD, VC, J, VC1, VC2

      COMMON /LEAFRES_mol/ RS_mol
      COMMON /LEAFPAR_mol/ RCUT_mol, RB_mol
*
*
      CO2I = CO2IN
*---- Leaf gross photosynthesis (mumol CO2 m-2 s-1)
      AUX = FARPHG( CO2I, GAMMA, VCMAX, KC, KO, O2,
     &           VC, J )

*     Internal CO2 concentration (mumol mol-1)
      X = AMAX1( 0.1, CO2E - (AUX-RD) * (RS_mol * 1.6 + RB_mol*1.37) )

      FUNCFAR = X
      RETURN
      END
****************************************************************
* SUBPROGRAM: TEMPCF
* Type: subroutine
* Purpose:
*  subroutine to calculate temperature dependence of
*  parameters in model of Farquhar et al.
*
* Input:
*   TLEAF   : (R4) leaf temperature                        [oC]
* Output:
*   TKC     : temperature correction for KC25              [-]
*   TKO     : temperature correction for KO25              [-]
*   TRN     : temperature correction for RD25              [-]
*   TVCMAX  : temperature correction for VCMAX25           [-]
*   TJMAX   : temperature correction for JMAX25            [-]
**********************************************************************
      SUBROUTINE TEMPCF( TLEAF, TKC, TKO, TVCMAX, TRN, TJMAX )
      IMPLICIT REAL(A-Z)

      SAVE
*
*     Activation energies for binding CO2 and O2 to Rubisco, maximal
*     carboxylation rate, potential rate of electron transport and
*     dark respiration rate, resp.
*     dimensions : J mol-1
*     p. 88: F,1980
      PARAMETER( EC = 59356., EO=35948., EVC=58520., EJ=37000.,
     &           ED = 66405.)
*
*     Gasconstant (J mol-1 K-1)
      PARAMETER( R = 8.314 )
*
*     Constants for optimum curve of temperatur dependent rate of
*     electron transport; S is entropy term, H is deactivation energy
```

```
*     p.88-89: F,1980
      PARAMETER( S=710., H=220000. )
*
*
      DATA CTABS / -99. /
*
*     Absolute temperature
      TABS = TLEAF + 273.2
*
*     temperature dependencies according to F,1980 (eq. 35),
*     TJMAX according to erratum on F,1982
      IF( TABS .NE. CTABS ) THEN

*        4.0335E-4 = 1. / (R * (25+273.2) )
         X      = 4.0355E-4 * (TABS-298.2) / TABS

         TKC    = EXP(EC *X)
         TKO    = EXP(EO *X)
         TVCMAX = EXP(EVC*X)
         TRN    = EXP(ED *X)

*        D      = 1+EXP((S-H/TABS)/R)
*        TJMAX = EXP(EJ*X)/D
*        TJMAX according to pers. comm. Farquhar to Ad Schapendonk (1985)
         D1     = 1. + EXP( (S-H/298.) / R )
         D2     = 1. + EXP( (S-H/TABS) / R )
         TJMAX = EXP( EJ * X ) * D1 / D2

         CTABS  = TABS
      ENDIF

      RETURN
      END
***********************************************************************
* Subprogram: CLCO2I
* Purpose:
*  function to calculate internal CO2 concentration when external
*  CO2 concentration is given
* Description:
*  an implicit function solver (RTWI) is called for soling the
*  implicit function FUNCFAR
*
*  Input is CO2 concentration in ambient air, output is equilibrium
*  internal CO2 concentration
***********************************************************************
      REAL FUNCTION CLCO2I( CO2E )
      IMPLICIT REAL(A-Z)

      INTEGER ITCNT,IERR

      SAVE FRACT

      EXTERNAL FUNCFAR

      DATA FRACT,EPS,ITCNT/ 0.72, 0.01, 50 /
*
*     perform initial guess for iteration if this is the first
*     call to FUNCFAR; if not take former FRACT (=CO2I/CO2E)
*     as initial value
      CO2IG = CO2E * FRACT
*
*     call subroutine to solve implicit equation
*     set iteration maximum (ITCNT) and precision (EPS)
      CALL RTWI( CO2I, VAL, FUNCFAR, CO2IG, EPS, ITCNT, IERR )
*
*  Test if RTWI failed; if so try once more with CO2IG=0.72*CO2E
      IF( IERR .NE. 0 ) THEN
         CO2IG = 0.72 * CO2E
         CALL RTWI( CO2I, VAL, FUNCFAR, CO2IG, EPS, ITCNT, IERR )
         IF( IERR .NE. 0 ) THEN
            WRITE(*,*)
     &      'Error in CLCO2I; IERR= ', IERR, CO2I, VAL, CO2IG
            STOP
         ENDIF
      ENDIF
*
*     Save fraction for computing CO2IG in possible next call RTWI
```

```
      IF( CO2E .NE. 0.0 ) THEN
         FRACT = CO2I/CO2E
      ELSE
         FRACT = 0.72
      ENDIF

      CLCO2I = CO2I

      RETURN
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C        SUBROUTINE RTWI
C
C        PURPOSE
C           TO SOLVE GENERAL NONLINEAR EQUATIONS OF THE FORM X=FCT(X)
C           BY MEANS OF WEGSTEIN-S ITERATION METHOD.
C
C        USAGE
C           CALL RTWI (X,VAL,FCT,XST,EPS,IEND,IER)
C           PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT.
C
C        DESCRIPTION OF PARAMETERS
C           X      - RESULTANT ROOT OF EQUATION X=FCT(X).
C           VAL    - RESULTANT VALUE OF X-FCT(X) AT ROOT X.
C           FCT    - NAME OF THE EXTERNAL FUNCTION SUBPROGRAM USED.
C           XST    - INPUT VALUE WHICH SPECIFIES THE INITIAL GUESS OF
C                    THE ROOT X.
C           EPS    - INPUT VALUE WHICH SPECIFIES THE UPPER BOUND OF THE
C                    ERROR OF RESULT X.
C           IEND   - MAXIMUM NUMBER OF ITERATION STEPS SPECIFIED.
C           IER    - RESULTANT ERROR PARAMETER CODED AS FOLLOWS
C                     IER=0 - NO ERROR,
C                     IER=1 - NO CONVERGENCE AFTER IEND ITERATION STEPS,
C                     IER=2 - AT ANY ITERATION STEP THE DENOMINATOR OF
C                             ITERATION FORMULA WAS EQUAL TO ZERO.
C
C        REMARKS
C           THE PROCEDURE IS BYPASSED AND GIVES THE ERROR MESSAGE IER=2
C           IF AT ANY ITERATION STEP THE DENOMINATOR OF ITERATION
C           FORMULA WAS EQUAL TO ZERO. THAT MEANS THAT THERE IS AT
C           LEAST ONE POINT IN THE RANGE IN WHICH ITERATION MOVES WITH
C           DERIVATIVE OF FCT(X) EQUAL TO 1.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           THE EXTERNAL FUNCTION SUBPROGRAM FCT(X) MUST BE FURNISHED
C           BY THE USER.
C
C        METHOD
C           SOLUTION OF EQUATION X=FCT(X) IS DONE BY MEANS OF
C           WEGSTEIN-S ITERATION METHOD, WHICH STARTS AT THE INITIAL
C           GUESS XST OF A ROOT X. ONE ITERATION STEP REQUIRES ONE
C           EVALUATION OF FCT(X). FOR TEST ON SATISFACTORY ACCURACY SEE
C           FORMULAE (2) OF MATHEMATICAL DESCRIPTION.
C           FOR REFERENCE, SEE
C           (1) G. N. LANCE, NUMERICAL METHODS FOR HIGH SPEED COMPUTERS,
C               ILIFFE, LONDON, 1960, PP.134-138,
C           (2) J. WEGSTEIN, ALGORITHM 2, CACM, VOL.3, ISS.2 (1960),
C               PP.74,
C           (3) H.C. THACHER, ALGORITHM 15, CACM, VOL.3, ISS.8 (1960),
C               PP.475,
C           (4) J.G. HERRIOT, ALGORITHM 26, CACM, VOL.3, ISS.11 (1960),
C               PP.603.
C
C     ..................................................................
C
      SUBROUTINE RTWI(X,VAL,FCT,XST,EPS,IEND,IER)
C
C     PREPARE ITERATION
      IER=0
      TOL=XST
      X=FCT(TOL)
      A=X-XST
      B=-A
      TOL=X
      VAL=X-FCT(TOL)
```

```
C
C     START ITERATION LOOP
      DO 6 I=1,IEND
      IF(VAL)1,7,1
C
C     EQUATION IS NOT SATISFIED BY X
    1 B=B/VAL-1.
      IF(B)2,8,2
C
C     ITERATION IS POSSIBLE
    2 A=A/B
      X=X+A
      B=VAL
      TOL=X
      VAL=X-FCT(TOL)
C
C     TEST ON SATISFACTORY ACCURACY
      TOL=EPS
      D=ABS(X)
      IF(D-1.)4,4,3
    3 TOL=TOL*D
    4 IF(ABS(A)-TOL)5,5,6
    5 IF(ABS(VAL)-10.*TOL)7,7,6
    6 CONTINUE
C     END OF ITERATION LOOP
C
C     NO CONVERGENCE AFTER IEND ITERATION STEPS. ERROR RETURN.
      IER=1
    7 RETURN
C
C     ERROR RETURN IN CASE OF ZERO DIVISOR
    8 IER=2
      RETURN
      END
```

Appendix VIII:

Listing of subroutine BIGLTR

```
***********************************************************************
* Subprogram: BIGLTR
* Purpose: calculation of canopy transpiration, assuming the canopy
*  to be a big leaf.
*
* Description:
*  Absorption of diffuse and direct PAR, NIR and UV are calculated,
*  and absorption of thermal radiation from ground, pipes and greenhouse
*  cover. Total boundary layer conductance is assumed as LAI times
*  single leaf boundary layer conductance. No aerodynamic resistance is
*  assumed. A spherical leaf angle distribution is assumed.
*  A Jarvis-type stomatal response is assumed.
*  Absorbed fluxes UV not used for calculation of transpiration; they
*  are assumed to cancel out against energy used for photosynthesis.
*
* Input:
*  PARDIF    : (R4)  flux diffuse PAR                          [J m-2 s-1]
*  PARDIR    : (R4)  flux direct PAR                           [J m-2 s-1]
*  NIRDIF    : (R4)  flux diffuse NIR                          [J m-2 s-1]
*  NIRDIR    : (R4)  flux direct NIR                           [J m-2 s-1]
*  UVDIF     : (R4)  flux diffuse UV                           [J m-2 s-1]
*  UVDIR     : (R4)  flux direct UV                            [J m-2 s-1]
*  ELEVN     : (R4)  solar elevation                           [radians]
*  SINELV    : (R4)  sine of solar elevation                   [-]
*  LAI       : (R4)  Leaf Area Index                           [-]
*  KDIFBL    : (R4)  extinction coeff. canopy with black leaves [-]
*  SCP       : (R4)  scattering coeff. of leaves for PAR       [-]
*  SCN       : (R4)  scattering coeff. of leaves for NIR       [-]
*  CO2air    : (R4)  CO2 concentration of greenhouse air       [mul 1-1]
*  TEMPAIR   : (R4)  temperature of greenhouse air             [oC]
*  VPDair    : (R4)  vapour pressure deficit of greenhouse air [kPa]
*  TPIPE     : (R4)  temperature of heating pipes              [oC]
*  SSPT      : (R4)  specific surf. of heating pipes above canopy [-]
*  SSPB      : (R4)  specific surf. of heating pipes below canopy [-]
*  TROOF     : (R4)  temperature of greenhouse cover           [oC]
*  TGROUND   : (R4)  temperature of greenhouse floor           [oC]
*  RTHRAD    : (R4)  resistance for thermal radiation at top of can. [s m-1]
*  REFGR     : (R4)  reflection coefficient of ground surface  [-]
*
* Output:
*  TRANSP    : (R4)  canopy transpiration                   [mg H2O m-2 s-1]
*  RADABS    : (R4)  absorbed short wave radiation          [J m-2 s-1]
*  NETRAD    : (R4)  net radiation of canopy                [J m-2 s-1]
*  GCAN      : (R4)  canopy conductance                     [m s-1]
*
*  Subprograms called: LONGRAD
*
***********************************************************************
      SUBROUTINE BIGLTR( PARDIF, PARDIR, NIRDIF, NIRDIR, UVDIF, UVDIR,
     &          LAI, SCP, SCN,
     &          KDIFBL, SINELV, TEMPair, VPDair, CO2air,
     &          TROOF, TGROUND, SSPT, SSPB, TPIPE, REFGR,
     &          TRANSP, RADABS, NETRAD, GCAN )
      IMPLICIT REAL(A-Z)

      LOGICAL LIGHT

      COMMON /PAR_BIGL1/ GCANmax, CCN1, CCN2, CCN3, CCN4

      COMMON /LEAFPAR/ RCUT, RB
      COMMON /LEAFPAR3/ Gmaxda, GNVPDa


      IF( PARDIF .GT. .1 ) THEN
         LIGHT = .TRUE.
      ELSE
         LIGHT = .FALSE.
```

```
      ENDIF

*---- Absorbed fluxes long wave radiation
*     Pipes at top
      HF_PC_T = LONGRAD( SSPT, TPIPE, TEMPair )
*     Pipes at bottom
      HF_PC_B = LONGRAD( SSPB, TPIPE, TEMPair )
      HF_PCTOT  = (HF_PC_T + HF_PC_B) * (1. - EXP( -KDIFBL * LAI ))

*     Canopy and Roof (positive when canopy temperature larger
*     than roof temperature)
      HF_CR = LONGRAD( 1., TEMPair, TROOF )
      HF_CRTOT = HF_CR * (1. - EXP( -KDIFBL * LAI ))
*     Ground and Canopy
*     positive when TGROUND > TCANOP
      HF_SC = LONGRAD( 1., TGROUND, TEMPair )
      HF_SCTOT = HF_SC * (1. - EXP( -KDIFBL * LAI ) )

      IF( LIGHT ) THEN
*     Absorbed fluxes short wave radiation
*     Extinction coefficients of canopy for diffuse radiation (PAR and
*     NIR)
      SQP = SQRT( 1.0 - SCP )
      KDIF = KDIFBL * SQP
      SQN = SQRT( 1.0 - SCN )
      KDIFN = KDIFBL * SQN

*     Reflection coefficient of canopy for diffuse PAR and NIR radiation
      REFHP = (1.0-SQP)/(1.0+SQP)
      REFHN = (1.0-SQN)/(1.0+SQN)

*     Extinction coefficient of canopy for direct PAR and NIR radiation
      KDIRBL = 0.5 / SINELV
      KDIR = KDIRBL * SQP
      KDIRN = KDIRBL * SQN

*     Reflection coefficient of canopy for direct PAR and NIR radiation
      REFHPD = REFHP * 1. / ( 0.5 + SINELV )
      REFHND = REFHN * 1. / ( 0.5 + SINELV )

*     Radiation reflected by ground surface
      PARREF = REFGR *
     & ( (1.-REFHP) * EXP( -KDIF * LAI ) * PARDIF
     & + (1.-REFHPD) * EXP( -KDIR * LAI ) * PARDIR )

      NIRREF = REFGR *
     & ( (1.-REFHN) * EXP( -KDIFN * LAI ) * NIRDIF
     & + (1.-REFHND) * EXP( -KDIRN * LAI ) * NIRDIR )

      UVREF = REFGR *
     & ( (1.-REFHP) * EXP( -KDIF * LAI ) * UVDIF
     & + (1.-REFHPD) * EXP( -KDIR * LAI ) * UVDIR )

      PARDIFAB = PARDIF * (1.-REFHP) * (1.- EXP( - KDIF * LAI ))
     &              + PARREF * (1. - EXP( - KDIF * LAI ))
      NIRDIFAB = NIRDIF * (1.-REFHN) * (1.- EXP( - KDIFN * LAI ))
     &              + NIRREF * (1. - EXP( - KDIFN * LAI ))
      UVDIFAB = UVDIF * (1.-REFHP) * (1.- EXP( - KDIF * LAI ))
     &              + UVREF * (1. - EXP( - KDIF * LAI ))

      PARDIRAB = PARDIR * (1.-REFHPD) * (1.- EXP( - KDIR * LAI ))
      NIRDIRAB = NIRDIR * (1.-REFHND) * (1.- EXP( - KDIRN * LAI ))
      UVDIRAB = UVDIR * (1.-REFHPD) * (1.- EXP( - KDIR * LAI ))

      PARabs = PARDIFAB + PARDIRAB
      NIRabs = NIRDIFAB + NIRDIRAB
      UVabs = UVDIFAB + UVDIRAB

      RADABS = PARABS + NIRABS + UVABS

      RADABS2 = PARABS + NIRABS

*     Total absorbed flux net radiation
      NETRAD = PARDIFAB + PARDIRAB + NIRDIFAB + NIRDIRAB + UVABS
     &         - HF_CRTOT + HF_PCTOT + HF_SCTOT
      NETRAD2 = PARDIFAB + PARDIRAB + NIRDIFAB + NIRDIRAB
     &         - HF_CRTOT + HF_PCTOT + HF_SCTOT
```

```
      ELSE
*        Net radiation at night
         NETRAD = - HF_CRTOT + HF_PCTOT + HF_SCTOT
         NETRAD2 = - HF_CRTOT + HF_PCTOT + HF_SCTOT
      ENDIF

* ==================================================================

*     PSYCHRometric constant [kPa oC-1]
      PSYCHR = 0.067

*     Volumetric heat capacity of air [J m-3 oC-1]
      RHOCP = 1200.

*     Evaporation energy of 1 mg of water [J mg-1]
      LABDA = 2.5

*     Canopy aerodynamic resistance
      RB_CAN = RB / LAI

*     Canopy conductance at night
      GLEAFD = Gmaxda * EXP( -GNVPDa * VPDair )
      GCAND = GLEAFD * LAI

      IF( LIGHT ) THEN
*       Negative-exponential response to absorbed PAR
        GCAN = AMAX1( 0.0001,
     &                 GCANmax * LAI
     &               * ( 1. -CCN1 * EXP(-CCN2 * PARabs) )
     &               * EXP( -CCN3 * VPDair )
     &               * EXP( -CCN4 * CO2air )
     &         )

*        Rectangular hyperbola for response to absorbed PAR
*        GCANL = AMAX1( 0.0001,
*     &               +  GCANmax * LAI
*     &               * ( PARabs / ( PARabs + CCN2 ) )
*     &               * EXP( -CCN3 * VPDair )
*     &               * EXP( -CCN4 * CO2air )
*     &         )

         GCAN = AMAX1( GCAN, GCAND )

      ELSE

*        Canopy conductance at night
         GCAN = GCAND

      ENDIF

      RS_CAN = 1. / GCAN

*     Resistance for thermal radiation
      RTHRAD = 200. / (1. - EXP( - KDIFBL * LAI ) )

*     Boundary layer Resistance for Heat
      RBH = RB_CAN / 0.93

*     Total Heat Resistance
      RBTH = RTHRAD * RBH / (RTHRAD+RBH)

*     Auxiliary variable [kPa oC-1]
      PSYCHR_ST = PSYCHR * (RB_CAN + RS_CAN) / RBTH

*     Water vapour in air [kPa]
      ES = .6107*EXP(17.4*TEMPair/(TEMPair+239.))
*     To determine slope of ES-curve 1 oC higher
      ES1 = .6107*EXP(17.4*(TEMPair+1.)/(TEMPair+1.+239.))
*     SLOPE of ES-curve [kPa oC-1]
      SLOPE = ES1-ES

      LE = ( SLOPE * NETRAD2 +  RHOCP * VPDair / RBTH )
     &       /  ( SLOPE + PSYCHR_ST )

*     Transpiration in mg water m-2 s-1
      TRANSP = LE / LABDA
```

```
RETURN
END
```

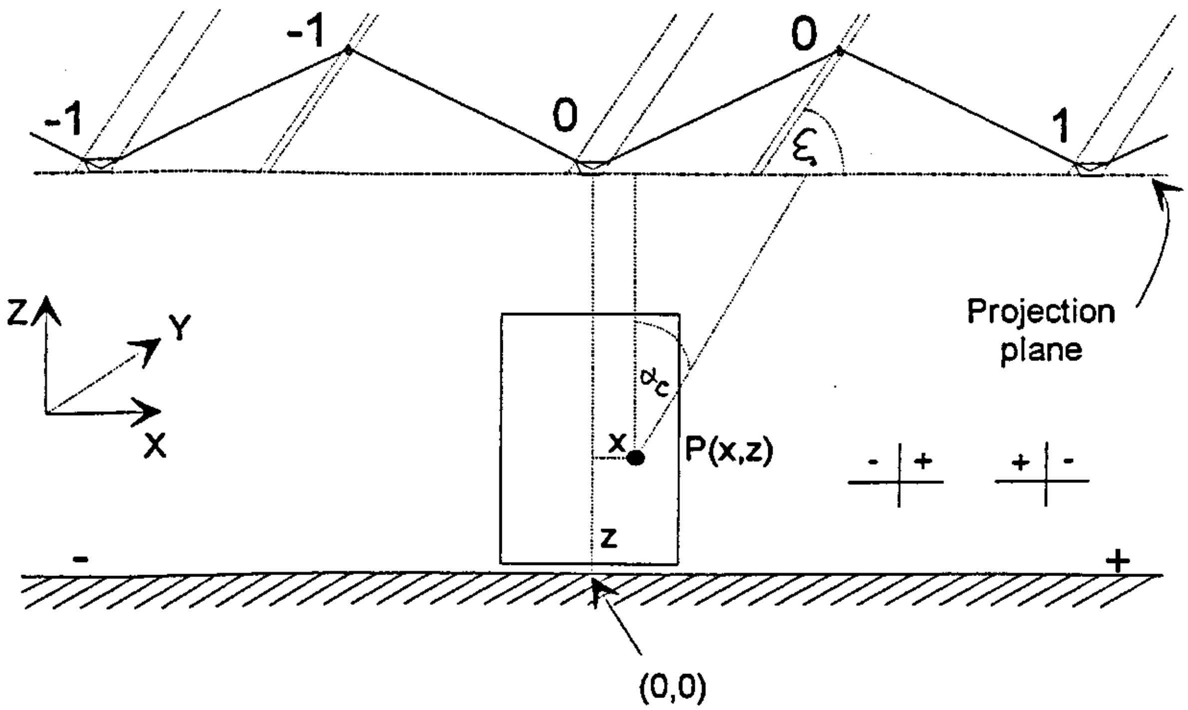
Appendix IX:
Listing of subroutine BIGLPH

```
***********************************************************************
* Subprogram: BIGLPH
* Purpose:
*   Calculate responses of crop photosynthesis to PAR, CO2 and temperature
*   assuming the canopy a big leaf
*
* Description: part of the model of Farquhar et al. (1980)
*  for leaf photosynthesis is extended to the canopy, as described
*  by Evans & Farquhar (1991). The maximal
*  electron transport capacities of individual leaf layers (Jmax) are
*  summed to obtain the canopy-Jmax. No account is taken of Rubisco
*  limited photosynthesis, i.e. it is assumed that photosynthesis is
*  always light-limited. The canopy is divided into a sunlit and
*  shaded part. Absorbed diffuse PAR intensity is evenly
*  spread over the total leaf and absorbed direct PAR over the sunlit
*  part.
*
* Input:
*  PARDIF   : (R4)  incident flux diffuse PAR                [J m-2 s-1]
*  PARDIR   : (R4)  incident flux direct PAR                 [J m-2 s-1]
*  CO2air   : (R4)  CO2 concentration of greenhouse air      [mumol mol-1]
*  TEMPAIR  : (R4)  temperature of greenhouse air            [oC]
*  SINELV   : (R4)  sine of solar elevation                  [-]
*  LAI      : (R4)  Leaf Area Index                          [-]
*  KDIF     : (R4)  extinction coeff. canopy                 [-]
*  JMAX     : (R4)  maximal electron transport rate of canopy
*                                                       [mumol e-2 m-2 s-1]
*  LGHTCON  : (R4)  conversion factor Joule PAR to PAR photons [mumol J-1]
*
* Output:
*  PGROS    : (R4) canopy gross photosynth. CO2 assimilation [mg CO2 m-2 s-1]
*
* Subprograms called:   FARQSIM
*
* Comment: note that calculation of Jmax is valid for temperatures
*           up to about 32 oC
***********************************************************************
      SUBROUTINE BIGLPH( LAI, KDIF, REFGR, PARDIF, PARDIR, SINELV,
     &                   CO2AIR, TEMPAIR, JMAX25, LGHTCON,
     &                   PGROS )
      IMPLICIT REAL( A-Z )

      SQP = SQRT( 0.85 )
      KDIRBL = 0.5 / SINELV
      KDIR = KDIRBL * SQP
      REFHPD = 0.05 / (0.5 + SINELV )

*---- Absorbed radiation

*     Ground reflected PAR
      PARREF = REFGR * ( 0.95 * EXP( - KDIF * LAI ) * PARDIF
     &         + (1.-REFHPD) * EXP( - KDIR * LAI ) * PARDIR )

*     Total diffuse PAR
      PARDIFAB = PARDIF * 0.95 * (1. - EXP(- KDIF * LAI ))
     &           + PARREF * (1. - EXP( - KDIF * LAI ))
*     Total direct PAR (including secondary diffuse)
      PARDIRABT = PARDIR * (1. - REFHPD ) * (1. - EXP( -KDIR * LAI))
*     Direct PAR, not scattered
      PARDIRAB = PARDIR * 0.85 * (1. - EXP( -KDIRBL * LAI))
*     Secondary diffuse
      PARDIRDIFAB = PARDIRABT - PARDIRAB

*     Partitioning big leaf into sunlit and shaded leaf area
*     Note that LAISUN and LAISH are fractions
      LAISUN = 1. / KDIRBL * (1. - EXP( -KDIRBL * LAI ))
      LAISH =  1. - LAISUN
```

```
*     Averaging of absorbed fluxes over leaf area
      PARSH = PARDIFAB + PARDIRDIFAB
      PARSUN = PARSH + 0.85 * KDIRBL * PARDIR

*---- Photosynthesis
*     Assumption CO2 concentration in chloroplasts is 0.67 times
*     CO2 concentration in ambient air (Evans & Farquhar, 1991)
      CO2C = 0.67 * CO2AIR

*     Temperature effect on CO2 compensation point
*     approximation of relation given by Brooks & Farquhar (1985)
      GAMMA = 1.7 * TEMPAIR

*     Temperature effect on maximal electron transport rate
*     approximation of optimum response for temperatures below 30 oC
      JMAX = JMAX25 * TEMPAIR / 25.

*     Photosynthesis of shaded and sunlit leaf area (mumol m-2 s-1)
      PHOTSH  = FARQSIM( PARSH * LGHTCON, CO2C, GAMMA, JMAX )
      PHOTSUN = FARQSIM( PARSUN * LGHTCON, CO2C, GAMMA, JMAX )

*     Canopy photosynthesis (mg CO2 m-2 s-1)
      PGROS = (PHOTSH * LAISH + PHOTSUN * LAISUN) * 0.044

      RETURN
      END

*************************************************************************
* Subprogram: FARQSIM
* Purpose:
*  calculation of leaf gross photosynthesis according to the
*  model of Farquhar et al. (1980) assuming photosynthesis
*  to be limited by regeneration of RuBP
*
* Input:
*  PARABS_M : (R4) absorbed PAR                              [mumol m-2 s-1]
*  CO2C     : (R4) CO2 concentration in chloroplasts         [mumol mol-1]
*  GAMMA    : (R4) CO2 compensation point in absence of
*                photorespiration                           [mumol mol-1]
*  JMAX     : (R4) maximal rate of electron transport      [mumol e- m-2 s-1]
*
* Output:
*  FARQSIM  : (R4) leaf gross photosynth. CO2 assimilation [mumol m-2 s-1]
*************************************************************************
      REAL FUNCTION FARQSIM( PARABS_M, CO2C, GAMMA, JMAX )
      IMPLICIT REAL(A-Z)

*     Curvature factor PAR response of electron transport
      THETA = .7

*     Fraction of PAR absorbed by non-photosynthetic tissues
      F = .3

*---- Electron transport rate
*       (2 electrons per absorbed photon)
      EFFRAD = PARABS_M / 2. * (1.-F)
      J = ( JMAX + EFFRAD -
     &     SQRT( (JMAX+EFFRAD)**2-4.*THETA*EFFRAD*JMAX )  )
     &     / (2. * THETA)

*     Electron transport/photophosphorylation limited rate of
*     RuP2 regeneration
      VC = J / 4. * (CO2C-GAMMA) / (CO2C + 2. * GAMMA )

      FARQSIM = VC

      RETURN
      END
*
```

Appendix X:

Listing of the points-model for greenhouse cover shading of a row crop.



igure X.1 Scheme of the calculation of cover shading on a row crop. Shades of the greenhouse cover are projected onto the projection plane. By projection of P on the projection plane it is determined whether any point P in the XZ-plane is recieving shade.
Here a north-south row and greenhouse is depicted, with view direction to the north.
Angle ξ is the angle of the solar beam in the XZ-plane with horizontal (negative for sun in eastern hemisphere). Angle α_c is angle of solar beam in the XZ-plane with vertical (positive for sun in eastern hemisphere).
X-coordinates to the right of the origin have a positive sign, to the left a negative sign. Ridges and gutters are numbered relative to the origin.

```
************************************************************************
* Subprogram: ROWPNT
*
* Purpose:
*  calculation of average fraction sunlit leaf area and absorbed direct
*  PAR at several horizontal layers in a rectangular section
*  of a row crop
*
*  Description:
*  At each point in a horizontal layer considered the fraction sunlit
*  leaf area (FSL) and the intensity of absorbed direct PAR (IDTOT)
*  are calculated according to the row model of Gijzen & Goudriaan (1989).
*  For each point it was also calculated whether is was shaded by
*  a ridge or a gutter (TRRID = 0 or 1, TRGUT = 0 or 1). Average fraction
*  sunlit leaf area and absorbed direct radiation of the horizontal layer
*  are obtained by averaging FSL * TRRID * TRGUT and IDTOT * TRRID * TRGUT
*  for all the points in the layer (see Eqns. 3 and 4 in Appendix II of
*  Gijzen (1994). 5 layers for 5-point Gaussian integration or 40 layers
*  for normal averaging may be considered.
*
* Input:
*   ANDIS   : leaf angle distribution (1: spherical, 2: horizontal,
*              3: planophile, 4: near-planophile)                [-]
*   SQP     : scattering factor (= SQRT(1-SCP) )                  [-]
*   LAD     : leaf area density                                 [m2 m-3]
*   ELEVN   : elevation of sun                                 [radians]
*   AZIMS   : azimuth of sun                                   [radians]
*   AZIMR   : azimuth crop row                                 [radians]
*   WIDTH   : width of crop row (size in X-direction)             [m]
*   PATH    : width of path between rows                          [m]
*   HEIGHT  : height of crop row (size in Z-direction)            [m]
*   XOFFW   : distance between first gutter at left side
*             and left side crop row                             [m]
*   ZOFFW   : distance between ground
*             and under side of crop row (= VOET)                [m]
*   HROW    : option F : whole width of row is considered
*             T : half width of row is considered                [-]
*   ROW_SIDE : 0 : left half of row, 1 : right half              [-]
*
*  Output:
*   ROWSLTB : table with average fractions sunlit leaf area
*              for horizontal layers                             [-]
*   ROWPATB : table with average intensities of absorbed
*              direct PAR for horizontal layers                  [-]
*
* Subprograms called:  BLKCOOR, DIRN2, TWODTR
*
************************************************************************
      SUBROUTINE ROWPNT( ANDIS, AZIMGR, ELEVN, AZIMS, ALPHAC,
     &          SUMSL, CH_SUMSL )
      IMPLICIT REAL (A-Z)

      INTEGER J
      INTEGER IX, NX, IZ, NZ,I1

*     number of horizontal points in rectangle, and number of horizontal
*     layers for Gaussian integration
      INTEGER NGRIDX, NGRIDZ
      PARAMETER (NGRIDX = 40, NGRIDZ = 5 )

      DIMENSION SUMSL ( NGRIDX, NGRIDZ )
      CHARACTER*1 CH_SUMSL ( NGRIDX, NGRIDZ )

      LOGICAL INI
      INTEGER ANDIS

      COMMON /GENCOM2/ SOLHR

*     YOFFW is dummy variable
      COMMON /GRGEOM1/ XOFFW, YOFFW, ZOFFW

      COMMON /ROWCHAR1/ AZIMR, WIDTH, HEIGHT, PATH, VOET
      COMMON /ROWCHAR2/ LAD, SQP

      LOGICAL HROW
      COMMON /ROWCHAR3/ HROW
```

```
*     Gaussian weights are not used
      DIMENSION XGAUSS5( 5 ), WGAUSS5( 5 )

      INTEGER ROW_PLACE, ROW_SIDE
      COMMON /ROWCHAR5/ ROW_PLACE, ROW_SIDE

      DIMENSION ROWSLTB( NGRIDZ )
      DIMENSION ROWPATB( NGRIDZ )
      COMMON /ROWOUT2/ ROWSLTB, ROWPATB

      SAVE

      DATA XGAUSS5 /0.0469101,0.2307653,0.5000000,
     &              0.7692347,0.9530900  /

      PI = 3.1415926
      RADN = PI / 180.

      NZ = NGRIDZ
      NX = NGRIDX

*     If half of row, than take half of width of row
      IF( HROW ) THEN
         TWIDTH = 0.5 * WIDTH
         IF( ROW_SIDE .EQ. 0) THEN
*           left half side of row
            P0 = 0.
         ELSE
*           right half side of row
            P0 = 0.5 * WIDTH
         ENDIF
      ELSE
*        Take whole width of row
         TWIDTH = WIDTH
         P0 = 0.
      ENDIF

      DIFAZIM = AZIMS - AZIMGR
      DIFAZIM = AMOD( DIFAZIM, PI )

*     Conversion of coordinates of light beam direction
*     ALPHAC is angle with vertical in XZ-plane
*     BETAC is angle with YZ-plane
*     ALPHAC is used here for compatibility with row model
*     of Gijzen & Goudriaan (1989)
*     angle KSI in model of Bot (1983) is complement of ALPHAC,
*     but has different sign
*     ALPHAC is for north-south oriented row negative when coming
*     from eastern hemisphere
      CALL BLKCOOR( AZIMR, AZIMS, ELEVN, ALPHAC, BETAC )
      SINELV = SIN( ELEVN )

      INI = .TRUE.
      CALL TWODTR( INI, DIFAZIM, ELEVN,
     &      XPOS, ZPOS,
     &      TRID, TGUT )
      INI = .FALSE.

      DO 40 IZ = 1, NZ
         PZ = XGAUSS5( IZ ) * HEIGHT

         TSLLA = 0.
         TPARDIRABS = 0.
         DO 30 IX = 1, NX
            PX = P0 + ( FLOAT(IX) - 0.5)/FLOAT(NX) * TWIDTH

            XPOS = PX + XOFFW

*           PZ is reckoned from top of row
            ZPOS = HEIGHT - PZ + ZOFFW

*           Convert sign of ALPHAC to switch left and right sides of row
*           Fraction sunlit leaf area, and absorbed direct light intensity
            CALL DIRN2( WIDTH, PATH, LAD, SQP, ANDIS,
     &         PX, PZ, ALPHAC, BETAC, IDTOT, FSL )
```

```
*----       Calculate whether given point is receiving shadow
*           from construction
            CALL TWODTR( INI, DIFAZIM, ELEVN,
     &      XPOS, ZPOS,
     &      TRID, TGUT )

            TSLLA = TSLLA + TRID * TGUT * FSL
            TPARDIRABS = TPARDIRABS + IDTOT * TGUT * TRID

            SUMSL( IX, IZ ) = TRID * TGUT
            IF( TRID * TGUT .EQ. 0 ) THEN
               CH_SUMSL(IX, IZ) = 'X'
            ELSE
               CH_SUMSL(IX, IZ) = ':'
*               CH_SUMSL(IX, IZ) = CHAR(250)
            ENDIF
*            SUMSL( IX, IZ ) = TGUT

30       CONTINUE

         ROWSLTB( IZ ) = TSLLA / FLOAT(NX)
         ROWPATB( IZ ) = TPARDIRABS / FLOAT( NX )

40    CONTINUE

      RETURN
      END
*********************************************************************
* SUBPROGRAM: TWODTR
*
* Purpose: test whether given point in XZ-plane (vertical plane
*  perpendicular to ridge-gutter system) is receiving shade from
*  either gutter or ridge
*
* Description:
*  The vertical direction is Z-direction, and the axis running parallel
*  to the beams and perpendicular to ridges and gutters is the X-direction.
*  Point (0.0) is the projection of the center of the gutter nearest
*  to the left onto the ground.
*  Construction elements are projected from above onto a projection plane
*  (with  height equal to the underside of the lowest element
*  (in this case the underside of the gutters), into the direction
*  of the sun. Then a given point under the projection
*  plane is projected onto the projection plane (into the direction of
*  sun), and it is tested whether the projection is hitting a shade
*  of a construction element.
*
* Control variables: INI
*
* Input:
*   DIFAZIM: difference azimuths greenhouse and sun           [radians]
*   ELEVN  : elevation of sun                                  [radians]
*   XPOS   : horizontal position point in XZ-plane                 [m]
*   ZPOS   : vertical position point in XZ-plane                   [m]
*   HLORID : height of lower half ridge                            [m]
*   HUPRID : height of upper half ridge                            [m]
*   WLORID : width of lower side ridge                             [m]
*   WUPRID : width of upper side ridge                             [m]
*   HRID   : height of ridge                                       [m]
*   SLOPE_SIDE_RID : angle side of ridge with horizontal      [radians]
*   HLOGUT : height of lower half gutter                           [m]
*   HUPGUT : height of upper half gutter                           [m]
*   WLOGUT : width of lower side gutter                            [m]
*   WUPGUT : width of upper side gutter                            [m]
*   HGUT   : height of gutter                                      [m]
*   ZPOS_GUT: height of underside gutter above ground              [m]
*   SLOPE_SIDE_GUT : angle side of gutter with horizontal      [radians]
*   SLOPE  : slope of glass                                    [radians]
*   SPANW  : distance between gutters                              [m]
*
* Output:
*   TRID   : transmissivity of ridges for direct radiation          [-]
*   TGUT   : transmissivity of gutters for direct radiation         [-]
*
* COMMENT:
*- Initialization must be done at each different solar position
*- Note that middle of shades are coinciding with centre of construction
```

```
*  elements, thus not the exact position is calculated
*- Transsection ridges is represented as trapezium with short side upwards.
*  Transsection gutters is represented as trapezium with long side upwards.
*  Transsection beams is represented as rectangle.
*- Some geometrical calculation are derived from the model of Bot (1983)
*  Equation numbers given refer to Bot (1983)
*********************************************************************
      SUBROUTINE TWODTR( INI, DIFAZIM, ELEVN,
     &      XPOS, ZPOS,
     &      TRID, TGUT )
      IMPLICIT REAL (A-Z)

      INTEGER I, IL, IR, NG, NR
      INTEGER IRID, IGUT

      COMMON /GEN/ SOLHR

      LOGICAL INI

      INTEGER NGUTL, NGUTR, NGL, NGR
      INTEGER NRIDL, NRIDR, NRL, NRR

      PARAMETER (NGUTL = -5, NGUTR = 5)
      PARAMETER (NRIDL = -5, NRIDR = 5)

* Place of ridges in horizontal projection plane, maximal 3 ridges
      DIMENSION XOFFRID( NRIDL:NRIDR ), ZOFFRID( NRIDL:NRIDR )
      DIMENSION DRL(NRIDL:NRIDR), DRR(NRIDL:NRIDR)

* Place of gutters in horizontal projection plane, maximal 2 gutters
      DIMENSION XOFFGUT(NGUTL:NGUTR), ZOFFGUT(NGUTL:NGUTR)
      DIMENSION DGL(NGUTL:NGUTR), DGR(NGUTL:NGUTR)

* Output tables
      DIMENSION RTB( 2, NRIDL:NRIDR )
      DIMENSION GTB( 2, NGUTL:NGUTR )
      COMMON /ROWOUT1/ RTB, GTB

      COMMON /CONSTR1/ HLOGUT, HUPGUT, WLOGUT, WUPGUT, HGUT, ZPOS_GUT,
     &                 SLOPE_SIDE_GUT
      COMMON /CONSTR2/ HLORID, HUPRID, WLORID, WUPRID, HRID, ZPOS_RID,
     &                 SLOPE_SIDE_RID
      COMMON /CONSTR3/ SLOPE, SPANW

      COMMON /CONSTR4/ KSI, LRID, LGUT, GUTSHADE, RIDSHADE

      INTEGER ROW_PLACE, ROW_SIDE
      COMMON /ROWCHAR5/ ROW_PLACE, ROW_SIDE

      SAVE

      PI = 3.1415926

      IF( INI ) THEN

         SINELV = SIN (ELEVN )
         COSELV = COS( ELEVN )
         J7 = COSELV * SIN( DIFAZIM )

* Angle KSI in YZ-plane between incident ray and Y-axis     (5.26)
         IF (J7 .EQ. 0.)  THEN
            KSI = .5*PI
         ELSE
            KSI = ATAN ( SINELV /J7 )
         ENDIF

* Gutter
*                                                              (5.43a)
        IF (ABS(KSI) .GE. SLOPE_SIDE_GUT) THEN
           LGUT = HGUT*COS(KSI) + WUPGUT*ABS(SIN(KSI))
        ELSE
           LGUT = HGUT*COS(KSI) + 0.5*(WLOGUT+WUPGUT)*ABS(SIN(KSI))
        ENDIF
        GUTSHADE = LGUT / ABS(SIN( KSI ))

* Ridge
*                                                             (5.43b)
```

```
         IF( ABS(KSI) .GE. SLOPE_SIDE_RID ) THEN
            LRID = HRID*COS(KSI) + WLORID*ABS(SIN(KSI))
         ELSE
            LRID = HRID*COS(KSI)+ 0.5*(WLORID+WUPRID)*ABS(SIN(KSI))
         ENDIF
         RIDSHADE = LRID / ABS(SIN(KSI))

*        Projection plane : make height equal to height of underside of gutters
         REFHEIGHT = ZPOS_GUT

*        Indices left and right side of projected elements
         IL = 1
         IR = 2

*        Range of gutters and ridges; numbers 0 are the ridges and gutters
*        closest (left side) to the stand
         IF( ROW_PLACE .EQ. 0 ) THEN
*           Under gutter
            NGL = -4
            NGR = 4
            NRL = -4
            NRR = 3
         ELSEIF( ROW_PLACE .EQ. 1 ) THEN
*           Under ridge
            NGL = -3
            NGR = 3
            NRL = -3
            NRR = 4
         ENDIF

*         Gutters
          DO IGUT = NGL, NGR
             XOFFGUT( IGUT ) = FLOAT( IGUT ) * SPANW
*            Height above projection plane
             ZOFFGUT( IGUT ) = 0.
*            Places of projections in projection plane
*            distances are subtracted because elements are located
*            above projection plane
             DISPGUT = XOFFGUT(IGUT) + ZOFFGUT(IGUT) / TAN( KSI )
*            Places of beginning and end of shades (left and right side)
             DGL(IGUT) = DISPGUT - 0.5 * GUTSHADE
             DGR(IGUT) = DISPGUT + 0.5 * GUTSHADE
             GTB(IL, IGUT) = DGL(IGUT)
             GTB(IR, IGUT) = DGR(IGUT)
          END DO

*         Ridges
          DO IRID = NRL, NRR
             XOFFRID( IRID ) = (FLOAT(IRID)+0.5) * SPANW
*            Height above projection plane
             ZOFFRID( IRID ) = 0.5 * SPANW * TAN( SLOPE )
*            Places of projections in projection plane
*            distances are subtracted because elements are located
*            above projection plane
             DISPRID = XOFFRID(IRID) + ZOFFRID(IRID) / TAN( KSI )
*            Places of beginning and end of shades (left and right side)
             DRL(IRID) = DISPRID - 0.5 * RIDSHADE
             DRR(IRID) = DISPRID + 0.5 * RIDSHADE
             RTB(IL,IRID) = DRL(IRID)
             RTB(IR,IRID) = DRR(IRID)
          ENDDO

          TANKSI = TAN( KSI )

      ENDIF
*-----end INI

      TRID = 1.
      TGUT = 1.

*     Project point in greenhouse on projection plane
      XPOS_PP = XPOS - (REFHEIGHT - ZPOS) / TANKSI

*     Test whether projection of point is falling on RIDGE
      DO IRID = NRL, NRR
        IF( XPOS_PP .GT. RTB(IL,IRID)
```

```
     & .AND. XPOS_PP .LT. RTB(IR,IRID)) THEN
            TRID = 0.
            GOTO 15
         END IF
      END DO
5     CONTINUE

*     Test whether projection of point is falling on GUTTER
      DO IGUT = NGL, NGR
         IF( XPOS_PP .GT. GTB(IL,IGUT)
     &   .AND. XPOS_PP .LT. GTB(IR,IGUT)) THEN
            TGUT = 0.
            GOTO 15
         END IF
      END DO
15    CONTINUE

      RETURN
      END
************************************************************************
* SUBPROGRAM: DIRN2
* Type: SUBROUTINE
* Date: 1-12-1993
* Author: H. Gijzen
* Purpose:
*   Calculation of the average fraction sunlit leaf area and relative
*   light intensity for direct light at a point (Z,W) in a row.
*   Direction of light is according to converted coordinates (Goudriaan, 1977)
*   The leaf area is homogeneously distributed in the row.
*
*  Input:
*      WIDTH   : (R4)   width of row                              [m]
*      PATH    : (R4)   width of path                             [m]
*      LAD     : (R4)   Leaf Area Density of row                  [m2 m-3]
*      SQP     : (R4)   square root of scattering coefficient     [-]
*      ANDIS   : (I4)   leaf angle distribution : 1 = spherical,
*                       2 = cucumber, 3 = horizontal              [-]
*      W       : (R4)   distance of point to left side of row     [m]
*      Z       : (R4)   distance of point to top of row           [-]
*      ALPHAC  : (R4)   converted azimuth                         [radians]
*      BETAC   : (R4)   converted inclination                     [radians]
* Output:
*      FRSUNL  : (R4)   fraction sunlit leaf area                   [-]
*      INTH    : (R4)   relative light intensity on horizontal plane [-]
*
* FATAL ERROR CHECKS: no
*
* WARNINGS: no
*
* SUBPROGRAMS CALLED: no
*
* FILE USAGE: no
*
* COMMENT:
*   Takes account of left or right side of row
************************************************************************
      SUBROUTINE DIRN2( WIDTH,PATH,LAD,SQP,ANDIS,W,Z,ALPHAC,BETAC,
     &                  INTH, FRSUNL )
      IMPLICIT REAL (A-Z)
      INTEGER ANDIS,NUNIT

* Sine of solar elevation
      SINELV = COS(BETAC)*COS(ALPHAC)
* Solar altitude smaller than 3 degrees, rel. light intensity and sunlit
* leaf area are set to zero
      IF( SINELV .LT. 0.0524 ) THEN
            INTH = 0.
            FRSUNL = 0.
            RETURN
      ENDIF

      IF( ALPHAC .LT. 0. ) THEN
         WN = WIDTH - W
         ALPHC = - ALPHAC
      ELSE
         ALPHC = ALPHAC
         WN = W
```

```
      ENDIF

      TANA = TAN( ALPHC )
      COSB = COS( BETAC )
      CALL ATNRAD( SINELV, ANDIS, SQP, OAV, REFD )

* Number of units (WIDTH+PATH) traversed by light, calculated from total
* horizontal pathlenght through canopy (TOTHOR)
      UNIT = WIDTH+PATH
      TOTHOR = Z*TANA

      REST = AMOD( TOTHOR+WN, UNIT )
      NUNIT = INT( (TOTHOR+WN-REST) / UNIT )
      IF (NUNIT .GE. 4) THEN
            LT = (Z * LAD) / COS(ALPHC) * WIDTH/UNIT
      ELSE
            CALL PATLEN( WIDTH,PATH,LAD,WN,Z,ALPHC,LT )
      ENDIF

      FRSUNL = EXP(-OAV*LT/COSB)

* INTH not yet corrected for reflection
      INTH = EXP(-OAV*LT/COSB*SQP)

      RETURN
      END
**************************************************************************
* SUBPROGRAM: BLKCOOR
* Type: SUBROUTINE
* Date: 20-4-1990
* Author: H. Gijzen
* Purpose:
*   Calculation of converted azimuth and converted inclination according to
*   Goudriaan (1977, p. 55)
*
* Input:
*    AZIMR   : (R4)  azimuth row                                [radians]
*    AZIMS   : (R4)  azimuth sun                                [radians]
*    BETA    : (R4)  solar elevation                            [radians]
* Output:
*    ALPHAC  : (R4)  converted azimuth                          [radians]
*    BETAC   : (R4)  converted inclination                      [radians]
*
* SUBPROGRAMS CALLED: no
*
* Comment:
*   XY-plane              :  vertical plane perpendicular to row direction
*   Converted azimuth     :  angle with vertical in yz-plane
*   Converted inclination : angle with xz-plane
**************************************************************************
      SUBROUTINE BLKCOOR( AZIMBL, AZIMS, BETA, ALPHAC, BETAC )

      PI = 3.1415926

      AZIMD = AZIMBL - AZIMS

*     New coordinates
*     sin (BETA) = cos (ALPHAC) * cos (BETAC)
*     sin (BETAC) = cos (AZIMD) * cos (BETA)

*     For AZIMBL = 0 then for northern hemisphere BETAC negative
      BETAC = ASIN( COS(AZIMD)*COS(BETA) )
      COSAC = AMIN1( 1., SIN(BETA)/COS(BETAC) )
      ALPHAC = ACOS( COSAC )

*     Signs are added to be able to determine whether beam direction is from
*     east, west, north or south hemisphere for block with azimuth at 0.
*     east hemisphere : AZIMD negative, ALPHAC positive
*     west hemisphere : AZIMD positive, ALPHAC negative
      IF( AZIMD .LT. 0. ) THEN
         ALPHAC = - ALPHAC
      ENDIF

      RETURN
      END
**************************************************************************
```

```
* SUBPROGRAM: PATLEN
* Type: SUBROUTINE
* Date: 01-06-1987
* Author: H. Gijzen
* Purpose:
*   Computation of leaf area traversed by light beam through row crop with
*   uniform leaf density.
*
* Input:
*    WIDTH   : (R4)   width of row                                   [m]
*    PATH    : (R4)   width of path                                  [m]
*    LAD     : (R4)   leaf area density                              [-]
*    W       : (R4)   Distance from left side of row                 [m]
*    Z       : (R4)   Distance to top of row                         [m]
*    ALPHAC  : (R4)   Angle of beam with verical                     [radians]
* Output:
*    LT      : (R4)   Leaf Area traversed by beam                    [-]
*
* FATAL ERROR CHECKS: no
*
* WARNINGS: no
*
* SUBPROGRAMS CALLED: none
*
* FILE USAGE: NONE
*
* COMMENT: path width should be higher than 0.0
* ***********************************************************************
      SUBROUTINE PATLEN (WIDTH,PATH,LAD,W,Z,ALPHAC,LT)
      IMPLICIT REAL(A-Z)

      UNIT = WIDTH+PATH
      TANA = TAN(ALPHAC)
      IF(Z.LT..00001) THEN
            LT = 0.
            RETURN
      ENDIF
* Light beam vertical
      IF((ABS(ALPHAC)).LT.0.00001) THEN
            P = Z
            GOTO 800
      ENDIF

* Number of units (WIDTH+PATH) traversed by beam (NUNIT), calculated from
* total horizontal pathlength through canopy (TOTHOR)
      TOTHOR = Z*TANA
      REST = AMOD(TOTHOR+W,UNIT)
      NUNIT = INT ((TOTHOR+W-REST)/UNIT)
      IF (REST .LE. WIDTH) THEN
            P = ( REST + (NUNIT)*WIDTH -W)/SIN(ALPHAC)
      ELSE
            P = ( (NUNIT+1)*WIDTH - W)/SIN(ALPHAC)
      ENDIF

800   CONTINUE
      LT = P * LAD
      RETURN
      END
***********************************************************************
* SUBPROGRAM: ATNRAD
* TYPE: SUBROUTINE
* Date: 28-AUG-1992
* PURPOSE: Calculation of projection and reflection coefficient of leaves
*   with given leaf angle distribution
*   for a beam with sine of angle with horizontal SINELV
*
* Input:
*   SINELV  :  (R4) sine of angle of beam with horizontal              [-]
*   ANDIS   :  (I4) index for leaf angle distribution
*                 1 = spherical , 2 = horizontal leaves,
*                 3 = near-planophile, 4 = planophile, 5 = cucumber [-]
*   SQP     : (R4) scattering factor for PAR                         [-]
* Output:
*   OAV     : (R4) average projection of leaves into direction of beam [-]
*   REFL    : (r4) reflection coefficient of leaves                    [-]
*
* Comment: formula based on Goudriaan (1988)
```

```
***************************************************************************
      SUBROUTINE ATNRAD( SINELV, ANDIS, SQP, OAV, REFL )
           IMPLICIT REAL(A-Z)
           INTEGER ANDIS
           DIMENSION FEQ( 3,5 )

* Distribution of leaf angles in classes 0-30 (1), 30-60 (2) and 60-90 (3)
* degrees
* ANDIS = 1 : spherical distribution
* ANDIS = 2 : horizontal leaves
* ANDIS = 3 : near-planophile
* ANDIS = 4 : planophile
* ANDIS = 5 : measured distribution cucumber (Proefstation Naaldwijk)
           DATA FEQ /
     &   .134, .366, .5,
     &    1., 0., 0.,
     &    0.37, 0.42, 0.21,
     &    0.615, 0.318, 0.067,
     &    0.3996, .3639, .2365/

      IF( ANDIS .EQ. 1 ) THEN
         OAV = .5
      ELSEIF( ANDIS .EQ. 2 ) THEN
                OAV = SINELV
           ELSE
         F1 = FEQ( 1, ANDIS )
         F2 = FEQ( 2, ANDIS )
         F3 = FEQ( 3, ANDIS )
* Mean values of projection for leaf inclination classes around 15, 45
* and 75 degrees, dependent on angle of beam with horizontal
         O15 = AMAX1( 0.26, 0.93*SINELV )
         O45 = AMAX1( 0.47, 0.68*SINELV )
              O75 = 1. - 0.268*O15 - 0.732*O45
* Average projection black leaves
              OAV = F1*O15 + F2*O45 + F3*O75
      ENDIF

      IF( SQP .GT. 0.99 ) THEN
                 REFL = 0.
           ELSE
* Reflection coefficient horizontal leaves
         REFL = (1. - SQP) / (1. + SQP)
* Refl. coeff. spherical leaf angle distribution
         IF( ANDIS .EQ. 1 ) THEN
                          REFL = REFL * 2. / (1. + 2. * SINELV)
         ELSEIF( ANDIS .GT. 2 ) THEN
            REFL = REFL * OAV * 2. / ( OAV + KDIFBL*SINELV )
                 ENDIF
           ENDIF

           RETURN
           END
*
```

Appendix XI:

Listing of the area-model for greenhouse cover shading of a plant stand

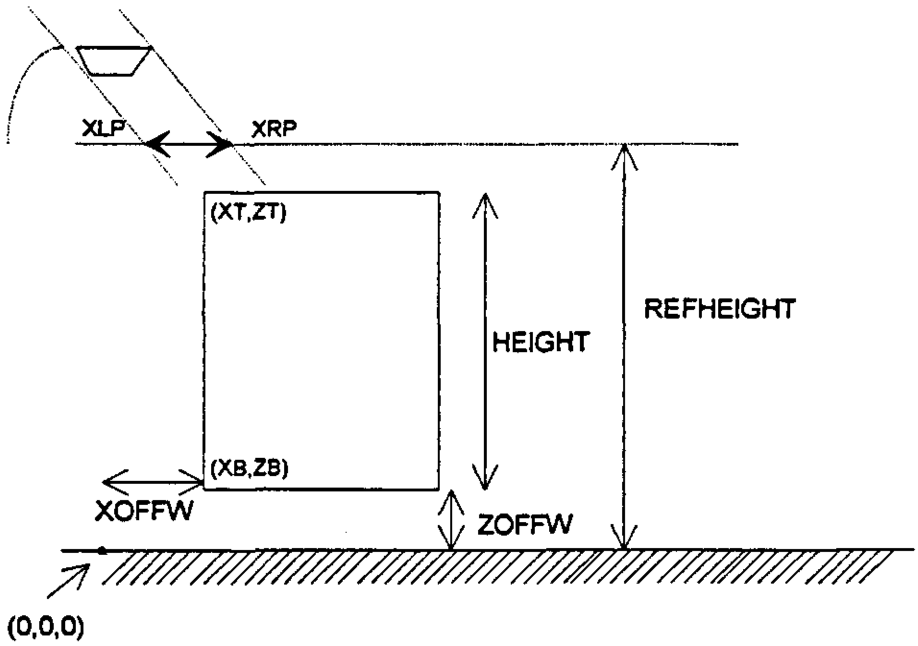


Figure XI.1 Scheme for calculation of the projection of the projected width of a gutter onto a crop stand (XZ-plane) (SUBROUTINE ROWAREA). Direction of view is to the north, angle ξ is negative for angles to the left of the vertical. The coordinates of the projected width of the gutter are XPL for the left side, and XRP for the right side. The crop stand has a distance ZOFFW to the ground, a distance XOFFW to the centre of the nearest gutter at the left side.
This scheme is also used for the YZ-plane, for projection of the beam system. Then the direction of view is to the east, angle ξ is replaced by angle ε, and the crop stand has a distance YOFFW to the centre of the nearest beam at the left side.

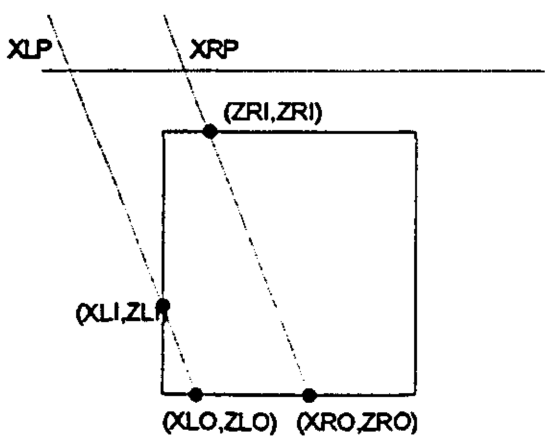


Figure XI.2 Naming of points of entry and exit of left and right side of the shade cast by a given cover element.

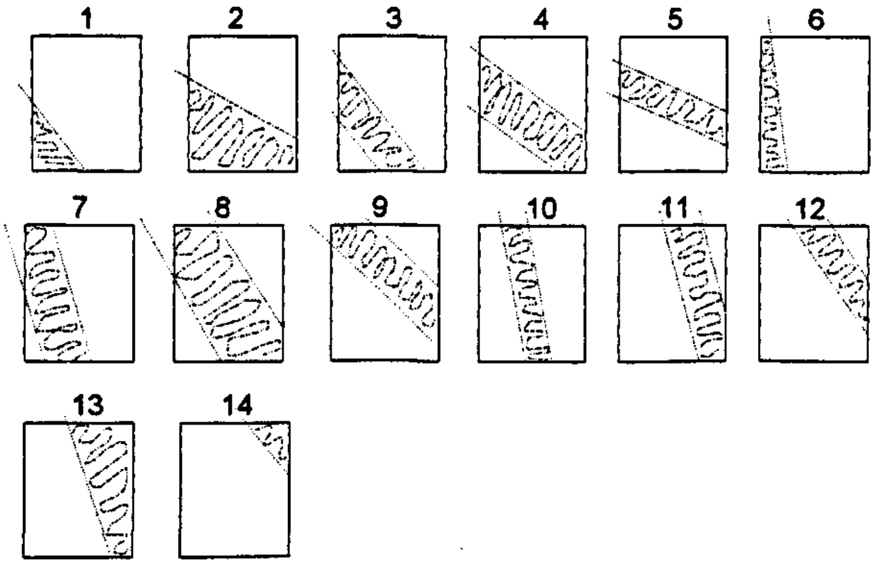


Figure XI.3 Numbering of possible shapes of shade area's, as applied in SUBROUTINE SHAD3.

```
************************************************************************
* SUBPROGRAM: ROWAREA
*
* Purpose:
*  calculation of the amount of shade that a crop stand is receiving
*  from construction elements ridges, gutters and beam
*
* Description:
*  The 3-dimensional position of a crop stand with respect to the
*  construction elements of a greenhouse is calculated. The vertical
*  direction is Z-direction, the axis running parallel to ridges and
*  gutters is Y-direction, and the axis running parallel to the beams
*  is the X-direction.
*  Construction elements are projected onto a projection plane with
*  height above ground REFHEIGHT. (This can be taken equal to the
*  underside of the lowest element (e.g. equal the underside of the beams)
*  Then the projected shades are projected onto the crop stand.
*  The crop stand is represented as a block with height (Z-direction)
*  width (X-direction) and depth (Y-direction). For the ridge-gutter
*  system the block is transsected in the XZ-plane, and the area of the
*  shade cast on the rectangle is calculated. The transmission of the
*  ridge-gutter system is than
*  T_ridgut = 1 - area_shade_xz / area_rectangle_xz.
*  This same is done for the beams, i.e. the block is transsected in the
*  YZ-plane.
*  T_beam = 1 - area_shade_yz / area_rectangle_yz.
*  Total transmission is T_ridgut * T_beam
*
*  Control variable: INI
*
*  Input:
*   DIFAZIM: difference azimuths greenhouse and sun          [radians]
*   ELEVN  : elevation of sun                                [radians]
*   HLORID : height of lower half ridge                          [m]
*   HUPRID : height of upper half ridge                          [m]
*   WLORID : width of lower side ridge                           [m]
*   WUPRID : width of upper side ridge                           [m]
*   HRID   : height of ridge                                     [m]
*   HLOGUT : height of lower half gutter                         [m]
*   HUPGUT : height of upper half gutter                         [m]
*   WLOGUT : width of lower side gutter                          [m]
*   WUPGUT : width of upper side gutter                          [m]
*   HGUT   : height of gutter                                    [m]
*   ZPOS_GUT: height of underside gutter above ground            [m]
*   SLOPE  : slope of glas`s                                  [radians]
*   SPANW  : distance between gutters                            [m]
*   WBEAM  : width of beam                                       [m]
*   HBEAM  : height of beam                                      [m]
*   BEAMDIS: distance between beams                              [m]
*   ZPOS_BEAM: height of underside beam above ground             [m]
```

```
*   REFHEIGHT: reference height (=height of projection plane)      [m]
*   AZIMR   : azimuth crop stand (angle direction of depth)     [radians]
*   WIDTH : width of crop stand (size in X-direction)               [m]
*   HEIGHT: height of crop stand (size in Z-direction)              [m]
*   DEPTH : depth of crop stand (size in Y-direction)               [m]
*   XOFFW : distance between first gutter at left side
*            and left side crop stand                               [m]
*   YOFFW : distance between first beam at front side
*            and front side crop stand                              [m]
*   ZOFFW : distance between ground
*            and under side of crop stand (=VOET)                   [m]
*
* Output:
*
*   TRID   : transmissivity of ridges for direct radiation           [-]
*   TGUT   : transmissivity of gutters for direct radiation          [-]
*   TBEAM  : transmissivity of beams for direct radiation            [-]
*
* Subprograms called: SHAD3
*
* Comment:
*  Note that middle of shades are coinciding with centre of construction
*  elements, thus not the exact position is calculated
*  Transsection ridges is represented as trapezium with short side upwards.
*  Transsection gutters is represented as trapezium with long side upwards.
*  Transsection beams is represented as rectangle.
*  Some geometrical calculation are derived from the model of Bot (1983)
*  Equation numbers given refer to Bot (1983)
*******************************************************************
      SUBROUTINE ROWAREA( INI, DIFAZIM, ELEVN,
     &      TRID, TGUT, TBEAM
     & )
      IMPLICIT REAL (A-Z)
      LOGICAL INI

      INTEGER I, IL, IR, NG, NR, M
      INTEGER IRID, IGUT, IBEAM

      INTEGER NGUTL, NGUTR, NGL, NGR
      INTEGER NRIDL, NRIDR, NRL, NRR
      INTEGER NBEAML, NBEAMR, NBL, NBR

      PARAMETER (NGUTL = -5, NGUTR = 5)
      PARAMETER (NRIDL = -5, NRIDR = 5)
      PARAMETER (NBEAML = -5, NBEAMR = 5)

*     Place of ridges in horizontal projection plane
      DIMENSION XOFFRID( NRIDL:NRIDR )
      DIMENSION DRL(NRIDL:NRIDR), DRR(NRIDL:NRIDR)

*     Place of gutters in horizontal projection plane
      DIMENSION XOFFGUT(NGUTL:NGUTR)
      DIMENSION DGL(NGUTL:NGUTR), DGR(NGUTL:NGUTR)

*     Place of beams in horizontal projection plane
      DIMENSION YOFFBEAM(NBEAML:NBEAMR)
      DIMENSION DBL(NBEAML:NBEAMR), DBR(NBEAML:NBEAMR)

      COMMON /CONSTR1/ HLOGUT, HUPGUT, WLOGUT, WUPGUT, HGUT, ZPOS_GUT,
     &                 SLOPE_SIDE_GUT
      COMMON /CONSTR2/ HLORID, HUPRID, WLORID, WUPRID, HRID, ZPOS_RID,
     &                 SLOPE_SIDE_RID
      COMMON /CONSTR3/ SLOPE, SPANW
      COMMON /CONSTR4/ HBEAM, WBEAM, ZPOS_BEAM, BEAMDIS
      COMMON /CONSTR6/ REFHEIGHT

      COMMON /ROWCHAR1/ AZIMR, WIDTH, HEIGHT, PATH, VOET
      COMMON /GRGEOM1/ XOFFW, YOFFW, ZOFFW
      COMMON /ROWCHAR7/ DEPTH

      SAVE

      IF( INI ) THEN
         SLOPE_SIDE_GUT = ATAN( HGUT/(0.5*(WUPGUT - WLOGUT)) )
```

```
           SLOPE_SIDE_RID = ATAN( HRID/(0.5*(WLORID - WUPRID)) )

           ZPOS_RID = ZPOS_GUT + 0.5 * SPANW * TAN( SLOPE )
           RETURN
        ENDIF
        PI = 3.1415926
        RADN = 0.017453

        TRID = 1.
        TGUT = 1.
        TBEAM = 1.

        SINELV = SIN (ELEVN )
        COSELV = COS( ELEVN )
        J6 = COSELV * COS( DIFAZIM )
        J7 = COSELV * SIN( DIFAZIM )

*       angle KSI in XZ-plane between incident ray and X-axis     (5.26)
        IF (J7 .EQ. 0.)  THEN
            KSI = .5*PI
        ELSE
            KSI = ATAN ( SINELV / J7 )
        ENDIF

*       angle EPSIL in YZ-plane between incident ray and Y-axis
        IF (J6 .EQ. 0.)  THEN
            EPSIL = .5*PI
        ELSE
            EPSIL = ATAN ( SINELV / J6 )
        ENDIF

        SINKSI = SIN( KSI )
        COSKSI = COS( KSI )
        TANKSI = TAN( KSI )
        TANEPSIL = TAN( EPSIL )

*       VERSHADE is shade in vertical plane
*       HORSHADE is shade in horizontal plane
* Gutter
*                                                                 (5.43a)
        IF (ABS(KSI) .GE. SLOPE_SIDE_GUT) THEN
           LGUT = HGUT*COSKSI + WUPGUT*ABS(SINKSI)
        ELSE
           LGUT = HGUT*COSKSI + 0.5*(WLOGUT+WUPGUT)*ABS(SINKSI)
        ENDIF
        GUT_HORSHADE = LGUT / ABS(SINKSI )
        GUT_VERSHADE = LGUT / ABS(COSKSI )

* Ridge
*                                                                 (5.43b)
        IF( ABS(KSI) .GE. SLOPE_SIDE_RID ) THEN
             LRID = HRID*COSKSI + WLORID*ABS(SINKSI)
          ELSE
             LRID = HRID*COSKSI+ 0.5*(WLORID+WUPRID)*ABS(SINKSI)
        ENDIF
        RID_HORSHADE = LRID / ABS(SINKSI)
        RID_VERSHADE = LRID / ABS(COSKSI)

*       Calculate whether ridges and gutters have coinciding shades;
*       If so, the shade of the ridge is diminished with the area of
*       overlapping shade
        U = SPANW * TAN(SLOPE)
        DO M = 1, 5
*       Coincidence of shades
        DSP = FLOAT( 2 * M - 1) * SPANW
*       Start and finish coincidence
        AKSI1 = (U-HLORID-HUPGUT) / ( DSP + WLORID+WUPGUT )
        AKSI2 = (U+HUPRID+HLOGUT) / ( DSP - WUPRID-WLOGUT )
*       Start and finish total coincidence
        AKSI3 = (U+HUPRID-HUPGUT) / ( DSP - WUPRID+WUPGUT )
        AKSI4 = (U-HLORID+HLOGUT) / ( DSP + WLORID-WLOGUT )

        KSI1 = ATAN( AKSI1 )
        KSI2 = ATAN( AKSI2 )
```

```
      KSI3 = ATAN( AKSI3 )
      KSI4 = ATAN( AKSI4 )

      LC = 0.
      IF( ABS(KSI) .GE. KSI3 .AND. ABS(KSI) .LT. KSI4 ) THEN
         RID_HORSHADE = 0.
         RID_VERSHADE = 0.
         GOTO 70
      ELSEIF( ABS(KSI) .GE. KSI1 .AND. ABS(KSI) .LT. KSI3 ) THEN
         LC = (HLORID+HUPGUT-U ) * COSKSI +
     &             (WLORID+WUPGUT + DSP ) * ABS( SINKSI )
         RID_HORSHADE = AMAX1( 0.0001, RID_HORSHADE - LC )
         RID_VERSHADE = RID_HORSHADE * ABS( TANKSI )
         GOTO 70
      ELSEIF( ABS(KSI) .GE. KSI4 .AND. ABS(KSI) .LT. KSI2 ) THEN
         LC = (HUPRID+HLOGUT+U) * COSKSI +
     &             (WUPRID+WLOGUT- DSP ) * ABS( SINKSI )
         RID_HORSHADE = AMAX1( 0.0001, RID_HORSHADE - LC )
         RID_VERSHADE = RID_HORSHADE * ABS( TANKSI )
         GOTO 70
      ENDIF

      END DO
70    CONTINUE

*     Beam shade
      BEAM_HORSHADE = WBEAM + HBEAM / ABS( TANEPSIL )
      BEAM_VERSHADE = HBEAM + WBEAM * ABS( TANEPSIL )

*     Simple transmission calculations
*     These values are used for low solar elevations
      TGUT2 = AMAX1( 0., 1. - GUT_HORSHADE / SPANW )
      TRID2 = AMAX1( 0., 1. - RID_HORSHADE / SPANW )
      TBEAM2 = AMAX1( 0., 1. - BEAM_HORSHADE / BEAMDIS )

*     Range of gutters and ridges; numbers 0 are the ridges and gutters
*     closest (left side) to the stand
      NGL = -5
      NGR = 5
      NRL = -5
      NRR = 5
      NBL = -5
      NBR = 5

*================================================================
* Gutters

*     No detailed calculations for low solar angles
      IF( ABS( KSI ) .LT. 6. * RADN ) THEN
          TGUT = TGUT2
      ELSE
      CROP_AREA = WIDTH * HEIGHT
      UNIT = SPANW
*     Height of gutter above projection plane
      VERDIS_GUT = ZPOS_GUT  - REFHEIGHT
      XB = XOFFW
      ZB = ABS( REFHEIGHT - ZOFFW )
      XT = XOFFW
      ZT = ABS( REFHEIGHT - ( ZOFFW + HEIGHT ) )

*     Calculate for various gutters the shade cast (AREA)
*     and sum the area's
      TAREA = 0.
      DO IGUT = NGL, NGR

         XLP = 0.
         XRP = 0.
         XLO = 0.
         XRO = 0.
         ICASE = 0

*        Projected distances in projection plane
         XOFFGUT( IGUT ) = FLOAT( IGUT ) * UNIT +
```

```
     &          AREA, AREA1, AREA2, RELAREA, XLO, XRO  )
           ENDIF
           TAREA = TAREA + AREA

199         CONTINUE
           IF( AREA .LT. 0. ) THEN
              WRITE( * , '(A,/,A,I3,2F9.3)')
     &         ' ERROR ROWAREA : AREA  .LT. 0. ',
     &         ' IRID, XLP, XRP : ', IRID, XLP, XRP
             STOP
           ENDIF

        END DO

        TRID = 1. - TAREA / CROP_AREA

        ENDIF

*==================================================================
*        Beams
*==================================================================
*       No detailed calculations for low solar angles
        IF( ABS( EPSIL ) .LT. 6. * RADN ) THEN
            TBEAM = TBEAM2
        ELSE
        CROP_AREA = DEPTH * HEIGHT
        UNIT = BEAMDIS
        VERDIS_BEAM = ZPOS_BEAM  - REFHEIGHT
        YB = YOFFW
        ZB = ABS( REFHEIGHT - ZOFFW )
        YT = YOFFW
        ZT = ABS( REFHEIGHT - ( ZOFFW + HEIGHT ) )

*       Calculate for various beams the shade cast (AREA)
*       and sum the area's
        TAREA = 0.
        DO IBEAM = NBL, NBR

           YLP = 0.
           YRP = 0.
           XLO = 0.
           XRO = 0.
           ICASE = 0

*          Projected distances in projection plane
           YOFFBEAM( IBEAM ) = FLOAT( IBEAM ) * UNIT +
     &                            VERDIS_BEAM / TANEPSIL

*          Calculate area
*          Determine beforehand whether element can cast shade
           IF( EPSIL .GE. 0. ) THEN
              IF( YOFFBEAM( IBEAM ) .GT. 1.5 * UNIT  ) THEN
                  AREA = 0.
                  AREA1 = 0.
                  AREA2 = 0.
                  GOTO 299
              ENDIF
           ELSE
              IF( YOFFBEAM( IBEAM ) .LT. -0.5 * UNIT ) THEN
                  AREA = 0.
                  AREA1 = 0.
                  AREA2 = 0.
                  GOTO 299
              ENDIF
           ENDIF

           IF( EPSIL .GE. 0. ) THEN

*             Places of beginning and end of shades (left and right side)
              YLP = YOFFBEAM( IBEAM ) - 0.5 * BEAM_HORSHADE
              YRP = YOFFBEAM( IBEAM ) + 0.5 * BEAM_HORSHADE
              DBL(IBEAM) = YLP
              DBR(IBEAM) = YRP
```

```
*             Shade of beam
              CALL SHAD3( TANEPSIL,
     &           BEAM_HORSHADE, BEAM_VERSHADE, YLP, YRP,
     &           YB, ZB, YT, ZT, DEPTH, HEIGHT,
     &           AREA, AREA1, AREA2, RELAREA, XLO, XRO  )
          ELSE
*             Places of beginning and end of shades (left and right side)
              DBL(IBEAM) = YOFFBEAM( IBEAM ) - 0.5 * BEAM_HORSHADE
              DBR(IBEAM) = YOFFBEAM( IBEAM ) + 0.5 * BEAM_HORSHADE

*             Mirroring with respect to middle of block
              YLP = 2. * (YOFFW + 0.5 * DEPTH )
     &              - YOFFBEAM( IBEAM ) - 0.5 * BEAM_HORSHADE
              YRP = 2. * (YOFFW + 0.5 * DEPTH )
     &              - YOFFBEAM( IBEAM ) + 0.5 * BEAM_HORSHADE

*             Shade of beam
              CALL SHAD3( ABS( TANEPSIL ),
     &           BEAM_HORSHADE, BEAM_VERSHADE, YLP, YRP,
     &           YB, ZB, YT, ZT, DEPTH, HEIGHT,
     &           AREA, AREA1, AREA2, RELAREA, XLO, XRO  )
          ENDIF
          TAREA = TAREA + AREA

299        CONTINUE
          IF( AREA .LT. 0. ) THEN
             WRITE( * , '(A,/,A,I3,2F9.3)')
     &        ' ERROR ROWAREA : AREA  .LT. 0. ',
     &        ' IBEAM, YLP, YRP : ', IBEAM, YLP, YRP
            STOP
          ENDIF

      END DO

      TBEAM = 1. - TAREA / CROP_AREA

      ENDIF

*=================================================================
      RETURN
      END
*************************************************************************
* Subprogram: SHAD3
* Purpose:
*  calculation of the area of shade cast on a rectangular transsection
*  of a block by an object above it
*
* Description:
*  A horizontal projection line is situated above the rectangle that is
*  representing the block.
*  The projection of the object on the projection plane is running from
*  coordinates XLP to XRP (left and right). The coordinates of the
*  rectangle are for the top left side (XT,ZT) and for the bottom
*  right side (XB,ZB). The rectangle has width WIDTH and height HEIGHT.
*  By simple geometrical calculations the points are calculated where
*  the projection of XLP and XRP are entering the rectangle and where
*  they are exiting the rectangle. The shaded area can consists of
*  a single parallelogram, a single triangle or a combination of these.
*************************************************************************
      SUBROUTINE SHAD3( TANKSI,
     &    HOR_SHADE, VER_SHADE, XLP, XRP,
     &    XB, ZB, XT, ZT, WIDTH, HEIGHT,
     &    AREA, AREA1, AREA2, RELAREA, XLO, XRO )
      IMPLICIT REAL(A-Z)

      LOGICAL LI_HIT, RI_HIT, LI_SIDE, RI_SIDE
      LOGICAL LI_TOP, RI_TOP
      LOGICAL LO_BOTTOM, RO_BOTTOM
      LOGICAL HIT

      INTEGER ICASE
      COMMON /SHAD3_OUT/ ICASE

      ZLI1 = TANKSI * ( XB - XLP )
```

```
      XLI1 = XB

      ZRI1 = TANKSI * ( XB - XRP )
      XRI1 = XB

*     projection of XLP hits side or top of rectangle
      LI_SIDE = .FALSE.
      LI_TOP = .FALSE.
*     projection of XRP hits side or top of rectangle
      RI_SIDE = .FALSE.
      RI_TOP = .FALSE.

      ICASE = 0

      AREA = 0.
      AREA1 = 0.
      AREA2 = 0.

*---- Left point
      IF( ZLI1 .GT. ZB ) THEN
         LI_HIT = .FALSE.

      ELSEIF( ZLI1 .GT. ZT ) THEN
         LI_SIDE = .TRUE.
         LI_HIT = .TRUE.
         XLI = XLI1
         ZLI = ZLI1

      ELSEIF( XLP + ZT / TANKSI .GT. ( XT + WIDTH ) ) THEN
         LI_HIT = .FALSE.
*            HIT = .FALSE.
         RELAREA = 0.
         RETURN

      ELSE
         LI_TOP = .TRUE.
         LI_HIT = .TRUE.
         XLI = XLP + ZT / TANKSI
         ZLI = ZT

      ENDIF

*---- Right point

      IF( ZRI1 .GT. ZB ) THEN
         RI_HIT = .FALSE.
         IF( .NOT. LI_HIT ) THEN
            RELAREA = 0.
            RETURN
         ENDIF

      ELSEIF( ZRI1 .GT. ZT ) THEN
         RI_SIDE = .TRUE.
         RI_HIT = .TRUE.
         XRI = XRI1
         ZRI = ZRI1

      ELSEIF( XRP + ZT / TANKSI .GT. ( XT + WIDTH ) ) THEN
         RI_HIT = .FALSE.
*!! Redundant
         IF( .NOT. LI_HIT ) THEN
*            HIT = .FALSE.
            RELAREA = 0.
            RETURN
         ENDIF

      ELSE
         RI_TOP = .TRUE.
         RI_HIT = .TRUE.
         XRI = XRP + ZT / TANKSI
         ZRI = ZT

      ENDIF
```

```
* ============= Out ===========================
      ZLO1 = TANKSI * ( XB - XLP + WIDTH )
      XLO1 = XB + WIDTH

      ZRO1 = TANKSI * ( XB - XRP + WIDTH )
      XRO1 = XB + WIDTH


*---- Left point
      IF( ZLO1 .GT. ZB ) THEN
         LO_BOTTOM = .TRUE.
         XLO = XLP + ZB / TANKSI
         ZLO = ZB
      ELSE
         LO_BOTTOM = .FALSE.
         XLO = XLO1
         ZLO = ZLO1
      ENDIF

*---- Right point

      IF( ZRO1 .GT. ZB ) THEN
         RO_BOTTOM = .TRUE.
         XRO = XRP + ZB / TANKSI
         ZRO = ZB
      ELSE
         RO_BOTTOM = .FALSE.
         XRO = XRO1
         ZRO = ZRO1
      ENDIF

* ==================================================
      IF( RI_SIDE .AND. .NOT. LI_HIT ) THEN
         IF( RO_BOTTOM ) THEN
* ---- case 1
            AREA = ( XRO - XB ) * ( ZB - ZRI ) * 0.5
            ICASE = 1
         ELSE
* ---- case 2
            AREA1 = ( ZRO - ZRI ) * 0.5 * WIDTH
            AREA2 = ( ZB - ZRO ) * WIDTH
            AREA = AREA1 + AREA2
            ICASE = 2
         ENDIF

      ELSEIF( RI_SIDE .AND. LI_SIDE ) THEN
         IF( RO_BOTTOM ) THEN
* ---- case 3
               AREA1 = HOR_SHADE * (ZB - ZLI )
               AREA2 = ( ZLI - ZRI ) * HOR_SHADE * 0.5
               AREA = AREA1 + AREA2
               ICASE = 3
         ELSE
            IF( LO_BOTTOM ) THEN
* ---- corner out
* ---- case 4
               AREA1 = VER_SHADE * WIDTH
               AREA2 = ( ZLO1 - ZB ) * ( XB + WIDTH - XLO )
               AREA = AREA1 - AREA2
               ICASE = 4
            ELSE
* ---- case 5
               AREA = VER_SHADE * WIDTH
               ICASE = 5
            ENDIF
         ENDIF

      ELSEIF( RI_TOP .AND. LI_SIDE ) THEN
          IF( RO_BOTTOM ) THEN
* ---- case 7
             AREA1 = ( ZT - ZRI1 ) * ( XRI - XT ) * 0.5
             AREA2 = ( ZLI - ZRI1 ) * HOR_SHADE * 0.5
             AREA = HOR_SHADE * (ZB - ZLI ) - AREA1 + AREA2
```

```
              ICASE = 7
           ELSE
              IF( LO_BOTTOM ) THEN
* ---- case 8
                 AREA1 = ( XLO - XB ) * ( ZB - ZLI ) * 0.5
                 AREA2 = ( XT + WIDTH - XRI ) * ( ZRO - ZT ) * 0.5
                 AREA = WIDTH * HEIGHT - AREA1 - AREA2
                 ICASE = 7
              ELSE
* ---- case 9
                 AREA1 = VER_SHADE * WIDTH
                 AREA2 = ( ZT - ZRI1 ) * ( XRI - XT ) * 0.5
                 AREA = AREA1 - AREA2
                 ICASE = 8
              ENDIF
          ENDIF

      ELSEIF( RI_TOP .AND. LI_TOP ) THEN
           IF( RO_BOTTOM ) THEN
* ---- case 10
              AREA = HOR_SHADE * HEIGHT
              ICASE = 9
           ELSE
              IF( LO_BOTTOM ) THEN
* ---- case 11
                 XX = ( ZB - ZRO ) / TANKSI
                 AREA1 = HOR_SHADE * HEIGHT
                 AREA2 = XX * ( ZB - ZRO ) * 0.5
                 AREA = AREA1 - AREA2
                 ICASE = 10
              ELSE
* ---- case 12
                 AREA1 = HOR_SHADE * ( ZRO - ZT )
                 AREA2 = HOR_SHADE * ( ZLO - ZRO ) * 0.5
                 AREA = AREA1 + AREA2
                 ICASE = 11
              ENDIF
          ENDIF
      ELSEIF( .NOT. RI_HIT .AND. LI_TOP ) THEN
* ---- case 13
           IF( LO_BOTTOM ) THEN
              AREA1 = ( XLO - XLI ) * HEIGHT * 0.5
              AREA2 = ( XB + WIDTH - XLO ) * HEIGHT
              AREA = AREA1 + AREA2
              ICASE = 12

           ELSE
* ---- case 14
              AREA = ( XT + WIDTH - XLI ) * ( ZLO - ZT ) * 0.5
              ICASE = 13

*             WRITE( 77, '(10F9.2)')
*     &          XLP, XRP, XLI1, XLI, XLO1, XLO,
*     &          ZLI1, ZLI, ZLO1, ZLO

           ENDIF
      ELSEIF( RI_TOP .AND. .NOT. LI_HIT ) THEN
           AREA1 = ( XRI - XT ) * HEIGHT
           AREA2 = ( XRO - XRI ) * HEIGHT * 0.5
           AREA = AREA1 + AREA 2
           ICASE = 6

      ENDIF

      RELAREA = AREA / HEIGHT * WIDTH

      RETURN
      END
```

Appendix XII:

Listing of general simulation routines

```
*************************************************************************
* SUBPROGRAM: ASTROG
* Type: SUBROUTINE
* Date: June 1990
* Author: H. Gijzen
* Purpose:
*   This subroutine calculates astronomic daylength,
*   and diurnal radiation characteristics such as daily
*   integral of sine of solar elevation, solar constant
*
* Description: Daylength, solar constant are calculated
*       for a given day. Also some intermediate variables are calculated
*       that are needed for
*       - calculation of solar position (declination, SINLD,
*         COSLD) and for
*       - generating diurnal course of radiation (SINLD, COSLD, DSINBE)
*
* Origin: ASTRO by D. van Kraalingen
*         Modified by Jan Goudriaan 4 Febr 1988
*         Modified by Jan Goudriaan and Kees Spitters 7 december 1989
*
* Timer variables: DAYNR
*
* Input:
*  DAYNR       : (R4)  Day number (Jan 1st = 1)                [-]
*  LAT         : (R4)  Latitude                                [radians]
*
* Output:
*
*  SOLARC  : (R4)  corrected solar constant                    [J m-2 s-1]
*  SINLD   : (R4)  Seasonal offset of sine of solar elevation     [-]
*  COSLD   : (R4)  Amplitude of sine of solar elevation           [-]
*  DECL    : (R4)  Declination of sun                           [radians]
*  DAYL    : (R4)  Astronomical daylength (base = 0 degrees)      [h]
*  DSINBE  : (R4)  Daily total of effective solar elevation       [s]
*
* FATAL ERROR CHECKS:
*
*        LAT > 67 degrees, LAT < -67 degrees
*
* WARNINGS: none
*
* SUBPROGRAMS CALLED: none
*
* FILE USAGE: none
*
* Read variables: none
*
* Write variables: none
*
*************************************************************************
      SUBROUTINE ASTROG( DAYNR,LAT,
     &  SOLARC,SINLD,COSLD,DECL,DAYL,DSINBE )

      IMPLICIT REAL (A-Z)

*-----PI and conversion factor from degrees to radians
      PARAMETER( PI=3.141592654, RADN=0.017453292 )

*-----check on input range of parameters
      IF( LAT.GT.  67.*RADN )  STOP 'ERROR IN ASTROG: LAT > 67'
      IF( LAT.LT. -67.*RADN )  STOP 'ERROR IN ASTROG: LAT <-67'

*-----declination of the sun as function of daynumber (DAYNR)
      DECL = -ASIN( SIN(23.45*RADN)*COS(2.*PI*(DAYNR+10.)/365.) )
```

```
*-----SINLD, COSLD and AOB are intermediate variables
      SINLD = SIN(LAT) * SIN(DECL)
      COSLD = COS(LAT) * COS(DECL)
      AOB = SINLD/COSLD

*-----daylength (h)
      DAYL  = 12.0*(1.+2.*ASIN(AOB)/PI)

      DSINBE= 3600.*(DAYL*(SINLD+0.4*(SINLD*SINLD+COSLD*COSLD*0.5))+
     &        12.0*COSLD*(2.0+3.0*0.4*SINLD)*SQRT(1.-AOB*AOB)/PI)

*-----corrected solar constant (J m-2 s-1)
      SOLARC = 1370. * (1.+0.033*COS(2.*PI*DAYNR/365.))

      RETURN
      END
************************************************************************
* Function for conversion year, month and day to daynumber of year
************************************************************************
      INTEGER FUNCTION DAYNUM( IYEAR, MONTH, DAY )
      IMPLICIT REAL(A-Z)

      INTEGER IYEAR, MONTH, DAY
      INTEGER CUMDAY( 0:11 )

      DATA CUMDAY /0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304,
     &                 334/

*     If MOD( IYEAR, 4 ) .EQ. 0 then leap-year
      IF( MOD( IYEAR, 4 ) .EQ. 0  .AND. MONTH .GE.3) THEN
         DAYNUM = CUMDAY( MONTH-1 ) + DAY + 1
      ELSE
         DAYNUM = CUMDAY( MONTH-1 ) + DAY
      ENDIF

      RETURN
      END
*----------------------------------------------------------------------*
*  REAL FUNCTION TEMP                                                  *
*  Authors: Daniel van Kraalingen, Kees Rappoldt                       *
*  Date   : 9-Jan-1987                                                 *
*  Purpose: This  function  is meant to reconstruct  the  course  of   *
*           temperature during a full day.  At daylight, temperature   *
*           follows a sinusoidal curve, at nighttime, an exponential   *
*           decrease is assumed.  To fully reconstruct the course of   *
*           temperature,  four temperatures are*needed.  The minimum   *
*           and maximum temperature of the particular day,  but also   *
*           the  maximum of the previous day and the minumum of  the   *
*           next day.                                                  *
*                                                                      *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)     *
*  name   type meaning                                    units  class *
*  ----   ---- -------                                    -----  ----- *
*  TEMP    R4  Function name, returned temperature          C      O   *
*  IWAR    I4  output, when .NE.0  warning !!!              -      C,O *
*  TMAX1   R4  Maximum temperature of previous day          C      I   *
*  TMIN2   R4  Minimum temperature of current day           C      I   *
*  TMAX2   R4  Maximum temperature of current day           C      I   *
*  TMIN3   R4  Minimum temperature of next day              C      I   *
*  DAYL    R4  Daylength                                  hours    I   *
*  HOUR    R4  Time of day                                hours    I   *
*                                                                      *
*  FATAL ERROR CHECKS (execution terminated, message):                 *
*  HOUR < 0  or  HOUR > 24                                             *
*  TMIN2 > TMAX2                                                       *
*                                                                      *
*  WARNINGS                                   value of IWAR returned   *
*  TMIN2 > TMAX1                                        1              *
*  TMAX2 < TMIN3                                       -2              *
*                                                                      *
*  SUBROUTINES and FUNCTIONS called : ERROR                            *
*  FILE usage : none                                                   *
*----------------------------------------------------------------------*
```

```
      REAL FUNCTION DCURTEMP (IWAR,
     $                    TMAX1,TMIN2,TMAX2,TMIN3,DAYL,HOUR)
      IMPLICIT REAL (A-Z)
      INTEGER IWAR
      SAVE

      PARAMETER (PI=3.14159, TAU=4.)

C     errors and warnings
      IWAR = 0
*      IF (HOUR.LT.0.)  CALL ERROR ('TEMP','HOUR < 0')
*      IF (HOUR.GT.24.) CALL ERROR ('TEMP','HOUR > 24')
*      IF (TMIN2.GT.TMAX2) CALL ERROR ('TEMP','TMIN > TMAX')
      IF (TMIN2.GT.TMAX1) IWAR = +1
      IF (TMAX2.LT.TMIN3) IWAR = -2

      SUNRIS = 12.-0.5*DAYL
      SUNSET = 12.+0.5*DAYL

      IF (HOUR.LT.SUNRIS) THEN
C        hour between midnight and sunrise
         TSUNST = TMIN2+(TMAX1-TMIN2)*SIN(PI*(DAYL/(DAYL+3.)))
         NIGHTL = 24.-DAYL
         TEMP1  = (TMIN2-TSUNST*EXP(-NIGHTL/TAU)+
     $            (TSUNST-TMIN2)*EXP(-(HOUR+24.-SUNSET)/TAU))/
     $            (1.-EXP(-NIGHTL/TAU))
      ELSE IF (HOUR.LT.13.5) THEN
C        hour between sunrise and normal time of TMAX2
         TEMP1 = TMIN2+(TMAX2-TMIN2)*SIN(PI*(HOUR-SUNRIS)/(DAYL+3.))
      ELSE IF (HOUR.LT.SUNSET) THEN
C        hour between normal time of TMAX2 and sunset
         TEMP1 = TMIN3+(TMAX2-TMIN3)*SIN(PI*(HOUR-SUNRIS)/(DAYL+3.))
      ELSE
C        hour between sunset and midnight
         TSUNST = TMIN3+(TMAX2-TMIN3)*SIN(PI*(DAYL/(DAYL+3.)))
         NIGHTL = 24.-DAYL
         TEMP1  = (TMIN3-TSUNST*EXP(-NIGHTL/TAU)+
     $            (TSUNST-TMIN3)*EXP(-(HOUR-SUNSET)/TAU))/
     $            (1.-EXP(-NIGHTL/TAU))
      END IF

      DCURTEMP = TEMP1

      RETURN
      END
**********************************************************************
* SUBPROGRAM: LONGRAD
* Type: REAL FUNCTION
* Purpose:
*     Calculation of long wave radiation exchange between
*     two objects. When temperature of object 1 is higher than
*     of object 2, radiation flux has positive sign.
*
* Input:
*   SPSURF   (R4) : specific surface of object1                 [-]
*   TOBJ1    (R4) : temperature of object 1                     [oC]
*   TOBJ2    (R4) : temperature of object 2                     [oC]
*
* Output:
*   LONGRAD (R4)  : long wave rad. flux emitted from obj. 1 to
*                   obj. 2                                      [J m-2 s-1]
**********************************************************************
      REAL FUNCTION LONGRAD( SPSURF, Tobj1, Tobj2 )
      IMPLICIT REAL(A-Z)

*     Stephan-Boltzmann constant [W m-2 K-4]
      SIGMA = 5.67E-8

      Tdiff = Tobj1 - Tobj2

*     Mean temperature
      MeanTK = (Tobj1 + Tobj2)/2. + 273.
```

```
*     Radiation flux (J m-2 s-1)
      LONGRAD = SPSURF * 4. * SIGMA * Tdiff * ( MeanTK ** 3 )

      RETURN
      END
************************************************************************
* SUBPROGRAM: SUNPOS
* Type: SUBROUTINE
* Date: 08-FEB-1989
* Author: H. Gijzen
* Purpose: Calculation of position of sun at given day of year,
*          time of day and latitude
*
* Description: Calculates solar elevation (height above horizon) and
*       solor azimuth (difference of direction of sun with north-south).
*       Azimuth: east negative sign, west positive sign
*
* Input:
*    LAT      : (R4) latitude of location                  [radians]
*    SINLD    : (R4) seasonal offset of sine of solar height [-]
*    COSLD    : (R4) amplitude of sine of solar height     [-]
*    DECL     : (R4) declination                           [radians]
*    SOLHR    : (R4) time of the day (solar time)          [h]
* Output:
*    ELEVN    : (R4) elevation of sun                      [radians]
*    AZIMS    : (R4) azimuth of sun                        [radians]
*    SINELV   : (R4) sine of solar elevation               [-]
*
* FATAL ERROR CHECKS: none
*
* WARNINGS: none
*
* SUBPROGRAMS CALLED: none
*
* FILE USAGE: none
*
* Read variables: none
*
* Write variables: none
*
* COMMENT:
************************************************************************
      SUBROUTINE SUNPOS( LAT,SINLD,COSLD,DECL,SOLHR,
     &                   ELEVN,AZIMS,SINELV )
      IMPLICIT REAL(A-Z)

      PI = 3.1415926

*---  Sine of solar elevation (inclination)
      SINELV = SINLD+COSLD*COS(2*PI*(SOLHR+12.)/24.)
      ELEVN = ASIN( SINELV )

*---  Solar azimuth
*     function from Campbell, 1981; Encyclop. of Physiol. Plant Ecol.,
*     vol. 12A
*     Cosine function is used because ACOS-function gives angles
*     higher than 90 degrees when solar azimuth is passing East-West line
      COSAZ = - (SIN(DECL) - SIN(LAT)*SINELV) /
     &        (COS(LAT)*COS(ELEVN))

*     Place upper limit and under limit to COSAZ as this variable can
*     be more than 1 or less than -1 because of calculation inaccuracy
      IF( COSAZ .LT. -1.0 ) THEN
            COSAZ = -1.0
      ELSEIF( COSAZ .GT. 1.0 ) THEN
            COSAZ = 1.0
      ENDIF
      AZIMS = ACOS( COSAZ )

*---  East has negative sign, West has positive sign
      IF( SOLHR.LE.12. ) THEN
          AZIMS = -AZIMS
      ENDIF
```

```
*---  Limit set to SINELV
      IF( SINELV .LT. 0. ) THEN
          SINELV = 0.
      ENDIF

      RETURN
      END
**********************************************************************
* Calculation of number days passed since 1-1-1980
**********************************************************************
      INTEGER FUNCTION TDAY80( IYEAR, IDAY )
      IMPLICIT REAL(A-Z)

      INTEGER TDAY, IY, IYEAR, DIF_YEAR, IDAY

*     Determine number of days from 1-1-1980 onwards until
*     31 december in previous year
      TDAY = 0
      DIF_YEAR = IYEAR - 80
      DO IY = 0, DIF_YEAR - 1
          TDAY = TDAY + 365
          IF( MOD(IY,4) .EQ. 0) TDAY = TDAY + 1
      END DO

*     Total number of days from 1-1-1980 onwards
      TDAY80 = TDAY + IDAY - 1

      RETURN
      END
***********************************************************************
* SUBPROGRAM: TRANSM2
* Type: SUBROUTINE
* Date: Jul-1994
* Author: H. Gijzen
* Purpose: Calculation of transmissivity of
*   greenhouse cover for diffuse and direct global radiation, PAR
*   and UV.
*
* Description:
*   Uses output from detailed model of Bot (1983).
*   Calculates transmissivity of greenhouse for direct radiation by
*   interpolation in table. Transmissivity for diffuse radiation is
*   constant factor. A correction factor is used for the transmission
*   of UV.
*
* Control variables: ITASK
* Init variables: ITASK
*
* Input:
*  ITASK      : (I4)  control variable for initialization
*                     (ITASK=1) and transmission calc. (ITASK=2)    [-]
*  IUTRAN     : (I4)  unit nr. for file reading                    [-]
*  FILNAM     : (CH*) name of input file                           [-]
*  AZIMS      : (R4)  azimuth of sun                               [radians]
*  AZIMGR     : (R4)  azimuth greenhouse                           [radians]
*  ELEVN      : (R4) elevation of sun                              [radians]
* Output:
*  TRDIF   : (R4) transmissivity of greenhouse for diffuse radiation
*                                                                  [-]
*  TRCOR_UV: (R4) correction fot transmissivity for UV radiation [-]
*                                                                  [-]
*  TRCON   : (R4) transmission of the construction for direct radiation
*                                                                  [-]
*  TRGLAS  : (R4) transmission of the glass for direct radiation
*                                                                  [-]
*
* FATAL ERROR CHECKS:
*   when premature end of input file found
*
* WARNINGS: none
*
* SUBPROGRAMS CALLED: AZINT
*
* FILE USAGE:
```

```
*   unit          file name     description
*   ====          =========     ===========
*  IUTRAN          FILNAM       input file with table of transmissivities
*
* Read variables:
*   name      unit        description
*   ====      ====        ===========
*   TRDIF    IUTRAN       transmissivity diffuse radiation
*   FMT      IUTRAN       format for reading transmissivities
*                         direct radiation
*   AZ       IUTRAN       2-dim. table azimuth values
*   EL       IUTRAN       1-dim. table elevation layers
*   TBCON    IUTRAN       2-dim. table transmissivities construction
*   TBGLAS   IUTRAN       2-dim. table transmissivities glass
*
* Write variables: none
*
* COMMENT:
*      when ITASK = 1 (initialization) data of transmissivities are
*      filled by reading from data file
*      when ITASK > 1 interpolation in tables takes places to find
*      transmissivity of direct radiation for given solar position
*      transmissivities are grouped according to azimuth values with
*      the same elevation (elevation layer)
*
* Note: interpolation here at ITASK .GT. 1 ( in TRANSM at ITASK .EQ. 1)
************************************************************************
      SUBROUTINE TRANSM2( ITASK, IUTRAN, FILNAM,
     &            AZIMGR, AZIMS, ELEVN, TRDIF, TRCOR_UV, TRCON, TRGLAS )
      IMPLICIT REAL(A-Z)
      INTEGER ITASK
      INTEGER EOFSKP, IOSSKP
      INTEGER IUTRAN
      CHARACTER*(*) FILNAM
      CHARACTER*40 LABEL, FMT

      INTEGER NLAYER, NENTR, IA, IE
      INTEGER I, IXMAX, IXMIN
      DIMENSION NENTR(20), EL (20)
      DIMENSION AZ(20,20), TBCON(20,20), TBGLAS(20,20)


      IF (ITASK .EQ. 1) THEN
         OPEN( UNIT = IUTRAN, FILE = FILNAM, STATUS = 'OLD' )

         CALL SKIPCM( IUTRAN, '*', EOFSKP, IOSSKP )
         IF (EOFSKP .EQ. -1) THEN
            WRITE (*,'(A,A,A)')
     &           ' TRANSM reading file ', FILNAM,
     &           ' End Of File found when searching TRANSM DIFFUSE '
            STOP
         ENDIF

*---  Diffuse radiation transmissivity
         READ( IUTRAN, * ) TRDIF


*---  Correction factor for UV-radiation
         CALL SKIPCM( IUTRAN, '*', EOFSKP, IOSSKP )
         IF (EOFSKP .EQ. -1) THEN
            WRITE (*,'(A,A,A)')
     &           ' TRANSM reading file ', FILNAM,
     &           ' End Of File found when searching TRCOR_UV '
            STOP
         ENDIF

*     in model transmissivity of UV is obtained by multiplying
*     transmissivity for global radiation by TRCOR_UV
         READ( IUTRAN, * ) TRCOR_UV


*---  Number of elevation layers
         CALL SKIPCM( IUTRAN, '*', EOFSKP, IOSSKP )
         IF (EOFSKP .EQ. -1) THEN
```

```
            WRITE (*,'(A,A,A)')
     &            ' TRANSM reading file ', FILNAM,
     &            ' End Of File found when searching NUMBER OF LAYERS '
            STOP
         ENDIF

         READ (IUTRAN, '(I6)') NLAYER

         CALL SKIPCM( IUTRAN, '*', EOFSKP, IOSSKP )
         IF( EOFSKP .EQ. -1 ) THEN
            WRITE (*,'(A,A,A)')
     &            ' TRANSM reading file ', FILNAM,
     &            ' End Of File found when searching FORMAT '
            STOP
         ENDIF

*---  Format for reading azimuth and transmissivity tables
         READ (IUTRAN, '(A)') FMT

         CALL SKIPCM( IUTRAN, '*', EOFSKP, IOSSKP )
         IF( EOFSKP .EQ. -1 ) THEN
            WRITE (*,'(A,A,/,A,A)')
     &            ' TRANSM reading file ', FILNAM,
     &            ' End Of File found when searching beginning of',
     &            ' direct transmissivity data '
            STOP
         ENDIF

         DO 50 IE=1,NLAYER
*---  Elevation of elevation layer (degrees)
            READ( IUTRAN,*, END=51 ) EL (IE)
*---  Number of entries in elevation layer
            READ (IUTRAN,'(I8)', END=51) NENTR(IE)
*---  Azimuth values corresponding with transmissivity data
            READ (IUTRAN, FMT, END=51) (AZ(IA,IE),     IA=1,NENTR(IE))
*---  Transmissivity construction
            READ (IUTRAN, FMT, END=51) (TBCON(IA,IE),   IA=1,NENTR(IE))
*---  Transmissivity glass
            READ (IUTRAN, FMT, END=51) (TBGLAS(IA,IE), IA=1,NENTR(IE))
50       CONTINUE
         GOTO 52
51       CONTINUE

         WRITE (*,'(A,A,/,A,A,/,A,I5,/,A,I5)')
     &        ' TRANSM reading file ', FILNAM,
     &        ' End Of File found when reading',
     &        ' direct transmissivity data',
     &        ' Total number of elevation layers is : ', NLAYER,
     &        ' Currently reading layer nr : ',IE
         STOP

52       CONTINUE

         CLOSE( IUTRAN )

      ELSE

      RADN = 0.017453292

*---  Conversion of radians to degrees
      A1 = (AZIMS - AZIMGR) / RADN
      A1 = AMOD( A1, 180. )
      E = ELEVN / RADN

*---  If necessary, mirroring of azimuth
      IF (A1.GE.90..AND.A1.LE.180.) A=180.-A1
      IF (A1.LT.0..AND.A1.GT.-90.) A=-A1
      IF (A1.LE.-90..AND.A1.GE.-180.) A=180.+A1
      IF (A1.GE.0. .AND. A1.LT.90.) A = A1

*---  Search for layer number
      DO 5 I=1,NLAYER
        IF(E.LT.EL(I)) GOTO 10
5     CONTINUE
```

```
      IXMIN = NLAYER
      IXMAX = NLAYER
      GOTO 20

10    IXMIN = MAX0(I-1,1)
      IXMAX = I

*---  Interpolation in azimuth
20    TC1 = AZINT( A,TBCON(1,IXMIN),AZ(1,IXMIN),NENTR(IXMIN) )
      TG1 = AZINT( A,TBGLAS(1,IXMIN),AZ(1,IXMIN),NENTR(IXMIN) )

      IF(IXMIN.EQ.IXMAX) THEN
         TRCON = TC1
         TRGLAS = TG1
      ELSE
*---  Interpolation in azimuth
         TC2 = AZINT( A,TBCON(1,IXMAX),AZ(1,IXMAX),NENTR(IXMAX) )
         TG2 = AZINT( A,TBGLAS(1,IXMAX),AZ(1,IXMAX),NENTR(IXMAX) )
*---  Interpolation in elevation
         TRCON = TC1+(TC2-TC1)*(E-EL(IXMIN))/(EL(IXMAX)-EL(IXMIN))
         TRGLAS = TG1+(TG2-TG1)*(E-EL(IXMIN))/(EL(IXMAX)-EL(IXMIN))
      END IF


      ENDIF

      RETURN
      END
************************************************************************
* SUBPROGRAM: AZINT
* Type: FUNCTION
* Date: OKT-1986
* Author: H. Gijzen
* Modifications:
* Purpose:
*      Interpolation in azimuth-table. Corresponding value in table of
*      transmissivity greenhouse construction or glass is
*      output of function.
* Control variables:
* Init variables:
* Timer variables:
*
* Input:
*    AZIMUTH     : (R4)  azimuth of beam                              [degrees]
*    AZIMTB      : (R4)  azimuth table (length 20)                    [-]
*    TRTB        : (R4)  transmissivity table (length 20)             [-]
*    NAZFIL      : (I4)  number of places in table that are filled    [-]
*
* Output:
*    AZINT       : (R4)  transmissivity found in table                [-]
*
* FATAL ERROR CHECKS: none
*
* WARNINGS: none
*
* SUBPROGRAMS CALLED: none
*
* FILE USAGE: none
*
* Read variables: none
*
* Write variables: none
*
* COMMENT:
************************************************************************
      FUNCTION AZINT (AZIMUTH,TRTB,AZIMTB,NAZFIL)
      IMPLICIT REAL(A-Z)
      INTEGER I, NAZFIL
      DIMENSION AZIMTB(20),TRTB(20)

      DO 30 I=1, NAZFIL
         IF(AZIMUTH.LT.AZIMTB(I)) GOTO 10
30    CONTINUE
      AZINT = TRTB(NAZFIL)
```

```
      RETURN

10    IF(I.EQ.1) THEN
         AZINT = TRTB(1)
      ELSE
         AZINT = TRTB(I-1)+(TRTB(I)-TRTB(I-1)) *
     &           (AZIMUTH-AZIMTB(I-1))/(AZIMTB(I)-AZIMTB(I-1))
      END IF
      RETURN
      END
```

•

Appendix XIII: Explanation of variables and parameters

ASRQ	assimilate requirement crop dry matter	g CH_2O g DM^{-1}
ASRQLV	assimilate requirement leaves	g CH_2O g DM^{-1}
ASRQRT	assimilate requirement roots	g CH_2O g DM^{-1}
ASRQSO	assimilate requirement storage org.	g CH_2O g DM^{-1}
ASRQST	assimilate requirement stems	g CH_2O g DM^{-1}
ATMTR	atmospheric transmission	-
AZIMGR	azimuth greenhouse	radians
AZIMS	azimuth of sun	radians
b	parameter model Ball et al.	µmol m^{-2} s^{-1}
CO_2AIR	CO_2 concentration of greenhouse air	µl l^{-1}
CO_2I	leaf internal CO_2 concentration	µmol mol^{-1}
CO_2IN	leaf internal CO_2 concentration	µmol mol^{-1}
CO_2SURF	CO_2 concentration at leaf surface	µmol mol^{-1}
CONV	convective heat loss from leaf	J m^{-2} s^{-1}
COSLD	amplitude of sine of solar height	-
CTWT	cumulative dry weight of crop	g DM m^{-2}
CWLV	cumulative dry weight of leaves	g DM m^{-2}
CWRT	cumulative dry weight of roots	g DM m^{-2}
CWSO	cumulative dry weight of storage organs	g DM m^{-2}
CWST	cumulative dry weight of stems	g DM m^{-2}
DATA_DAY_MIN	total number of minutes since 0.00 h in data file	min
DAYL	astronomical daylength (base = 0 degrees)	h
DAYMIN	total number of minutes since midnight	min
DAYNR	day number of year (Jan 1st = 1)	-
DAYTASKS	flag to indicate when daily tasks should be done	-
DECL	declination of sun	radians
DELT	time step	h
DELT	time step	d
DELTF	time step for fast loop	h
DELTMIN	time step for fast loop	min
DMAINT	daily total of maintenance costs	g CH_2O m^{-2} d^{-1}
DSINBE	daily total of effective solar elevation	s
DTGA	daily total gross assimilation	g CO_2 m^2 d^{-1}
EFF0	leaf light use efficiency in absence of oxygen	mg CO_2 (µmol phot.)$^{-1}$
EFFRAD	photon flux absorbed by photosystems	µmol m^{-2} s^{-1}
ELEVN	elevation of sun	radians
FARPHG	leaf gross photosynthesis	µmol CO_2 m^{-2} s^{-1}
FCICS	param. for dependance Ci on Cs	-
FCO_2CURV	param. for curvature CO_2 response PNMAX	-
FCVPD	param. for dependance FCICS on VPDsurf	kPa^{-1}
FDIFGLOB	fraction diffuse in global radiation	-
FDIFPAR	fraction diffuse in PAR	-
FINDAY	finish day of simulation (day number of year)	d

FINYEAR	finish year of simulation	yr
FLV	dry matter partitioning to leaves	-
FRT	dry matter partitioning to roots	-
FSO	dry matter partitioning to storage organs	-
FST	dry matter partitioning to stems	-
GAMMA	CO_2 compensation point in absence of photorespiration	$\mu mol\ mol^{-1}$
GB	conductance leaf boundary layer	$m\ s^{-1}$
GCUT	cuticular conductance to H_2O	$m\ s^{-1}$
GLEAF	leaf conductance	$m\ s^{-1}$
GLEAF0	leaf conductance at zero leaf gross phot.	$m\ s^{-1}$
GLEAFD	leaf conductance in the dark	$m\ s^{-1}$
GLOBDIF	diffuse global radiation inside greenhouse	$J\ m^{-2}\ s^{-1}$
GLOBDIFO	diffuse global radiation outside greenhouse	$J\ m^{-2}\ s^{-1}$
GLOBDIR	direct global radiation inside greenhouse	$J\ m^{-2}\ s^{-1}$
GLOBDIRO	direct global radiation outside greenhouse	$J\ m^{-2}\ s^{-1}$
GLOBRAD	global radiation inside greenhouse	$J\ m^{-2}\ s^{-1}$
GLOBRADDIF	diffuse global radiation	$J\ m^{-2}\ s^{-1}$
GLOBRADDIR	direct global radiation	$J\ m^{-2}\ s^{-1}$
GLRADO	global radiation outside greenhouse	$J\ m^{-2}\ s^{-1}$
GLTOT	total canopy conductance (sum of stom. + cut. cond.)	$m\ s^{-1}$
GLV	rate of DM increase of leaves	$g\ m^{-2}\ d^{-1}$
GMAXD	maximal leaf conductance at night	$m\ s^{-1}$
GNVPD	parameter for leaf surface VPD response of GMAXD	kPa^{-1}
GRT	rate of DM increase of roots	$g\ m^{-2}\ d^{-1}$
GS	stomatal conductance	$m\ s^{-1}$
GS_mol	stomatal conductance to H_2O diffusion	$mol\ m^{-2}\ s^{-1}$
GSin	initial estimate for stomatal conductance	$m\ s^{-1}$
GSMAX	maximal stomatal conductance	$m\ s^{-1}$
GSO	rate of DM increase of stor. org.	$g\ m^{-2}\ d^{-1}$
GST	rate of DM increase of stems	$g\ m^{-2}\ d^{-1}$
GSTOT	total canopy conductance (sum of stom. cond.)	$m\ s^{-1}$
GTW	rate of DM increase of crop	$g\ m^{-2}\ d^{-1}$
HFCR	thermal rad. to greenhouse cover	$J\ m^{-2}\ s^{-1}$
HFCRTOT	thermal radiation from canopy to roof	$J\ m^{-2}\ s^{-1}$
HFPC	thermal rad. from heating pipes	$J\ m^{-2}\ s^{-1}$
HFPCTOT	thermal radiation from pipes to canopy	$J\ m^{-2}\ s^{-1}$
HFSC	thermal rad. from ground	$J\ m^{-2}\ s^{-1}$
HFSCTOT	thermal radiation from soil to canopy	$J\ m^{-2}\ s^{-1}$
HOUR	hour of day	h
IDAY	day number of year	d
IOPHASE	control variable for I/O	
ITASK	control variable for initialization (ITASK=1), rate calculation (2), integration (3), terminal calculations (4), and resetting (5)	-
ITOLD	old value of ITASK	-
IYEAR	year	yr
J	potential electron transport rate	$\mu mol\ m^{-2}\ s^{-1}$
JMAX	rate of electron transport	$\mu mol\ e-\ m^{-2}\ s^{-1}$
JMAX25	maximal rate of electron transport, at 25 °C	$\mu mol\ e-\ m^{-2}\ s^{-1}$
KC	M.M. constant for CO_2 binding to RuBP	$\mu mol\ mol^{-1}$
KC25	Michaelis Menten constant for CO_2 to RuBP under standard conditions	$\mu mol\ mol^{-1}$

KDIF	extinc. coeff. for diffuse PAR	-
KDIFBL	diffuse extinc. coeff. canopy with black leaves	-
KDIFN	extinc. coeff. for diffuse NIR	-
KDIR	extinc. coeff. for direct PAR	-
KDIRBL	direct extinc. coeff. canopy with black leaves	-
KDIRN	extinc. coeff. for direct NIR	-
KO	M.M. constant for O2 binding to RuBP	mmol mol^{-1}
KO25	Michaelis Menten constant for O2 binding to RuBP under standard conditions	mmol mol^{-1}
LAI	Leaf Area Index	-
LAIC	partial leaf area index	-
LAT	latitude of location	radians
LGHTCON	mumol photons per Joule PAR	μmol J^{-1}
m	parameter model Ball et al.	-
MAINLV	maint. costs leaves at 25 °C	g CH_2O g dm^{-1} d^{-1}
MAINRT	maint. costs roots at 25 °C	g CH_2O g dm^{-1} d^{-1}
MAINSO	maint. costs storage org. at 25 °C	g CH_2O g dm^{-1} d^{-1}
MAINST	maint. costs stems at 25 °C	g CH_2O g dm^{-1} d^{-1}
MAINT	crop maintenance respiration	mg CH_2O m^{-2} s^{-1}
MAINTS	crop maintenance respiration at 25 °C	mg CH_2O m^{-2} s^{-1}
NETRAD	net radiation of canopy	J m^{-2} s^{-1}
NIR_DF	absorbed diffuse NIR at given leaf layer	J m^{-2} s^{-1}
NIR_REF	NIR reflected by ground surface	J m^{-2} s^{-1}
NIR_S	absorbed NIR by sunlit leaves (angle dependent)	J m^{-2} s^{-1}
NIR_SH	absorbed total diffuse NIR at given leaf layer	J m^{-2} s^{-1}
NIRABS	absorbed NIR energy flux	J m^{-2} s^{-1}
NIRDIF	flux diffuse NIR	J m^{-2} s^{-1}
NIRDIFO	diffuse NIR outside greenhouse	J m^{-2} s^{-1}
NIRDIR	flux direct NIR	J m^{-2} s^{-1}
NIRDIR_D	absorbed direct comp. of direct NIR at given leaf layer	J m^{-2} s^{-1}
NIRDIR_T	absorbed total direct NIR at given leaf layer	J m^{-2} s^{-1}
NIRDIRO	direct NIR outside greenhouse	J m^{-2} s^{-1}
O2	internal O_2 concentration	mmol mol^{-1}
OAV	aveage projection leaves into direction beam	-
OUTDELDAY	output interval for daily output	d
OUTDELMIN	output interval for output in fast loop	min
OUTPUTD	output flag for daily output	-
OUTPUTF	flag for output in fast loop	-
PAR_DF	absorbed diffuse PAR at given leaf layer	J m^{-2} s^{-1}
PAR_REF	PAR reflected by ground surface	J m^{-2} s^{-1}
PAR_S	absorbed PAR by sunlit leaves (angle dependent)	J m^{-2} s^{-1}
PAR_SH	absorbed total diffuse PAR at given leaf layer	J m^{-2} s^{-1}
PARABS	absorbed PAR energy flux	J m^{-2} s^{-1}
PARDIF	flux diffuse PAR	J m^{-2} s^{-1}
PARDIR	flux direct PAR	J m^{-2} s^{-1}
PARDIR_D	absorbed direct comp. of direct PAR at given leaf layer	J m^{-2} s^{-1}
PARDIR_T	absorbed total direct PAR at given leaf layer	J m^{-2} s^{-1}
PARDIRTOT	direct PAR absorbed by canopy	J m^{-2} s^{-1}
PAROUT	PAR outside the greenhouse	J m^{-2} s^{-1}
PGROS	canopy gross photosynthesis	mg CO_2 m^{-2} s^{-1}
PGROSL	leaf gross photosynthesis	mg CO_2 m^{-2} s^{-1}

PHOTRED	factor for reduction of photosynth. capacities with depth in canopy	-
PHOTREDCOF	factor for reduction phot. capacity in canopy	-
PMM	leaf maximal endogeneous photosynth. capacity	mg CO_2 m^{-2} s^{-1}
PNETL	leaf net assimilation	mg CO_2 m^{-2} s^{-1}
PNMAX	leaf maximal net photosynthesis	mg CO_2 m^{-2} s^{-1}
PSIPL	water potential of the crop	MPa
PSIROOTM	water potential of the Root Medium	MPa
PSIWIL	water potential at wilting	MPa
PSYCHR	psychrometric constant	kPa $^oC^{-1}$
Q10KM	Q10 effective M.M. constant Rubisco	-
Q10MN	Q10 maintenance respiration	-
Q10RD	Q10 of leaf dark respiration	-
Q10VCM	Q10 of carboxylation velocity	-
RADABS	short wave radiation absorbed by canopy	J m^{-2} s^{-1}
RB	boundary layer resistance for vapour	s m^{-1}
RB_mol	boundary layer resistance to H_2O	s m^2 mol^{-1}
RBH	leaf boundary layer resistance for heat	s m^{-1}
RBTH	leaf total heat resistance	s m^{-1}
RC	carboxylation resistance	s m^{-1}
RCUT	cuticula resistance for vapour	s m^{-1}
RD	leaf dark respiration	mg CO_2 m^{-2} s^{-1}
RD	leaf dark respiration	µmol CO_2 m^{-2} s^{-1}
RD25	leaf dark respiration at 25 ^oC	µmol CO_2 m^{-2} s^{-1}
REFGR	reflection coefficient of ground surface	-
REFHN	reflection coeff. of canopy for diffuse NIR	-
REFHND	reflection coeff. of canopy for direct NIR	-
REFHP	reflection coeff. of canopy for diffuse PAR	-
REFHPD	reflection coeff. of canopy for direct PAR	-
REFTMP	reference temperature maint. resp.	^oC
RESWAT	resistance of crop for water transport	MPa s m^2 mg^{-1}
RHOCP	volumetric heat capacity of air	J m^{-3} oC^{-1}
RIONUPT	ion uptake flux	mol m^{-2} s^{-1}
RNG	range of leaf projections into direction of beam	-
RRAD	leaf resistance for thermal radiation	s m^{-1}
RTHRAD	resistance for thermal rad. at top of can.	s m^{-1}
RTHRAD	resistance for thermal radiation at top of canopy	s m^{-1}
RWATCON	rate of water uptake by crop	mg H_2O m^{-2} s^{-1}
RWATCONWI	relative water content crop at wilting	-
RWUPT	rate of water uptake	mg m^{-2} s^{-1}
SCN	scattering coeff. of leaves for NIR	-
SCP	scattering coeff. of leaves for PAR	-
SCREEN	fraction opening of screens	-
SIM_DAY_MIN	total number of minutes since 0.00 h as counted by time loop	min
SINELV	sine of solar elevation	-
SINLD	seasonal offset of sine of solar height	-
SOLARC	corrected solar constant	J m^{-2} s^{-1}
SOLHR	time of the day (solar time)	h
SSPB	specific surf. of heating pipes below canopy	-
SSPT	specific surf. of heating pipes above canopy	-
STARTDAY	start day of simulation (day number of year)	d
STYEAR	start year of simulation	yr

SUNPER	absorbed direct PAR by leaves perpendicular to beam, at top of canopy	J m^{-2} s^{-1}
SUNPERN	absorbed direct NIR by leaves perpendicular to beam, at top of canopy	J m^{-2} s^{-1}
SUNRISE	time of sunrise (solar time)	h
SUNSET	time of sunset (solar time)	h
TAIR	temperature greenhouse air	^oC
TEMPAIR	temperature of greenhouse air	^oC
TEMPAIR_OUT	temperature outside air	^oC
TERMNL	flag for terminal tasks	-
TGROUND	temperature of greenhouse floor	^oC
THETA	param. for degree of curvature of light response of electron transport	-
TIMCOR	difference local and solar time	h
TLEAF	leaf temperature	^oC
TMIN80	total number of minutes since 1-1-1980:0.00 h	min
TOTDAY80	total number of days since 1-1-1980:0.00 h (at 1-1-1980 TOTDAY80 is 1)	d
TPIPE	temperature of heating pipes	^oC
TRAN_SIM	simulated canopy transpiration	mg H_2O m^{-2} s^{-1}
TRANLEAF	leaf transpiration	mg H_2O m^{-2} s^{-1}
TRANSP	rate of transpiration	mg m^{-2} s^{-1}
TRCON	transmission cover construction for direct radiation	-
TRCOR_UV	correction factor transmission for UV	-
TRDIF	transmissivity greenhouse for diffuse global rad.	-
TRDIR	transmission greenhouse for direct global rad.	-
TRGLAS	transmission cover cladding for direct radiation	-
TROOF	temperature of greenhouse cover	^oC
UV_REF	UV reflected by ground surface	J m^{-2} s^{-1}
UVDIF	flux diffuse UV	J m^{-2} s^{-1}
UVDIFO	diffuse UV outside greenhouse	J m^{-2} s^{-1}
UVDIR	flux direct UV	J m^{-2} s^{-1}
UVDIRO	direct UV outside greenhouse	J m^{-2} s^{-1}
VC	maximal carboxylation rate	µmol m^{-2} s^{-1}
VCMAX	maximal carboxylation velocity	µmol CO_2 m^{-2} s^{-1}
VCMAX25	maximal carboxylation velocity at 25 ^oC	µmol CO_2 m^{-2} s^{-1}
VPAIR	vapour pressure greenhouse air	kPa
VPD	Vapour Pressure Deficit	kPa
VPDAIR	Vapour Pressure Deficit of greenhouse air	kPa
VPDLA	leaf-air VPD	kPa
VPDSURF	VPD at leaf surface	kPa
VPLEAF	saturated water vapour pressure at leaf temp.	kPa
WATCON	water content crop	g m^{-2}
WATCONI	initial water content crop	g m^{-2}
WATCONMAX	maximal water content crop	g m^{-2}
WLV	dry weight of leaves	g DM m^{-2}
WLVI	initial leaf dry weight of crop	g m^{-2}
WRT	dry weight of roots	g DM m^{-2}
WRTI	initial root dry weight of crop	g m^{-2}
WSO	dry weight of storage organs	g DM m^{-2}
WSOI	initial dry weight of storage organs	g m^{-2}
WST	dry weight of stems	g DM m^{-2}
WSTI	initial stem dry weight of crop	g m^{-2}