

Mario LAGONEGRO Vincent HULL

SIMULAZIONE DEI PROCESSI TROFICI DI AMBIENTI ACQUATICI

Manuale d'uso di un modello numerico con programmi

in FORTRAN

Quaderni GEAD - EQ n. 8

Trieste 1989

INTRODUZIONE.

Questo lavoro tratta di una nuova versione più aggiornata ed efficiente di un modello che era stato inizialmente costruito con presupposti limitati. Si trattava di un programma in GW-BASIC in grado di simulare i processi di crescita di due specie fitoplanctoniche di un sistema ecologico acquatico che utilizzavano azoto, fosforo e silice come nutrienti. L'idea di sviluppare questo modello era stata suggerita da un report dell'UNESCO (1983) prodotto per un workshop che aveva avuto per tema l'analisi quantitativa e la simulazione degli ecosistemi, tenutosi alla Stazione Zoologica di Napoli.

Si sa che l'appetito vien mangiando! una volta realizzato il programma con le due fitospecie, ci é sembrato ovvio introdurre, nel processo trofico del sistema, una specie zooplanctonica erbivora, cui fece subito seguito l'apparizione di un carnivoro. Questo modello più completo ed il relativo programma é stato ampiamente descritto in precedenza (Lagonegro e Hull, 1987; Hull e Lagonegro, 1988; Hull, Lagonegro & Puccia, 1988) e proposto come uno strumento semplice di introduzione alla simulazione e di analisi ambientale di ecosistemi di limitata complessità, ma includenti le interazioni fondamentali, per gli idrobiologi, ma anche per studenti e laureandi afferenti al campo delle scienze biologiche e sociali.

Notoriamente il GW-BASIC, utilizzato anche in forma compilata, non si distingue per velocità di calcolo ed efficienza di programmazione numerica. Il tempo speso in lunghe attese per simulazioni, che comprendevano periodi di mesi, ci sembrò sempre più sprecato, così che decidemmo di passare il modello al FORTRAN (Hull, Lagonegro & Falcucci, in stampa). Il numero delle variabili biologiche fu svincolato dal limite numerico (basta intervenire nel dimensionamento di vettori e matrici), le zoospecie potevano essere erbivore, carnivore od onnivore (a scelta dell'utente che le poteva definire intervenendo nei valori di opportune matrici di efficienza di grazing e predazione), comparve l'ossigeno con alcune delle sue intricate vie di flusso nel sistema ecologico, la simulazione del ciclo notte-giorno in relazione all'intensità luminosa, i modi in cui i nutrienti vengono immessi nel sistema ecologico diventarono tre (al posto di due), come nel modello più piccolo.

A questo punto diventava necessario fornire un manuale che aiutasse l'utente a districarsi con questo nuovo strumento. Dato che poi i programmi ausiliari a quello del modello, già esaurientemente descritti in Lagonegro e Hull (1987), avevano subito alcuni modesti aggiustamenti nelle prestazioni (ma non assolutamente nel concetto), ci è sembrato opportuno riportarli in appendice, assieme al nuovo listato del modello, intendendo così fornire all' eventuale utilizzatore, un kit completo ed autoconsistente di programmi, anche a coloro che non disponessero del vecchio modello.

Delle versioni precedenti del programma si usava indicare l'hardware minimo necessario per gestirli; in questo caso possiamo solo informare che, se è pur possibile utilizzare tutto il software sviluppato con un micro computer (con almeno 512 Kbyte di memoria RAM, sistema operativo MS-DOS e un disco rigido), il tempo per correrlo è assolutamente insostenibile, anche per le utenze più pazienti e che sarà, quindi, meglio poter disporre di un mini computer tipo VAX o consimile, con il quale il FORTRAN del sistema operativo MS-DOS è perfettamente compatibile.

Il modello è in grado di effettuare le simulazioni anche per periodi superiori all'anno, in quanto riutilizza parametri e matrici della base di dati creata con AQINPUT5 per continuare i calcoli che esegue, modificando automaticamente i valori di partenza delle variabili ecologiche, tuttavia la vocazione di un modello come AQUAMOD è di operare specialmente su periodi di più breve durata, intendendosi per breve durata anche solo alcuni giorni.

Si ricorda infine che si potranno ottenere copie di tutti i programmi su supporto magnetico, facendone richiesta agli autori, inviando un dischetto intonso da 5.25 pollici.

IL MODELLO.

Il modello numerico di simulazione AQUAMOD5 fa capo all'approccio **riduzionistico** di studio degli ecosistemi, ovvero ogni componente del sistema naturale è analizzato singolarmente e messo poi in relazione agli altri, meccanicisticamente. Come i pezzi di un puzzle rendono conto del disegno d'insieme solo se sono stati tutti assemblati, così il modello riproduce la realtà (la simula) imitando l'effetto che ogni singolo componente, o fattore, esercita su tutte le variabili ecologiche. In questo senso è da considerarsi anche **deterministico**, in quanto calcola esattamente i valori che predice. Si può, infine, aggiungere che è un modello di tipo **dinamico** in quanto le variabili del sistema risultano tutte funzione del tempo.

AQUAMOD5 è stato sviluppato per simulare la crescita del plancton vegetale ed animale di un ambiente acquatico; si sono, pertanto, analizzate le diverse componenti dei sistemi acquatici che influenzano la crescita planctonica (temperatura delle acque, radiazione luminosa, disponibilità di nutrimento, abbondanza dei predatori, tassi di mortalità e respirazione, ecc.). Sono stati definiti in dettaglio i diversi tipi di relazione tra componenti considerati e crescita planctonica, quindi, si è proceduto alla traduzione di tali relazioni in linguaggio formale (matematico). Si è poi eseguito l'assemblaggio di tutte le trascrizioni matematiche delle relazioni in equazioni differenziali, che sono risolte dall'elaboratore con il programma AQUAMOD5, rispetto ad un fattore di incremento temporale (metodo step-wise). Il risultato del calcolo delle equazioni differenziali è un nuovo valore simulato della crescita planctonica.

STRUTTURA DEL PROGRAMMA AQUAMOD5.

Il programma é strutturato secondo una logica tradizionale, con un main program che chiama le subroutines specializzate, le quali a loro volta attivano, ove necessario, funzioni particolari di calcolo, come quella che è stata sviluppata per valutare le cinetiche o quella che riproduce l'effetto del ciclo notte-giorno. Una subroutine particolare, inoltre, provvede ad inizializzare cautelativamente le aree di memoria dedicate alle variabili.

Tutte le variabili importanti sono dimensionate in COMMON che portano una speciale etichetta, il cui nome chiarisce subito in quale settore del sistema esse giocano il loro ruolo. Essi sono:

/VEGET/... per le grandezze relative alle fitospecie ed alle funzioni loro caratteristiche;

/BESTIE/... per le analoghe grandezze relative alle zoospecie;

/DEPTH/... per le quantità tipo temperature e biomasse totali, cioè per le quantità che possono interessare in modo generico sia le fito che le zoospecie;

/NUTR/... per le variabili che descrivono l'evoluzione dei nutrienti come azoto (ammoniacale e nitrico), fosforo (orto-fosfato) e silicio (silice reattiva);

/OSSI/... che raggruppa le analoghe variabili riferentesi all'ossigeno, il quale, per la peculiarità delle sue interazioni merita di avere un situazione distinta dagli altri elementi chimici;

/LUCE/... che raggruppa le variabili relative alle formule usate, in opzione, per il calcolo della intensità luminosa;

/PARAMS/... che raggruppa una miscellanea di parametri che pilotano i loops di calcolo e l'assegnazione delle dosi di nutrienti e dei prelievi per consumi di altro genere.

Questi COMMON appaiono nelle subroutines specifiche al loro settore e vengono usati anche per alleggerire il carico di parametri passati alle stesse, mediante lo statement CALL e renderlo, quindi, più leggibile all'utente che volesse adattare AQUAMOD5 a proprie esigenze.

Vediamo ora, blocco per blocco, cosa fanno ognuna delle parti del programma.

Main program AQMODL5.

1) Assegna, mediante i **COMMON** e le dichiarative, lo spazio di memoria necessario alle variabili.

2) Chiama la subroutine **PRESET**, che inizializza opportunamente (in genere azzerandole) le memorie di cui sopra. Essa è indispensabile nelle macchine che, all'atto del caricamento e lancio del programma, non effettuano un azzeramento della memoria che viene ad esso dedicata dai valori lasciati da altri programmi, che condividono la risorsa in multiprogrammazione. Naturalmente esistono comandi di sistema operativo che adempiono a questo compito, ma è più comodo e sicuro per l'utente non doverci pensare ogni volta che cambia macchina. In ogni caso il tempo speso è modestissimo, a malapena avvertibile in un micro, insignificante in un mainframe.

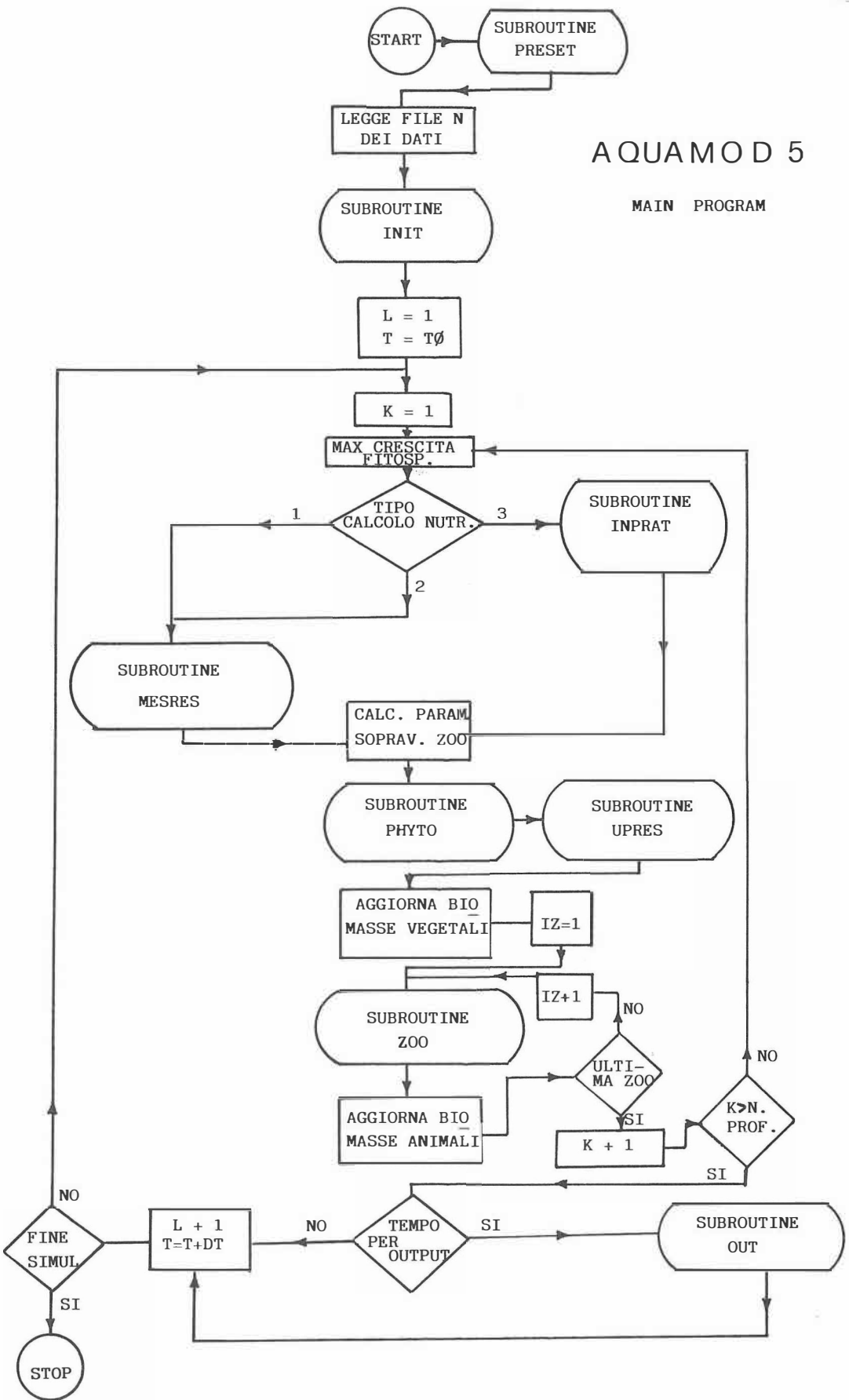
3) Chiede il nome del file su disco che contiene il piccolo data-base che è stato creato con un programma indipendente (vedi **AQINPT5** in appendice e Lagonegro e Hull (1987) per l'uso del programma), riportante tutti i dati necessari a far partire il lavoro di simulazione. Un altro filename richiesto è quello su cui si dovrà dirigere l'output del programma di simulazione. Si consiglia di lavorare con un disco rigido, non perchè il data-base sia grande ma semplicemente per il fatto che il programma del modello scriverà su disco molte informazioni ed un impiego ripetuto e persistente di un floppy lo porta presto fuori uso.

4) La subroutine **INIT** legge il data-base di cui sopra ed assegna i valori di partenza sia alle variabili biologiche che a quelle chimico-fisiche.

5) Comincia ora il loop sulla variabile tempo. Si parte da un valore iniziale **TEBASE** e si procede poi a passi **DT**, di incremento temporale eseguendo tutti i calcoli a ciascuna profondità prevista dall'utente, finchè non si sia superata la durata di simulazione definita e letta da **INIT**. Ad intervalli regolari, distanti un tempo pure letto in **INIT**, il programma stampa la informativa ed i valori delle grandezze biologiche e le scrive su di un file per usi ulteriori (plot e/o tabulazioni e/o riarrangiamento in qualche altro modo; per il plot esiste un programma ausiliario, **AQPLOT5** (vedi appendice), per gli altri usi l'utente dovrà scriversi autonomamente i programmi specifici). L'intero metodo è già stato descritto in dettaglio in precedenti lavori (Lagonegro & Hull, 1987; Hull e Lagonegro, 1988; Hull, Lagonegro & Falcucci, in stampa).

AQUAMOD 5

MAIN PROGRAM



6) Per ogni valore del tempo T, il calcolo di simulazione viene eseguito ad una o più profondità, a seconda delle richieste dell'utente contenute nel data-base di cui sopra. Le profondità dovranno, se due o più, apparire in ordine di valore crescente, a partire da quelle più vicine alla superficie, in quanto che l'evoluzione della biomassa vegetale alle profondità più superficiali condiziona, attraverso il coefficiente di estinzione, l'intensità della luce trasmessa a quelle più profonde.

7) A ciascuna delle profondità considerate, nel senso ora descritto, viene calcolato l'input dei nutrienti. Una opzione contenuta nel data-base condiziona il modo in cui il calcolo verrà eseguito, quindi, la subroutine chiamata in azione, MESRES oppure INPRAT. La prima verrà chiamata se:

(a) i dati di nutrienti in input nel data-base sono le concentrazioni effettivamente messe a disposizione delle fitospecie. Con esse si calcoleranno i valori delle cinetiche di assunzione, che permetteranno di definire il nutriente limitante, nel senso già descritto nei lavori prima citati, ovvero quello con il più basso rapporto (concentrazione/costante di semisaturazione).

(b) i dati esprimono i residui non utilizzati dalle fitospecie, che hanno già prelevato i loro fabbisogni sia singoli che complessivi; non si parla in questo caso di nutriente limitante e gli unici fattori influenti sulla crescita saranno la luce e la temperatura.

Si può dire che la differenza tra la situazione descritta in a) e quella descritta in b) è da ricercarsi nell'orario di prelievo dei campioni di nutrienti: nel caso a) ci si riferisce a prelievi eseguiti al mattino, mentre nel caso b) a prelievi serali.

La INPRAT verrà chiamata se:

(c) nel data-base si trovano i valori iniziali dei nutrienti, cioè i valori delle concentrazioni al tempo TEBASE, ed i ratei periodici di input relativi al tempo selezionato per simulazioni.

8) In base alla concentrazione di ossigeno presente nel sistema ed ai valori dei parametri contenuti nel data base vengono calcolati, per ogni zoospecie, un coefficiente di sopravvivenza respiratoria che serve a simulare gli effetti di eventuali anossie.

9) La subroutine PHYTO, usando i dati sui nutrienti e sulla luce, calcola i valori massimi delle funzioni di crescita delle fitospecie nelle date condizioni e, se non ci si trova nel caso (7b), indica anche il nutriente limitante.

10) Con questi valori delle funzioni di crescita la subroutine UPRES calcola il fabbisogno di nutrienti, lo confronta con la disponibilità e, al caso di insufficiente disponibilità, limita la crescita.

11) A questo punto il programma aggiorna le biomasse vegetali.

12) La subroutine ZOO calcola i fattori di crescita delle zoospecie.

13) Il programma aggiorna le biomasse animali, tenendo conto anche del coefficiente legato ad eventuali anossie.

14) Vengono ripetuti tutti i passi dal (7) al (13) per ciascuna delle ulteriori profondità prese in considerazione, se ve ne sono.

15) Se il valore del tempo è quello opportuno, si produrrà da parte della subroutine OUT la stampa della situazione al momento attuale (con analogia scrittura su disco di cui al passo (5)). Se l'intervallo tra due outputs è maggiore di 12 ore, ogni stampa e scrittura su file si compone di due blocchi, uno prodotto alle ore 12 del giorno in cui avviene l'output, l'altro alle 24. Questa caratteristica permette di seguire la presenza di un eventuale ciclo notte-giorno.

16) Si aggiorna il valore del tempo per un nuovo calcolo. Se questo valore del tempo eccede il limite previsto nel data-base, il programma si ferma, altrimenti torna a ripetere tutti i passi dal (6) in giù.

Subroutine INIT.

- 1) Legge dal data-base, che si trova su file (nome letto nel main program):
 - a) il minimo livello di luce per la acclimatazione (usato nelle opzioni STEELE e DI TORO del calcolo del fattore di crescita riferentesi all'azione limitante della luce);
 - b) la concentrazione di soglia alla quale le fitospecie cominciano a consumare (per sopravvenuta indisponibilità di azoto ammoniacale) anche azoto nitrico;
 - c) il tipo di input dei nutrienti al sistema (passo (7), paragrafo precedente);
 - d) un parametro usato nel calcolo della temperatura e dedotto con un programma ausiliario (AQTEMP5 vedi appendice) prima di fare il data-base;
 - e) la data di inizio della simulazione (mese e giorno).

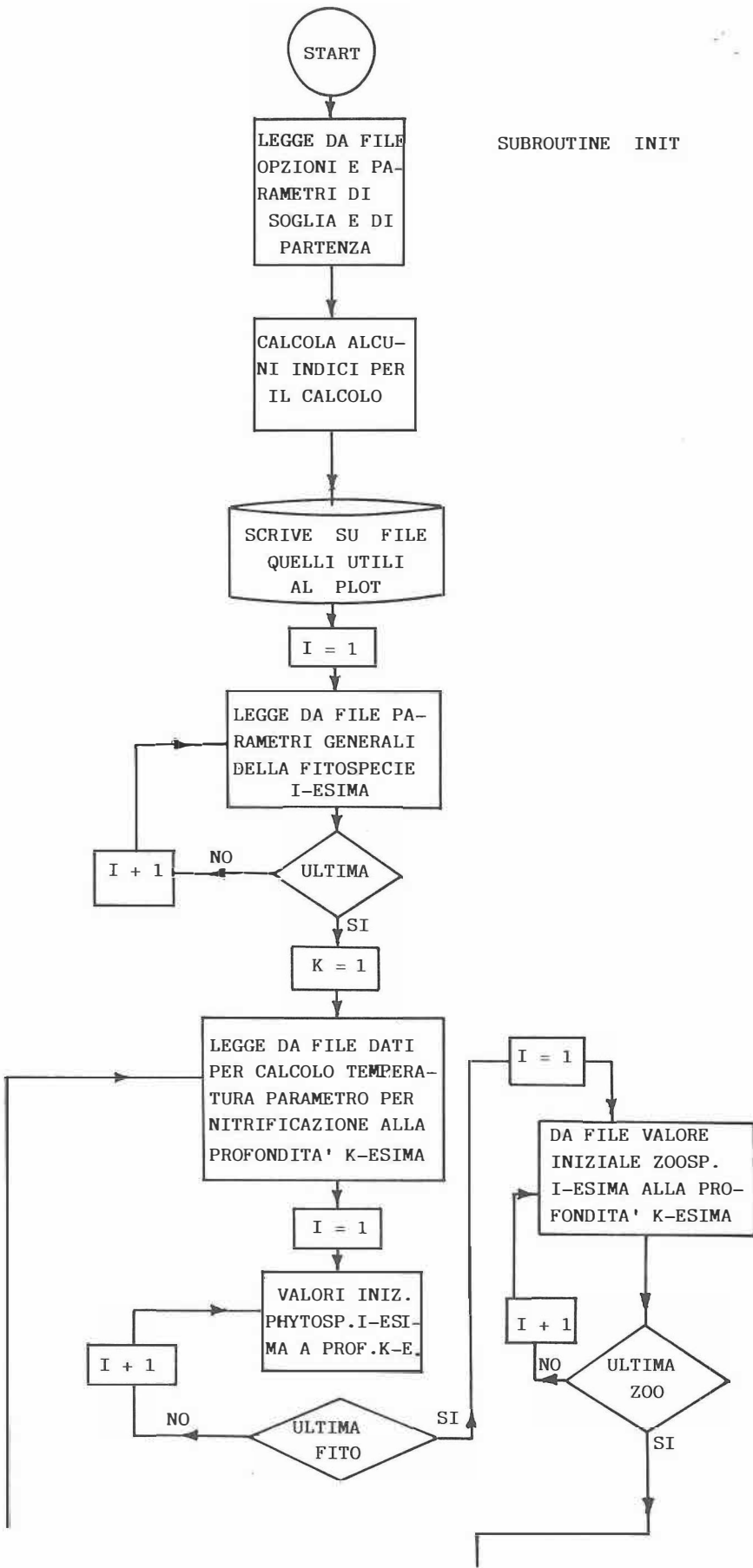
- 2) Calcola alcuni parametri per pilotare il main program (loops sul tempo e sulle profondità, periodicità della uscita delle stampe).

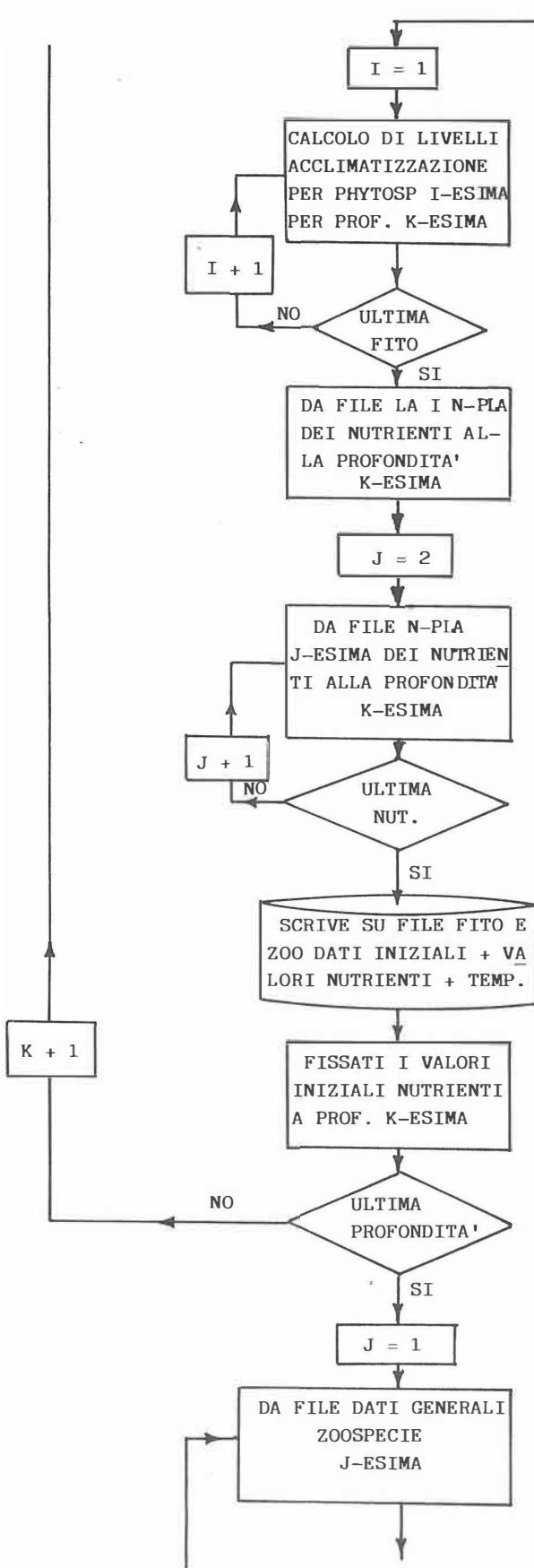
- 3) Scrive su file quelli che saranno usati dal programma ausiliario AQPLOT5.

- 4) Legge dal file di input i parametri generali delle fitospecie, quali:
 - a) ratei di respirazione e mortalità;
 - b) la costante di crescita; c) la temperatura corrispondente alla crescita massima;
 - d) il parametro di consumo dell'ossigeno per unità di biomassa vegetale persa per respirazione;
 - e) il parametro di produzione dell'ossigeno per unità di biomassa vegetale;

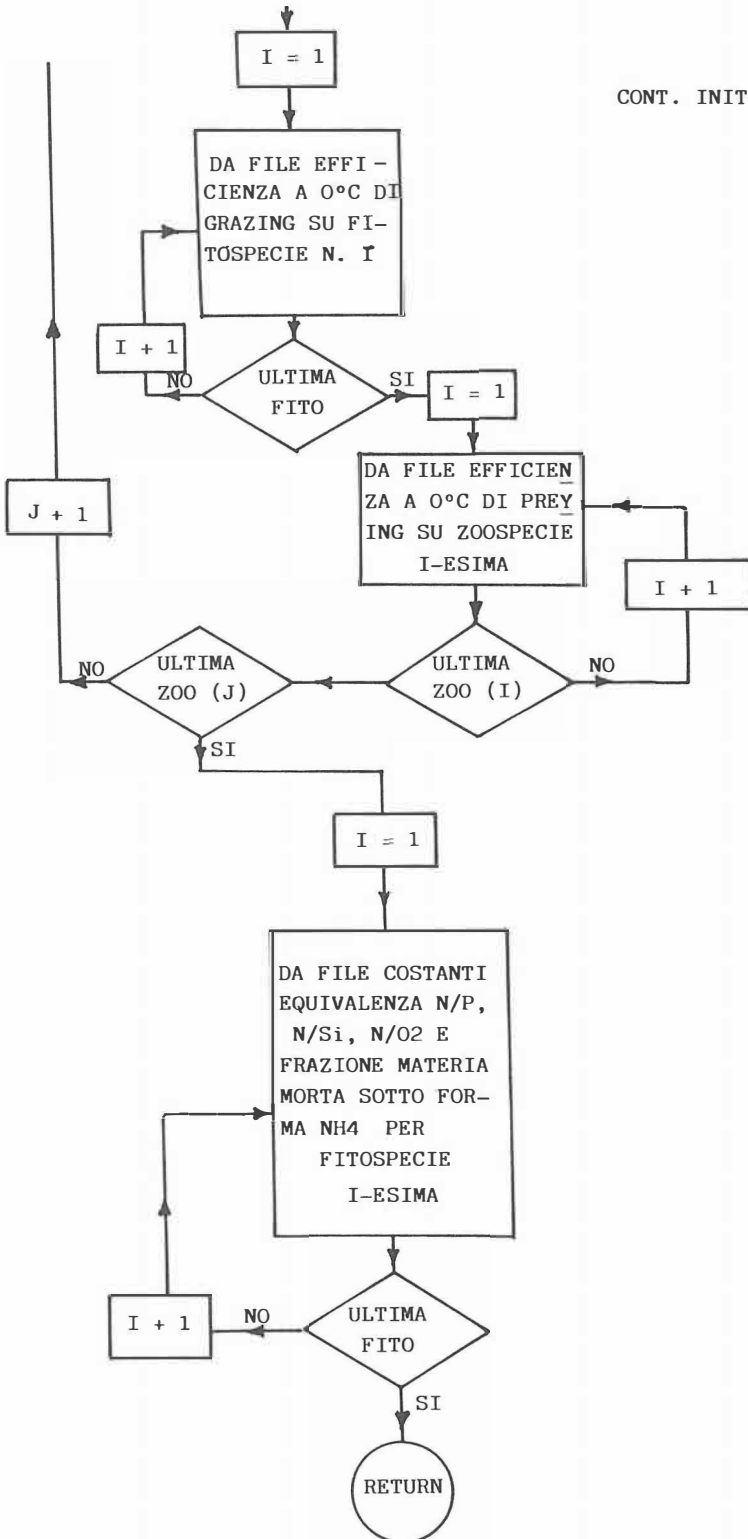
- 5) Inizia un loop di letture che definiscono le caratteristiche e concentrazioni iniziali per le fitospecie, le zoospecie e per i nutrienti alle singole profondità richieste. Verranno lette:

SUBROUTINE INIT





CONT. INIT



a) parametri per il calcolo della temperatura alle diverse profondità, dedotti mediante l'uso del programma ausiliario AQTEMP5; essi sono la temperatura media annua e la metà dello scarto tra il valore massimo e minimo;

b) parametro indicante il rapporto di NH₄ che viene ossidato a nitrato per unità di concentrazione di ossigeno presente alle diverse profondità;

c) valori iniziali delle biomasse vegetali alle diverse profondità;

d) analoghi valori per le biomasse animali;

e) prima n-ple di valori dei nutrienti più un rateo; nel dettaglio essi sono:

e1) giorno dell'anno cui è riferita la misura (la prima è TEBASE);

e2) concentrazioni iniziali (in mg/m³) di NH₄, NO₃, PO₄, SI, O₂;

e3) frazione di O₂ diretta agli usi del sedimento.

f) Tutte le altre n-ple di valori per i nutrienti ed il rateo:

f1) giorno dell'anno cui è riferita la misura;

f2a) concentrazioni misurate o concentrazioni residue (vedi passo (7) del main program, opzioni (a) e (b)) in mg/m³;

f2b) ratei di input (in (mg/m³)/giorno) (opzione (c));

f3) frazione di O₂ diretta agli usi del sedimento.

All'interno del loop, la subroutine calcola autonomamente i valori iniziali della intensità di acclimatazione della luce a ciascuna delle profondità considerate e scrive poi sul file di output per il programma di grafica.

6) Dal data-base vengono letti dati generali caratteristici delle singole zoospecie, quali:

a) costante di semi-saturazione per il grazing (sarà nulla per i carnivori puri);

b) costante di semi-saturazione per la predazione (sarà nulla per gli erbivori puri);

c) rateo di mortalità e rateo di respirazione a 0 gradi;

d) il parametro di consumo dell'ossigeno per unità di biomassa animale persa in respirazione;

e) la frazione di materia morta animale che torna al sistema sotto forma di ammoniaca;

f) la costante di semi-saturazione relativa alla concentrazione di O₂ necessaria alla sopravvivenza del 50% della zoospecie.

7) Nuovamente, per ciascuna delle zoospecie, legge: a) l'efficienza di grazing sulle fitospecie (sarà nulla per i carnivori puri);

b) quella di predazione sulle zoospecie (valutata a 0 gradi), compresa la specie medesima (sarà nulla per gli erbivori puri, ma consentirà di considerare anche il fenomeno di cannibalismo).

Durante la simulazione questa efficienza a 0 gradi sarà moltiplicata per un fattore che terrà conto della temperatura effettiva al momento del calcolo.

8) Per ciascuna delle fitospecie, legge dal data-base :

a) i rapporti che indicano quanto fosforo, silice, ossigeno vengono consumati per consumo unitario di azoto;

b) la frazione di materia morta vegetale che torna al sistema sotto forma di ammoniaca.

Subroutine MESRES.

Nei dati letti in INIT sono compresi i valori dei nutrienti per ogni misura effettuata nell'arco di tempo che copre il periodo di simulazione prescelto (max. 60 valori), a partire dal gruppo di valori al tempo TEBASE. La subroutine MESRES entra in azione quando anche le altre n-ple di dati sui nutrienti sono concentrazioni, esattamente come i valori iniziali. Queste concentrazioni possono essere poi input effettivi al sistema ecologico simulato, oppure residui lasciati dalle fitospecie.

Dato che il passo di integrazione DT delle equazioni nel modello é di solito molto più piccolo della distanza temporale tra due n-ple, si richiede un lavoro di interpolazione per avere il valore dei nutrienti ad un qualsiasi tempo T intermedio; anzi, la subroutine calcola il valore medio nell'intervallo ampio DT che inizia in T. Vediamo come (Fig. 1 A); siano:

T, T+DT estremi dell'intervallo in questione;

T(k), T(k+1) valori contigui del tempo in due n-ple di valori

in input per i nutrienti, con:

$$T(k) \leq T, T+DT \leq T(k+1) ;$$

NUT(k), NUT(k+1) corrispondenti valori del generico nutriente;

Allora avremo che:

$$NUT(T) = NUT(k) + ((T - T(k)) / (T(k+1) - T(k))) * (NUT(k+1) - NUT(k))$$

$$NUT(T+DT) = NUT(k) + ((T+DT - T(k)) / (T(k+1) - T(k))) * (NUT(k+1) - NUT(k))$$

$NUTC(T) = (NUT(T+DT) + NUT(T)) / 2$ sarà il valore della concentrazione del nutriente che verrà utilizzata nel calcolo al tempo T.

La subroutine esegue, ovviamente, questo calcolo per ognuno dei nutrienti, fornendo così i valori degli stessi, da utilizzare con le due ipotesi viste prima per quel che riguarda i flussi dall'esterno. A questi valori dovranno poi sommarsi eventualmente i flussi interni al sistema, positivamente se si aggiungono alle quantità già preesistenti, negativamente in caso contrario, fornendo alla fine gli stocks aggiornati; ciò avviene, effettivamente, nella subroutine UPRES.

SUBROUTINE MESRES

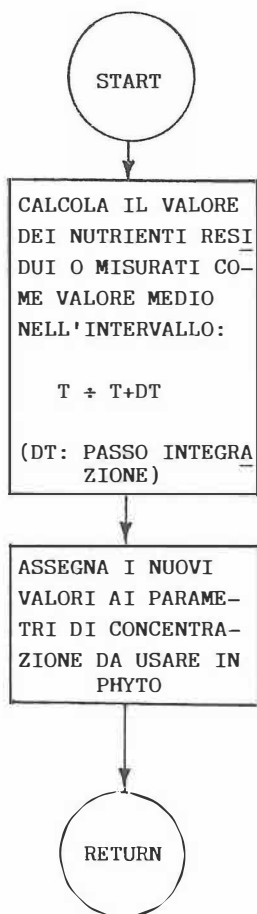
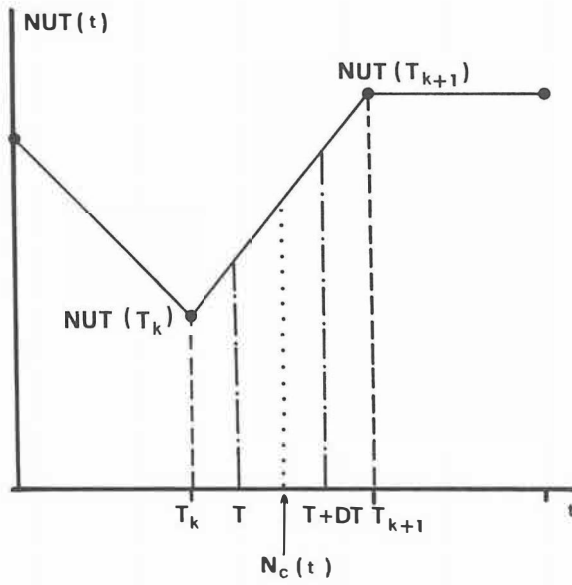
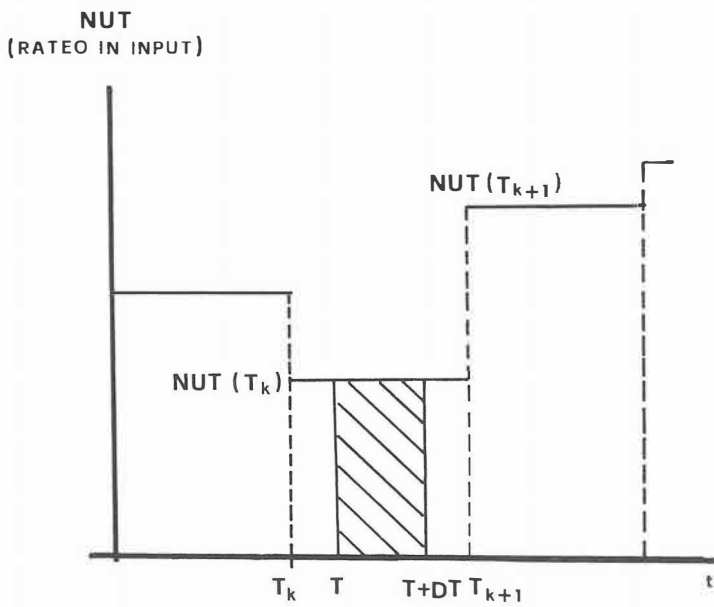


FIGURA 1



A



B

Subroutine INPRAT.

Questa subroutine viene chiamata in azione nel caso in cui le n-ple dei nutrienti successive alla prima (che riporta, come nel caso precedente, i valori iniziali) siano invece i ratei giornalieri di input, così che il calcolo delle quantità da aggiungere ai residui, se ve ne sono, è il seguente (Fig. 1 B):

$$\text{NUT}(T+DT)=\text{NUT}(T)+\text{NUT}(k)*DT$$

dove:

$\text{NUT}(T)$ è il residuo lasciato da UPRES al passo precedente;

$\text{NUT}(k)$ è la velocità con cui il dato nutriente entra nel si-

stema nel periodo da $T(k)$ a $T(k+1)$, in $((\text{mg}/\text{m}^3)/\text{giorno})$;

$T, T+DT, T(k), T(k+1)$ come nel caso precedente.

Questa immagine sarà tanto più realistica quanto più numerosi saranno i dati di riferimento, cioè le n-ple successive alla prima. Sarà cura dell'utente programmare le osservazioni sperimentali in modo regolare e ben intervallati, eventualmente infittendoli nei periodi di maggior variabilità, senza per questo però appesantire il data-base letto da INIT con un numero eccessivo di n-ple.



Subroutine PHYTO

Questa subroutine calcola i fattori di crescita delle singole fitospecie, tenendo conto dei fattori limitazionali della luce e dei nutrienti, fatta salva l'ipotesi che i nutrienti esprimano valori residui, nel qual caso esiste una piena disponibilità in ogni momento del fabbisogno. Se ci si trova in questa situazione solo il fattore luminoso viene effettivamente calcolato in PHYTO, mentre l'effetto della temperatura al tempo T è già stato precedentemente incluso nel valore della massima intensità di crescita precalcolato nel main program. Quindi, per ciascuna delle fitospecie:

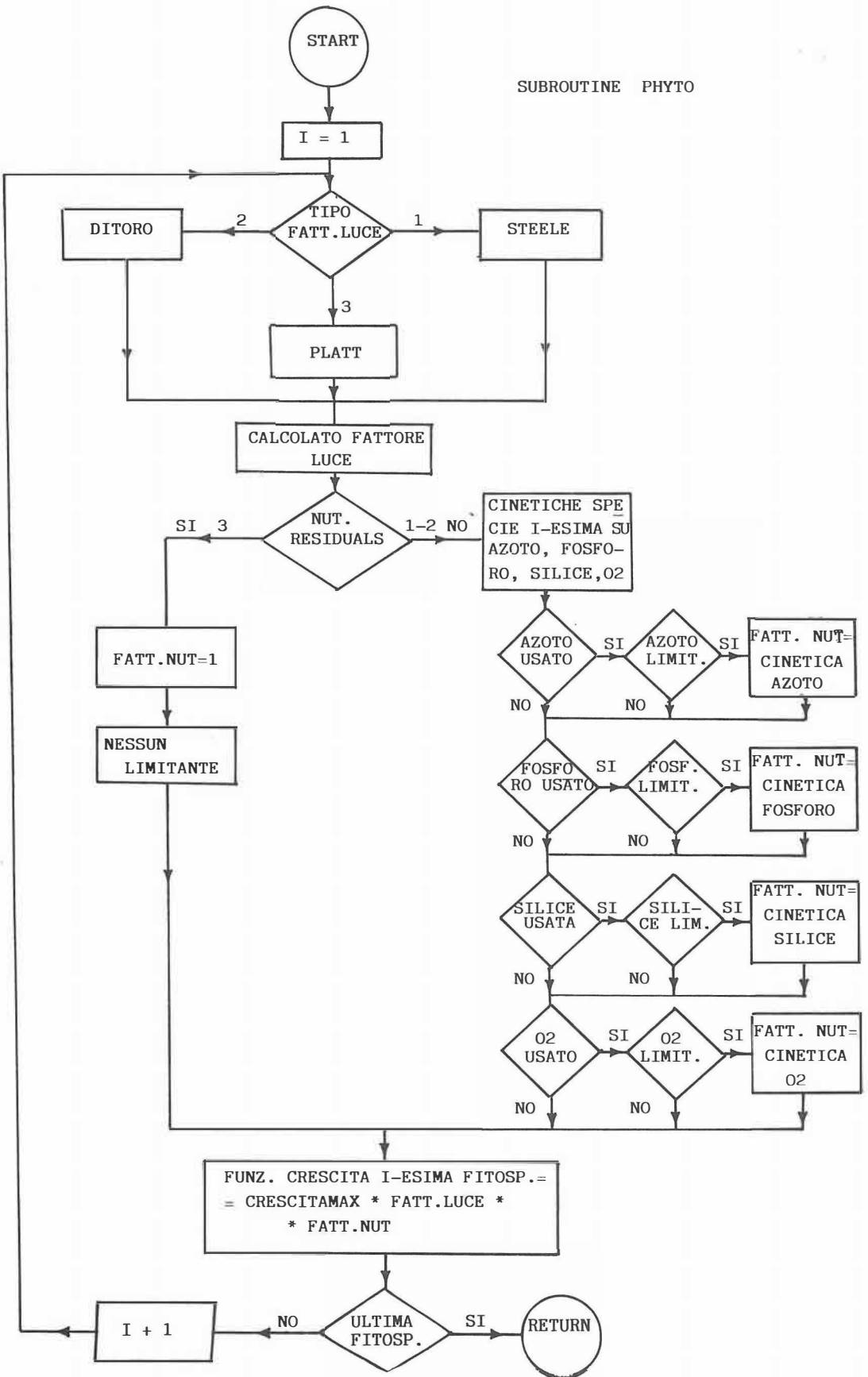
1) in base all'opzione letta da INIT calcola il fattore di crescita dovuto alla luce; le opzioni possibili sono date dalle formule di STEELE (Steele, 1962), di DI TORO (Di Toro et al. 1977), di PLATT (Platt et al. 1982).

2) Nelle due ipotesi che i dati sui nutrienti siano o concentrazioni misurate in input, oppure concentrazioni aggiornate con i ratei di input, la subroutine calcola le cinetiche per ciascun nutriente, scegliendo come "limitante" quello con la cinetica di valore minore. Naturalmente, ove si trattasse di un nutriente ininfluenza per la particolare fitospecie, l'utente avrà posto uguale a 0 nel data-base letto da INIT la corrispondente costante di semisaturazione.

Nel caso in cui i dati dei nutrienti siano residui lasciati dalle fitospecie dopo il soddisfacimento dei fabbisogni, nessuno di loro viene considerato limitante, perciò non influenza il valore del fattore di crescita.

3) Alla fine, per ciascuna fitospecie, si avrà il valore della funzione di crescita massima permessa da temperatura, luce e nutrienti alla data profondità ed al tempo T .

SUBROUTINE PHYTO



Subroutine UPRES.

La subroutine calcola i valori complessivi degli stocks di nutrienti, inclusi i flussi interni al sistema, valuta le necessità di uptake delle fitospecie, rapporta queste ultime alla disponibilità, salvo che nell'ipotesi già vista dei residui, e, sempre con la medesima riserva, stabilisce l'eventuale limitazione alla crescita. I singoli passi sono:

1) Calcola l'ossigeno prodotto per fotosintesi con la formula seguente, espressa come loop di programma:

FOTOS=0 la variabile esprime il valore dell'ossigeno prodotto

for I=1 to NFITO do

begin

if DF(I) > 0 then FOTOS=FOTOS+FTE(I)*FITO(I,K)*DF(I)

end

dove:

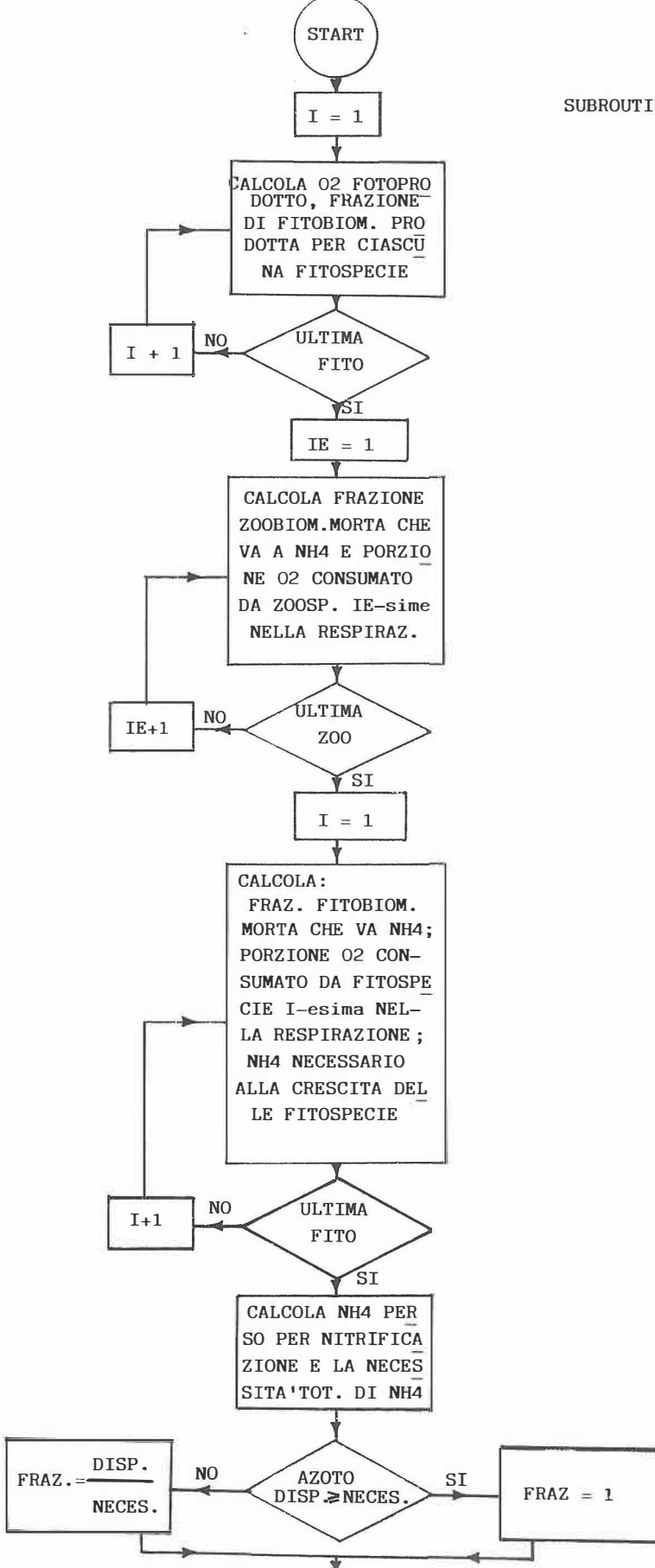
NFITO é in numero di fitospecie nel sistema;

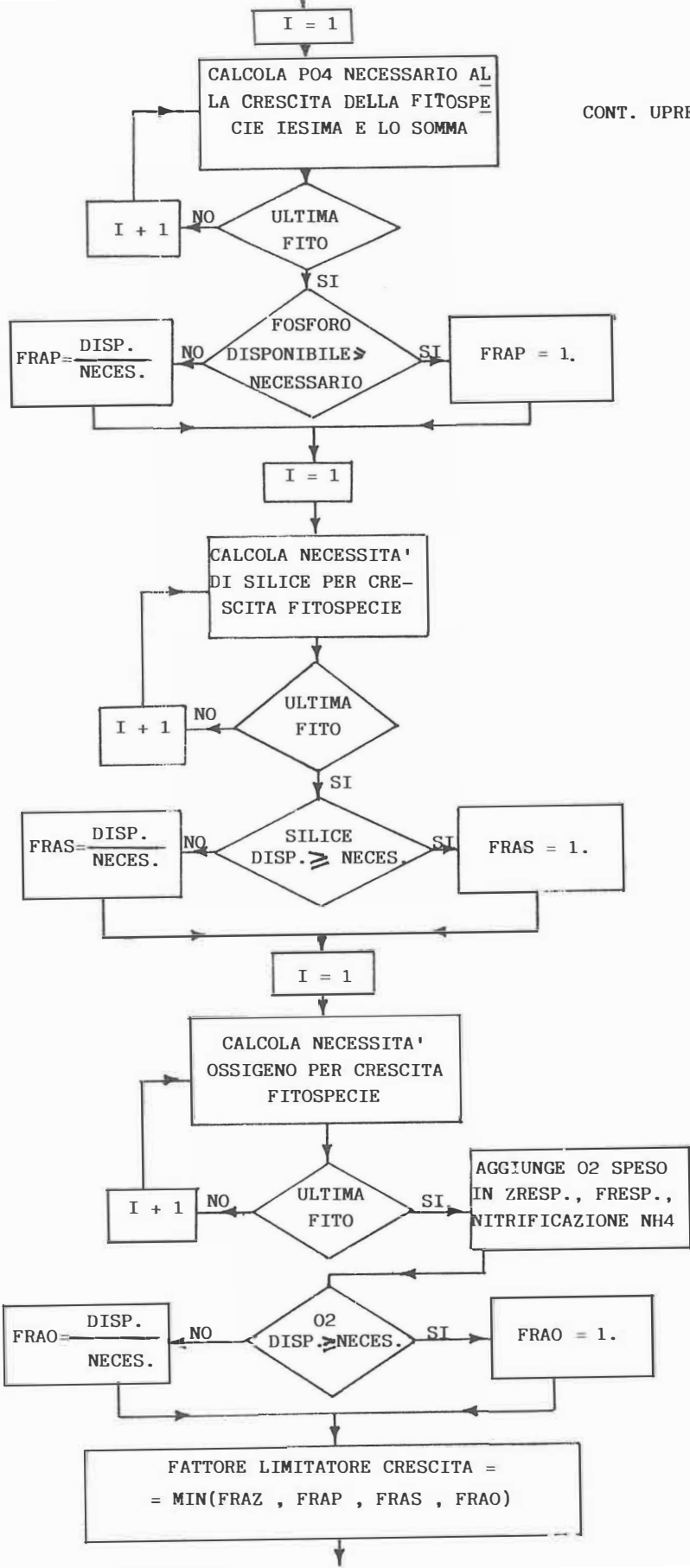
FTE(I) é l'ossigeno prodotto per unità di biomassa dalla fitospecie i-esima;

FITO(I,K) é la biomassa della fitospecie i-esima presente al tempo T alla profondità k-esima;

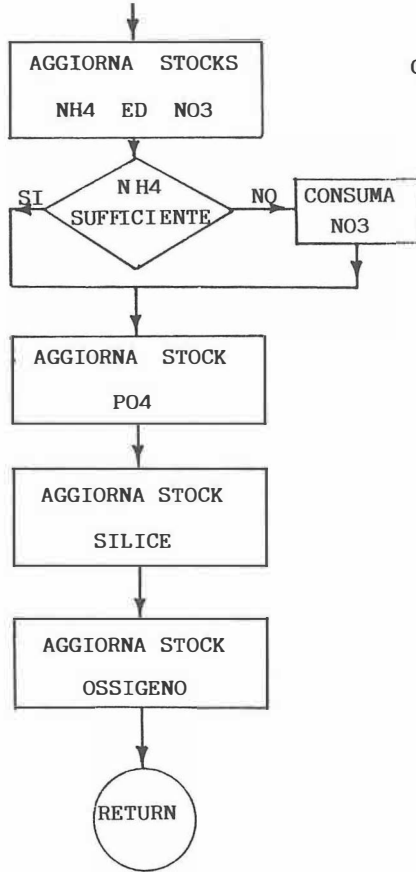
DF(I) é l'attuale ritmo di crescita, corretto per mortalità e respirazione, della fitospecie i-esima. Nella somma

DF(I) viene usata col suo valore se positiva, con valore nullo se negativa, dato che il consumo di O₂ viene valutato in altra parte;





CONT. UPRES



2) Calcola la quantità di biomassa animale morta che ritorna al sistema sotto forma di azoto ammoniacale alla profondità k-esima al tempo T. Analogamente calcola l'ossigeno consumato durante la respirazione dalle zoospecie. Il loop é:

DEMA = 0 la variabile esprime la frazione di materia morta complessiva che viene riconvertita in NH₄;

ZRES = 0 la variabile esprime l'ossigeno complessivamente consumato dalle zoospecie al tempo T ed alla profondità k-esima;

for I=1 to NZOO do

begin DEMA=DEMA+ZNH(I)*ZOO(I,K)*MORTZ(I);

FRESP(TEMP(K))=FRESP(0)*EXP(RZ*TEMP(K));

ZRES=ZRES+ZRE(I)*ZOO(I,K)*FRESP(TEMP(K)) end

dove:

NZOO é il numero di zoospecie presenti;

ZNH(I) é l'NH₄ prodotto per unità di biomassa morta per lazoospecie i-esima;

ZOO(I,K) é la biomassa della zoospecie i-esima alla profondità k-esima;

MORTZ(I) é la mortalità della zoospecie i-esima;

TEMP(K) é la temperatura al tempo T alla profondità k-esima;

FRESP(0),RZ sono l'efficienza di respirazione a 0 gradi ed il fattore di crescita della efficienza con la temperatura, naturalmente per la specie i-esima;

ZRE(I) é l'ossigeno consumato per unità di biomassa spesa inrespirazione dalla zoospecie i-esima;

3) Calcoli analoghi a quelli del passo (2) vengono ora eseguiti per le fitospecie, assieme alla valutazione delle necessità di NH₄ da parte delle medesime. Il loop é:

DEMA=DEMA(ZOO) la variabile esprime la quantità vista al passo (2);

FRES = 0 la variabile esprime l'ossigeno consumato dalle fitospecie nella respirazione;

DNH4 = 0 esprime il fabbisogno di azoto ammoniacale delle fitospecie;

for i=1 to NFITO do

begin DEMA=DEMA+FNH(I)*FITO(I,K)*MORTF(I);

FRES=FRES+FRE(I)*FITO(I,K)*RE(I);

if FC(I) > 0 then DNH4=DNH4+FC(I)*FITO(I,K)

end

dove:

FNH(I) é la quantità di NH4 prodotto per unità di biomassa vegetale morta della fitospecie i-esima;

MORTF(I) é la mortalità della fitospecie i-esima;

FRE(I) é l'ossigeno consumato per unità di fito-biomassaspesa nella respirazione dalla fitospecie i-esima;

RE(I) é il rateo di respirazione per la fitospecie i-esima;

FC(I) é la funzione di crescita, corretta per mortalità e respirazione, della fitospecie i-esima. Essa viene presa col suo valore se positiva, altrimenti nulla.

4) Al consumo di azoto ammoniacale previsto per le fitospecie viene aggiunta la quantità persa nei processi di ossidazione dell'ammoniaca a nitrato al tempo T, profondità k-esima.

5) Confronta la necessità di azoto con la disponibilità (NH4+NO3); se questa é adeguata, assegna ad un parametro FRAZ il valore di 1, altrimenti lo pone pari al valore del rapporto tra la disponibilità e la necessità, perciò minore di 1.

6) Calcola il fabbisogno di PO₄ delle fitospecie, con il loop:

DPO₄ = 0 la variabile esprime il fabbisogno di fosforo;

for i=1 to NFITO do

begin

if FC(I) > 0 then DPO₄=DPO₄+FC(I)*FITO(I,K)*FNP(I);

end

dove FNP(I) é la quantità simultanea di fosforo assimilata per unità di azoto dalla fitospecie i-esima.

Il confronto tra DPO₄ e la disponibilità di fosforo permette ora di dare un valore alla quantità FRAP, analoga alla FRAZ del passo (5).

7) Analogo calcolo viene fatto per la silice. Si ha:

DSIL = 0 la variabile esprime il fabbisogno di silice delle fitospecie alla profondità k-esima;

for I=1 to NFITO do

begin

if FC(I) > 0 then DSIL=DSIL+FC(I)*FITO(I,K)*FNSI(I)

end

dove FNSI(I) é la quantità simultanea di silice assimilata per unità di azoto dalla fitospecie i-esima.

Anche in questo caso viene ora assegnato un valore ad un indice, FRAS, analogo a FRAZ e FRAP.

8) Il calcolo per l'ossigeno é all'inizio simile ai precedenti. Si ha il loop:

DO₂ = 0 la variabile descrive il consumo di ossigeno nei vari processi;

for I=1 to NFITO do

begin

if FC(I) > 0 then DO2=DO2+FC(I)*FITO(I,K)*FNO(I)

end

dove FNO(I) rappresenta la quantità di ossigeno consumato simultaneamente, nei processi di crescita, per unità di azoto dalle fitospecie.

A questa quota vengono sommati i contributi ZRES e FRES relativi alla respirazione delle zoo e fitospecie, nonché la quota di ossigeno spesa nei processi di ossidazione dell'ammoniaca.

Completato il calcolo, si effettua il solito confronto fra il fabbisogno e la disponibilità e si assegna il valore al parametro FRAO, analogo ai precedenti.

9) Sceglie il minore tra i fattori FRAZ, FRAP, FRAS e FRAO come fattore limitante la crescita (FLC), rendendola proporzionale alla disponibilità del meno abbondante, rispetto al fabbisogno, dei nutrienti considerati. Nell'ipotesi dei residui o nel caso di sufficiente disponibilità questo fattore vale 1, il che equivale a dire che non si ha limitazione alla crescita dovuta ai nutrienti.

10) A questo punto vengono aggiornati gli stocks dei nutrienti, tenendo anche conto del fattore FLC, che sarà minore od al massimo uguale ad 1. Perciò si ha:

$NH_4(\text{dopo}) = NH_4(\text{prima}) - FLC * DNH_4 - FRAO * NH_4(\text{prima}) + DEMA$

se NH_4 risulta inferiore alla soglia minima letta in INIT, inizia l'assimilazione anche dell'azoto nitrico:

$NO_3(\text{dopo}) = NO_3(\text{prima}) + FRAO * NH_4(\text{prima}) - (\text{eventuale consumo})$

$PO_4(\text{dopo}) = PO_4(\text{prima}) - FLC * DPO_4$

$SILICE(\text{dopo}) = SILICE(\text{prima}) - FLC * DSIL$

$O_2(\text{dopo}) = O_2(\text{prima}) + FOTOS - FRAO * (ZRES + FRES + O_2(\text{ossidaz.}))$

Nel caso che FLC sia minore di 1, ci potranno essere dei residui per i meno limitanti dei nutrienti, ma non per quello che ha fornito il valore di FLC. Può darsi che FLC sia casualmente generato da FRAO ed avere, perciò, lo stesso valore di FRAO, ma bisogna ricordare che i due coefficienti agiscono su traccianti diversi di processo.

Naturalmente questo passo (10) non viene effettuato se i dati sui nutrienti si riferiscono a residui lasciati dalle fitospecie che hanno completamente soddisfatto i loro fabbisogni.

Subroutine ZOO.

Questa subroutine calcola i fattori di crescita per le zoospecie; è inserita in un loop e così tratta una zoospecie alla volta. Vediamo i passi:

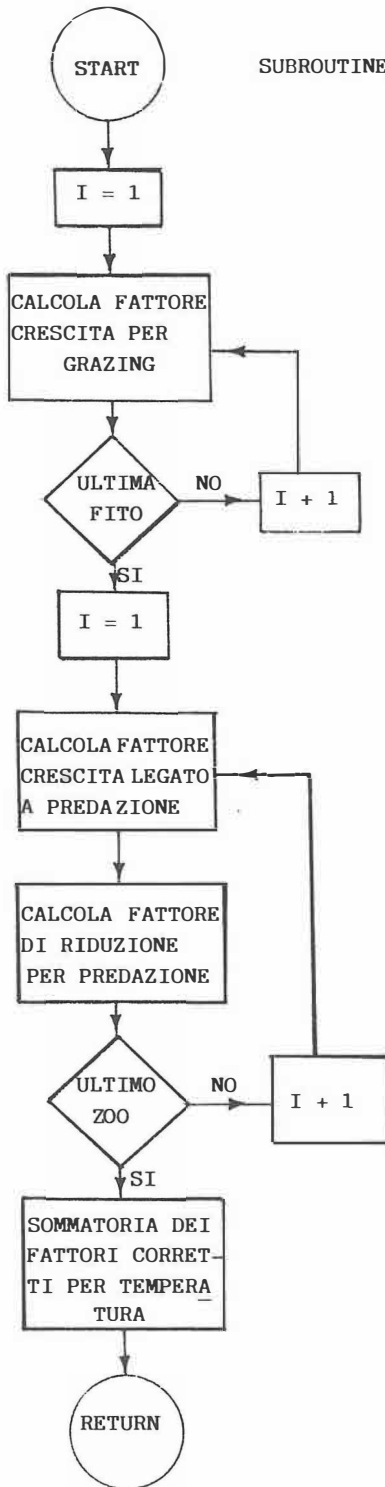
1) Calcola il contributo alla crescita dato da ciascuna delle fitospecie soggette a pastura dalla zoospecie considerata. Poichè nel data-base letto da INIT compare la efficienza di grazing di ogni zoospecie su ogni fitospecie, è possibile escludere per una data zoospecie quelle fitospecie che non fanno parte della sua dieta semplicemente ponendo uguale a 0 la corrispondente efficienza di grazing nei loro riguardi. Chiaramente, le efficienze saranno tutte nulle per i carnivori puri.

2) Calcola il contributo alla crescita dovuto alla predazione su tutte le zoospecie. In base alle efficienze di predazione lette dal data-base è possibile escludere od includere il cannibalismo così come le preferenze dietetiche. Per gli erbivori puri il contributo sarà ovviamente nullo, essendo tutte nulle le efficienze di predazione.

3) Calcola di quanto la zoospecie viene ridotta per effetto della predazione che le altre esercitano su di essa.

4) I vari fattori vengono corretti per gli effetti legati alla temperatura e poi sommati.

SUBROUTINE ZOO



Subroutine OUT.

La subroutine OUT provvede alla stampa delle grandezze chimico-fisiche e biologiche agli istanti di tempo prefissati con un parametro letto da INIT. I passi sono ripetuti per ciascuna delle profondità:

- 1) Stampa data, ora, profondità, temperatura, coefficiente di estinzione della luce, valore di FLC;
- 2) Per tutte le fitospecie, stampa il valore della funzione effettiva di crescita (corretta cioè dei fattori non puramente nutrizionali) ed il nome del nutriente limitante, se non si lavora nell'ipotesi dei residui come dati in input dal data-base.
- 3) In questa ultima ipotesi, stampa i consumi di azoto, fosforo, silice ed ossigeno da parte delle fitospecie. In caso valgano le ipotesi dei dati come concentrazioni misurate in input o ratei di input, allora stampa i valori dei residui lasciati dopo l'uptake più o meno limitato da FLC.
- 4) Stampa la biomassa e la sua velocità di variazione (si ha crescita, ovvero aumento di biomassa se sarà positiva, diminuzione se negativa) al momento attuale per ciascuna delle fitospecie.
- 5) Stampa biomassa e fattore limitante per la presenza di ossigeno per ciascuna delle zoospecie.
- 6) Alla fine scrive sul file di output tutte le informazioni necessarie per la grafica o per altri usi.

START

SUBROUTINE OUT

STAMPA DATA, PROFON-
DITA', TEMPERATURA,
COEFF. ESTINZ., F.
L. CRESCITA

I = 1

STAMPA FUNZ. DI CRE-
SCITA E NUTR. LIMI-
TANTE PER FITOSPECIE
I-ESIMA

ULTI-
MA FITO

NO

SI

STAMPA I RESIDUI DI
N, P, Si, O2

NO

SI

STAMPA I CONSUMI DI
N, P, Si, O2

I = 1

STAMPA BIOMASSA E
SUA VARIAZIONE ISTAN-
TANEA PER FITOSPECIE
I-ESIMA

ULTIMA
FITO

NO

SI

I + 1

I = 1

STAMPA BIOMASSA E
FATTORE O2 PER ZOO-
SPECIE I-ESIMA

ULTIMA
ZOO-
SP.

NO

SI

I + 1

SCRIVE SU FILE
PER PLOT DATI FI-
TO E ZOOSPECIE
E NUTRIENTI

RETURN

Funzione DAYL.

Questa funzione simula il ciclo notte-giorno con una formula dedotta da dati sperimentali; si tratta comunque di una funzione armonica, come a quella che simula l'intensità luminosa durante l'anno (Lagonegro & Hull, 1987). I passi sono:

1) Calcola la durata in ore della luce diurna, con la relazione:

$$\text{DURATALUCE} = 24 * \text{FOTOP} + 3$$

dove:

FOTOP é il fotoperiodo al tempo T durante l'anno, pure luicalcolato con una funzione periodica (Kremer e Nixon, 1978).

Le tre ore aggiunte sono il periodo di luce indiretta diffusa, metà all'alba e metà al tramonto.

2) Calcola le ore di inizio e fine della illuminazione, sia diretta che indiretta, con le relazioni:

$$\text{ALBA} = 13 - 0.5 * \text{DURATALUCE}$$

$$\text{TRAMONTO} = 13 + 0.5 * \text{DURATALUCE}$$

L'intervallo viene centrato sulle ore 13, come dedotto dai dati sperimentali.

3) Calcola l'intensità relativa della luce al tempo T durante la giornata con le relazioni:

$$\text{DAYL} = 0 \text{ se } T < \text{ALBA} \text{ oppure } T > \text{TRAMONTO}$$

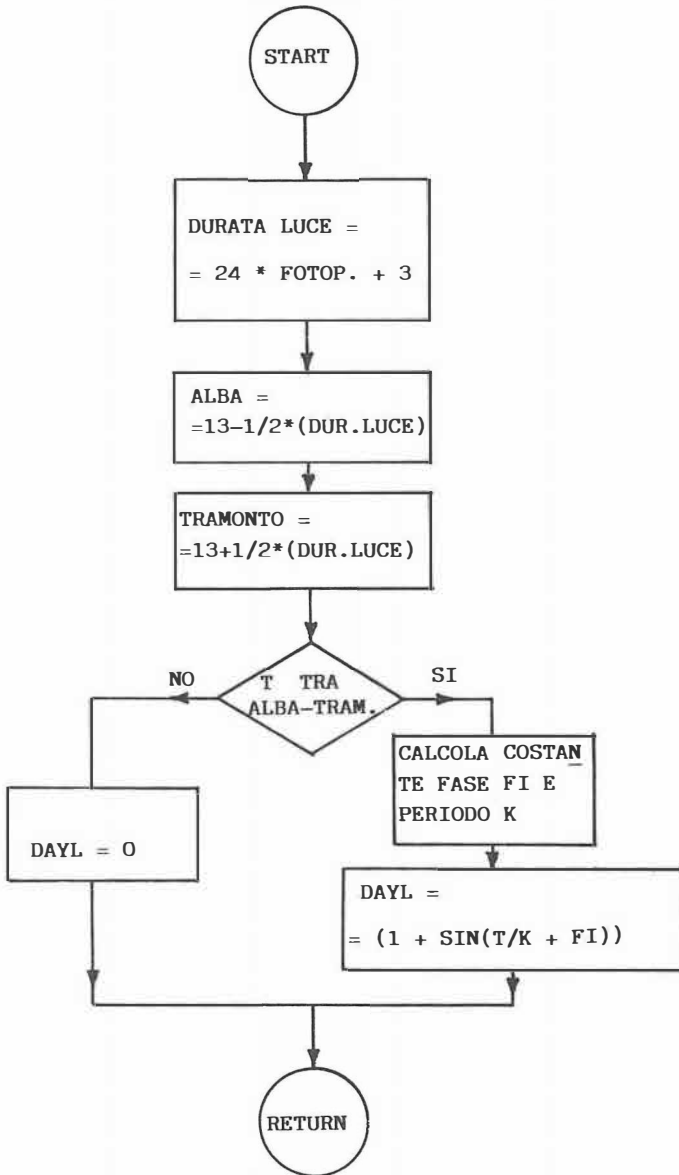
$$\text{DAYL} = \text{ABS}(1 + \text{SIN}(T/K + \text{FI})) \text{ nel restante periodo di tempo.}$$

K e FI sono dedotti ad ogni passo dal valore della durata della luce e poi usati nel calcolo.

Nel main program la intensità della radiazione in superficie sarà data dalla:

$$\text{RADSUP}(T) = \text{RADMX}(T) * \text{DAYL}(\text{FOTOP}, T)$$

FUNCTION DAYL



Funzione RADMX.

Questa funzione calcola una stima dell'intensità massima giornaliera della radiazione solare al tempo T, che compare qui mediante il numero del giorno dell'anno nella relazione:

$$\text{RADMX}(T)=(600-340*\text{COS}(6.283*(\text{NG}+10)/365))*F1*F2$$

dove NG é il numero del giorno, pari a $\text{MOD}((T/24),365)$. Ciò permette di far partire il programma in qualsiasi giorno dell'anno, per un qualunque numero di giorni od anni. Le funzioni periodiche si rinnovano ciascuna secondo il suo proprio periodo. F1 (=0.85) ed F2 (=0.45) rappresentano coefficienti di correzione per l'effetto della luce diffusa e riflessa nell'atmosfera. Per ulteriori particolari confronta il citato lavoro di Kremer e Nixon.

Funzione CINE.

Essa calcola i valori delle cinetiche di Michaelis-Menten associate all'assunzione dei nutrienti ed alla stima della probabilità di sopravvivenza delle zoospore in funzione della concentrazione dell'ossigeno.

Miscellanea.

Ci sono numerose altre funzioni usate nel programma le quali non agiscono come funzioni indipendenti, ma come parte di subroutines oppure nel main program. Esse calcolano, per esempio, la massima intensità di crescita, la temperatura, il fotoperiodo, i fattori di efficienza respiratoria, e così via. La loro descrizione analitica é stata ampiamente discussa in Lagonegro & Hull (1987), così che sarà agevole riconoscerle nel listato per chiunque si ponga il problema di adattare il presente modello alle proprie esigenze, mentre per l'utente generico non importa ovviamente dove una grandezza sia calcolata, purché lo sia. E' probabile che in eventuali future versioni del modello esse vengano enucleate come parti indipendenti per accrescere la flessibilità e la adattabilità del programma.

ESEMPIO ESPLICATIVO

Per fornire una visione d'insieme, il più possibile, completa del modello intendiamo mostrare con un breve esempio alcuni risultati, anche grafici, che possono essere ottenuti con AQMODL5. A questo fine riportiamo alcuni grafici prodotti con AQPLOT5 relativi ad una simulazione protratta sul breve periodo (5 giorni). I dati, immessi con AQINPT5, che sono serviti per produrre i risultati fanno riferimento ad una serie di osservazioni condotte sulla laguna di Orbetello, ancora in via di rilevazione sperimentale, ma che risultano efficaci, anche in questa fase assolutamente preliminare, per essere utilizzati a fini interpretativi sia dei processi che delle interazioni che avvengono nell'ecosistema analizzato. Siamo infatti convinti che l'utilizzazione di un modello di simulazione, come quello che abbiamo descritto, presenti delle prerogative di sfruttamento anche nelle fasi di sperimentazione sul campo, in quanto è in grado di suggerire informazioni quanto mai utili sul "peso" dei singoli fattori che influenzano i processi di crescita delle componenti biologiche che si analizzano.

In tabella 1 sono riportati i risultati prodotti dalla simulazione nei cinque giorni selezionati (corrispondenti al periodo 30 maggio - 4 giugno) per i gruppi biologici, mentre in tabella 2 si riportano le interpolazioni dei nutrienti e dell'ossigeno disciolto. I risultati della simulazione sono forniti ogni quattro ore. Le variabili ecologiche sono tutte trattate in termini di flusso di azoto, i gruppi biologici sono espressi in mg/mc di N, mentre i nutrienti e l'ossigeno disciolto in millimoli/mc di elemento.

TABELLA 1.

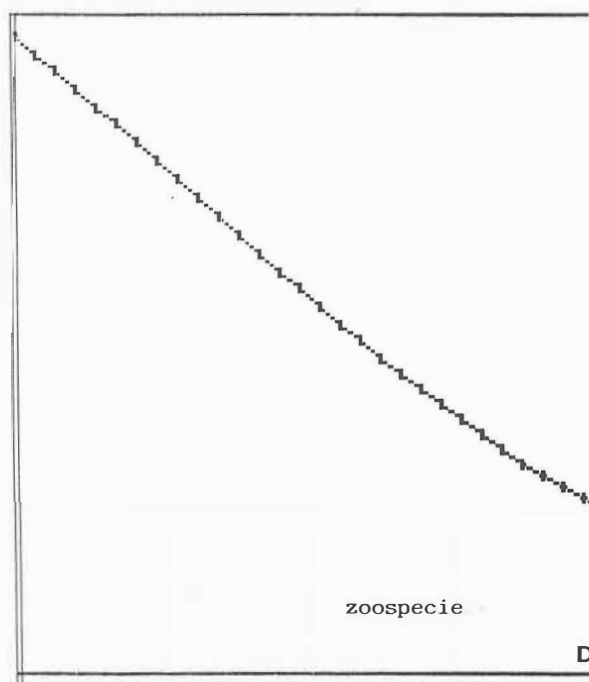
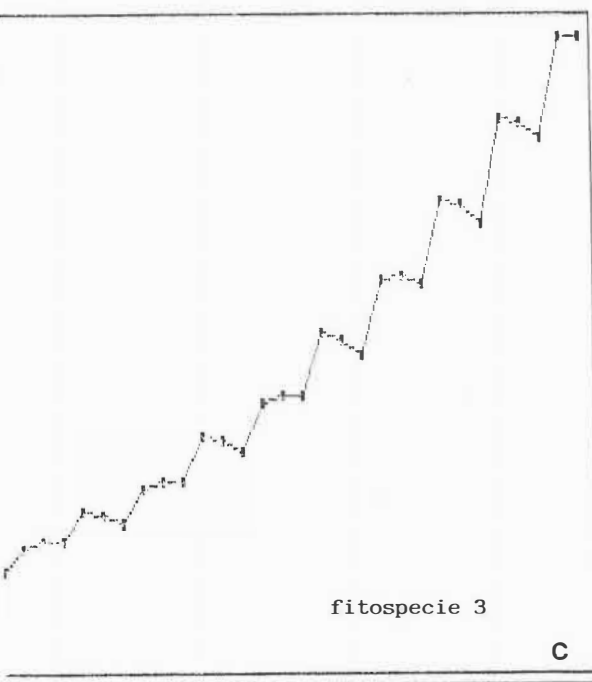
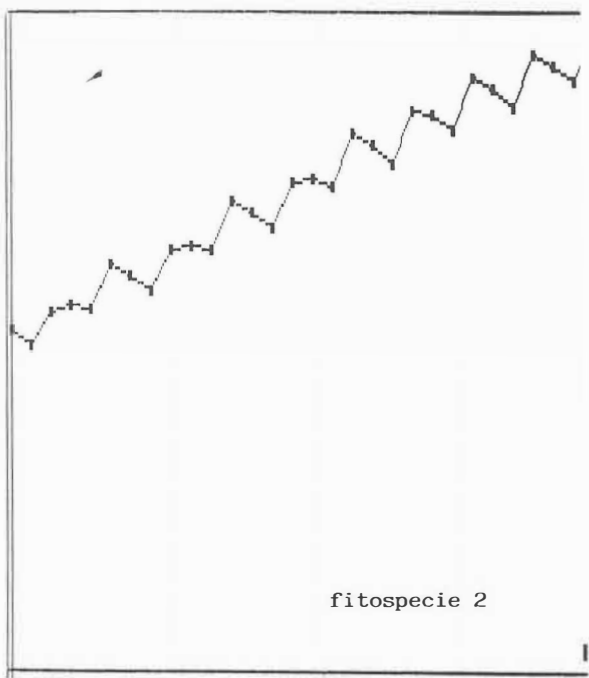
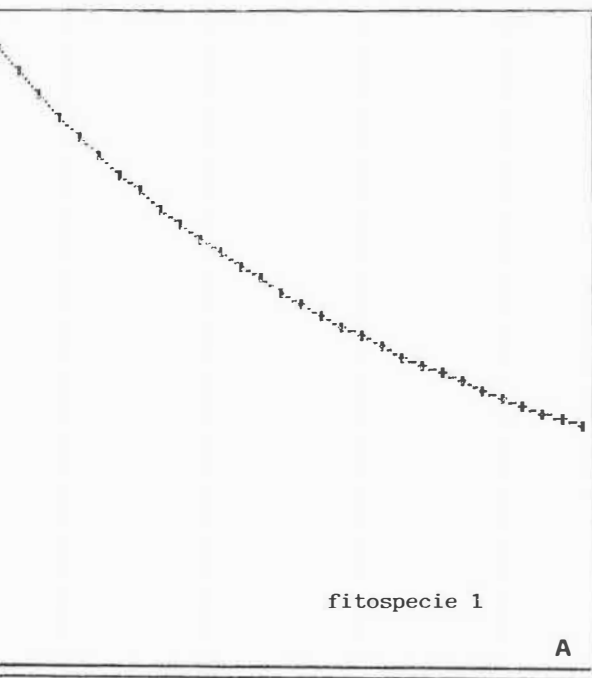
Giorno	Fitosp.1	Fitosp.2	Fitosp.3	Zoosp.1
150.0	80.0	40.0	25.0	25.0
150.2	77.0	38.3	23.7	24.3
150.3	74.1	42.5	28.8	23.7
150.5	71.4	43.1	30.0	23.0
150.7	68.8	42.8	30.1	22.2
150.8	66.4	48.1	37.1	21.5
151.0	64.0	46.8	36.2	20.8
151.2	61.2	44.9	34.5	20.1
151.3	59.6	49.9	42.1	19.4
151.5	57.6	50.4	43.8	18.6
151.7	55.6	49.9	44.0	17.9
151.8	53.7	55.8	54.3	17.2
152.0	52.0	54.5	53.2	16.5
152.2	50.3	52.4	50.8	15.8
152.3	48.7	58.0	62.3	15.1
152.5	47.2	58.2	64.2	14.4
152.7	45.7	57.3	63.7	13.7
152.8	44.3	63.4	78.3	13.0
153.0	43.0	62.1	77.2	12.4
153.2	41.7	60.0	74.0	11.7
153.3	40.5	66.1	91.1	11.1
153.5	39.3	65.6	92.0	10.5
153.7	38.1	64.1	91.1	9.9
153.8	37.1	70.0	109.2	9.4
154.0	36.0	68.9	108.4	8.8
154.2	35.0	66.6	104.1	8.3
154.3	34.0	72.8	127.9	7.8
154.5	33.1	71.6	126.8	7.3
154.7	32.2	69.7	123.4	6.9
154.8	31.3	75.1	147.1	6.5
155.0	30.5	74.1	146.9	6.0

Valori delle biomasse dei gruppi biologici simulati dal modello AQMODL5.

TABELLA 2.

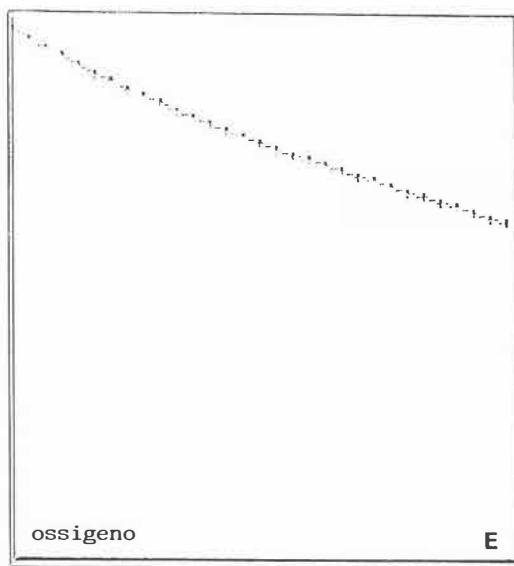
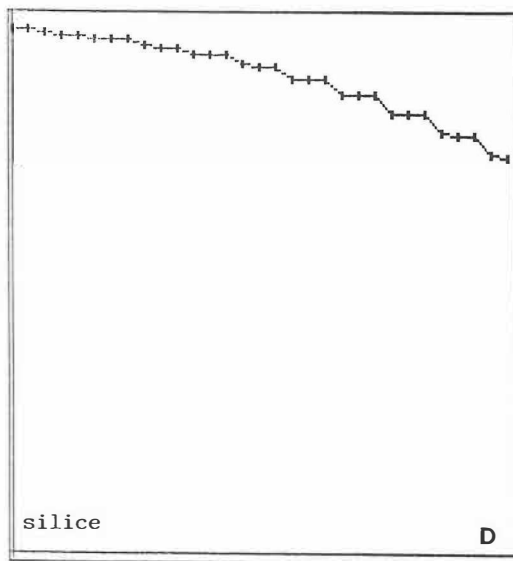
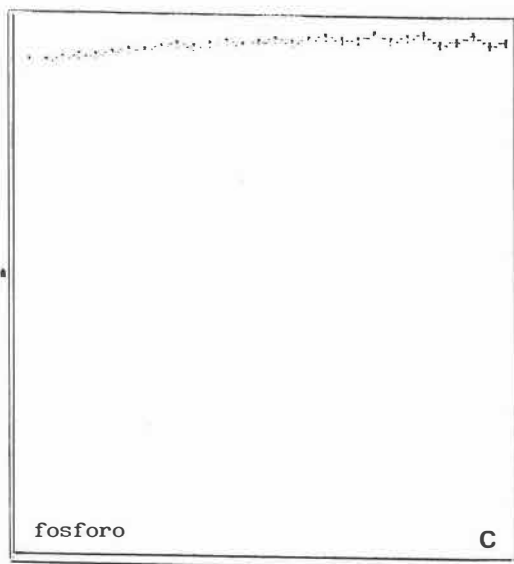
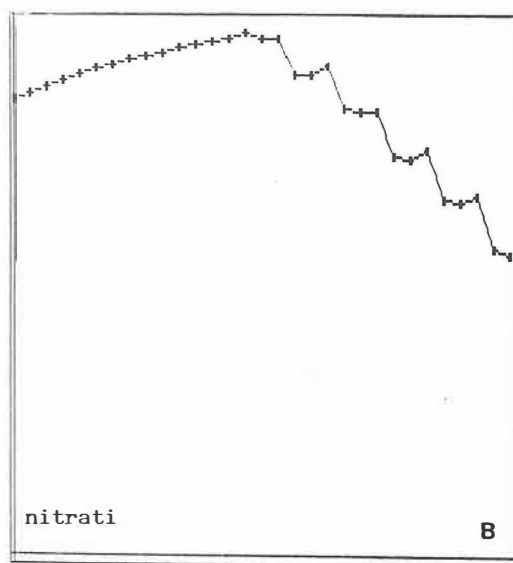
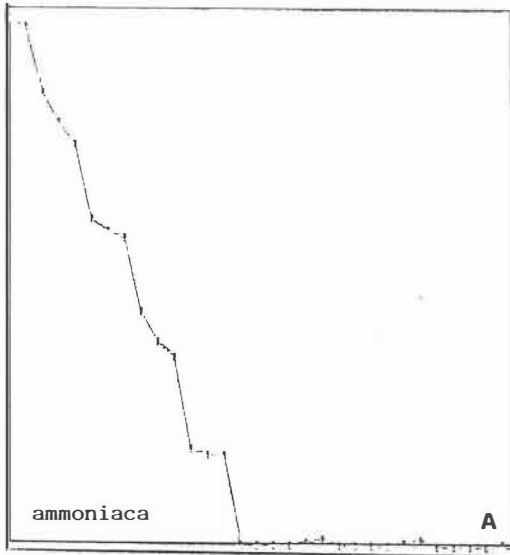
Giorno	N-NH4	N-NO3	P-PO4	Si-SiO3	O2
150.0	8.1	21.0	9.3	47.5	56.3
150.2	7.9	21.3	9.4	47.6	55.3
150.3	6.9	21.6	9.4	47.2	54.3
150.5	6.5	21.8	9.4	47.2	53.4
150.7	6.2	22.1	9.5	47.1	52.6
150.8	5.0	22.3	9.5	46.6	51.6
151.0	4.8	22.6	9.5	46.7	50.9
151.2	4.8	22.8	9.6	46.8	50.1
151.3	3.6	23.0	9.6	46.2	49.2
151.5	3.1	23.2	9.6	46.0	48.5
151.7	2.9	23.3	9.7	46.0	47.7
151.8	1.5	23.5	9.7	45.2	47.0
152.0	1.4	23.6	9.7	45.2	46.4
152.2	1.4	23.7	9.8	45.3	45.8
152.3	0.0	24.1	9.8	44.4	45.0
152.5	0.0	23.8	9.8	44.2	44.4
152.7	0.0	23.7	9.9	44.2	43.8
152.8	0.0	22.1	9.8	43.1	43.1
153.0	0.0	22.1	9.8	43.0	42.6
153.2	0.1	22.5	9.9	43.1	42.1
153.3	0.0	20.7	9.8	41.8	41.4
153.5	0.0	20.5	9.9	41.6	40.9
153.7	0.0	20.6	9.9	41.6	40.3
153.8	0.0	18.6	9.8	40.1	39.6
154.0	0.0	18.5	9.9	40.0	39.2
154.2	0.0	18.8	10.0	40.1	38.7
154.3	0.0	16.5	9.8	38.2	38.0
154.5	0.0	16.4	9.9	38.1	37.6
154.7	0.0	16.7	9.9	38.1	37.1
154.8	0.0	14.3	9.8	36.3	36.4
155.0	0.0	14.1	9.8	36.0	36.0

Valori delle interpolazioni eseguite sui nutrienti dal modello AQMODL5.



Andamento simulato delle variabili biologiche

FIG. 2



Andamento interpolato **FIG. 3**
delle variabili
ecologiche

Nelle figure 2a, 2b e 2c sono riportate singolarmente le curve di crescita dei tre gruppi vegetali nel periodo selezionato per la simulazione. Mentre il primo è in declino, a causa delle condizioni ambientali (temperatura elevata) non favorevoli alla crescita per la specie, le altre due sono in fase esponenziale di crescita, evidenziando al contempo un'attività giornaliera sostenuta ed una stasi notturna. Il gruppo zooplanctonico (Fig. 2d) mostra di non gradire particolarmente le specie vegetali presenti, essendo comprese solo in modo marginale nella sua dieta, quindi è anch'esso in declino nella prima fase del periodo di simulazione ed in leggera ripresa nella seconda.

Per quanto si riferisce ai nutrienti (Fig. 3a, 3b, 3c e 3d) si può evidenziare come venga consumato preferibilmente l'azoto nella forma ridotta, almeno fintanto che risulta presente in concentrazione sufficiente alla crescita, poi viene assimilato anche la forma più ossidata. Il fosforo e la silice reattiva non subiscono significativi abbassamenti di concentrazione, segno che l'ambiente ne è sufficientemente arricchito, almeno tanto da non limitare la crescita dei gruppi vegetali analizzati. L'ossigeno (Fig 3e) mostra di essere in declino costante, sia di giorno che di notte, dimostrando che il bilancio con l'attività fotosintetica della popolazione vegetale è decisamente a favore del consumo.

SUMMARY

The numerical simulation model, discussed in this manual, is derived from a more simple one, which has been developed by us in the last few years. This first model, written in BASIC programming language under MS-DOS operating system, was able to perform computations of two phytoplanktonic species growth, which were nutrient, light and temperature limited. Further versions of the model saw an herbivorous zooplankter and a carnivorous included into simulations and, more over, different ways to input nutrients to the system, a day-night effect etc.

The aim of our work in the field of system ecology was, and still is, to provide Italian reaserchers as well as students in biological and social sciences of modeling tools for their work.

The model of this version, called AQMODL5, is in the reductionist view of ecosystem studies, this means that every component of the natural system has been analyzed singularly and then related to the others in a mechanistic form. As the pieces of a puzzle give the idea of the whole design only when all assembled, so this model reproduces the reality (it does simulate it) imiting the effect that every single component produces on the ecological variables. In this sense it can be also considered deterministic, as it computes exactly the predicted values, and said dynamic, as all variables are time dependent.

AQMODL5 has been developed to simulate plankton growth in an aquatic environment, so the different components that, some how, influence plankton's growth (such as water temperature, sun radiation, availability of food, abundance of predators, mortality and respiration rates etc.) have been analyzed. All the different kinds of relation between considered components and growth have been defined in detail, then relations have been translated in a formal language (mathematics). All math transformations have been assembled into differential equations that are solved by a computer in respect to a time increment factor. The results of the defferential equations are new predicted values of ecological variables.

In this booklet the most new version of the model is decribed and discussed with an example. More informations about how the model has been built, and how all parameters have been chosen, can be found in the references referred to the authors. Copies of the programs on 5.25 in. diskettes can be obtained on request supporting a new disk.

RIFERIMENTI BIBLIOGRAFICI

Di Toro D.M., O'Connor D.J., Thomann R.V.; 1971 - A dynamic model of phytoplankton populations in the Sacramento - San Joaquin Delta. *Advanc. Chem. Series*, 106, 131-180.

Eppley R.; 1972 - Temperature and phytoplankton growth in the sea. *Fish. Bull.*, 70, 1063-1085

Hull V., Lagonegro M.; 1988 - AQUAMOD: an introductory purpose simulation model of plankton dynamics. *Coenoses* 3(1):55-60.

Hull V., Lagonegro M., Puccia C.J.; 1988 - Modellizzazione di sistemi ecologici complessi: modelli qualitativi e di simulazione di ambienti acquatici. CLUP (Eds.) Milano, 109 PP.

Kremer J.N., Nixon S.W.; 1978 - A coastal marine ecosystem -simulation and analysis. *Ecological Studies V. 24* Springer-Verlag (Eds.) Heidelberg, 217 PP.

Lagonegro M., Hull V.; 1987 - Models for simple aquatic systems: equations and programs. *Quad. GEAD n.6 Univ. Trieste*, 100 PP.

Platt T., Harrison W.G., Irwin B., Horne E.P., Gallegos C.L.; 1982 - Photosynthesis and photoadaptation of marine phytoplankton in the Arctic Sea. *Deep Sea Res.*, 29, 1159-1170.

Riley G.A.; 1956 - Oceanography of Long Island Sound. II Physical oceanography. *Bull. Bingham Oceanog. Coll.*, 15, 15-46.

Steele J.H.; 1962 - Environmental control of photosynthesis in the sea. *Limnol. Oceanog.*, 7, 137-150.

UNESCO 1983 - Quantitative analysis and simulation of Mediterranean coastal ecosystem: the Gulf of Naples, a case study. *UNESCO Reports in Marine Sciences*, n.20, UNESCO Paris, 158 PP.

Walsh J.J.; 1976 - Models in the sea. In: *The ecology of the seas*, Cushing D.H. and J.J. Walsh (Eds.), Oxford, 388-408.

APPENDICE

In questa appendice sono riportati i listati dei programmi del pacchetto, di cui AQMODL5, il programma che implementa le caratteristiche del modello di simulazione, e' il pezzo centrale. Non c'e' nulla da aggiungere su di esso, in quanto gia' discusso in precedenza, in compenso verranno fatti alcuni commenti ai programmi ausiliari, anche se il loro uso e' pilotato da un menu' e percio' sono autoesplicetivi. Chi volesse ulteriori dettagli, li puo' trovare in tutti i lavori gia' citati.

PROGRAM AQMODL5

```
*---THE FOLLOWING COMMON STATEMENTS GROUP VARIABLES TRYING TO
*---HAVE HOMOGENEOUS BLOCKS,TO MAKE IT EASIER TO THE USER TO
*---CHANGE THINGS BY EDITING THE PROPER LINES,ONLY DEPTH IS A
*---CHARACTERISTIC PRESENT IN MANY COMMONS.
*.....
*---/VEGET/ GROUPS VARIABLES FOR PHYTO SPECIES (3) AT 5 DEPTHS
COMMON/VEGET/PHYT(3,5),PKN(3),PKP(3),PKSI(3),DESP(3),DTDEC(3)
1,U(3),UMAX(3),R(3),TMAX(3),PK1(3),ROPT(3),RE(3),CMD(3),PKD(3)
2,URI(3),FNP(3),FNSI(3),DPHYT(3),PHYFR(3),FNH(3),FND(3)
*---/BESTIE/ GROUPS VARIABLES FOR ZOO SPECIES (3) AT 5 DEPTHS
COMMON/BESTIE/ZOOP(3,5),FKCNV(3),FKMAX(3,3),FKMAX(3,3),FKF(
13),D(3),DZOO(3),ROZ(3),RZ,ZNH(3),ZO2(3),ZOF(3)
*---/DEPTH/ GROUPS TYPICAL VARIABLES & TOTAL BIOMASSES AT 5 DEPTHS
COMMON/DEPTH/NZ,TM(5),TSC(5),TEMP(5),THICK(5),ZOOT(5),PHP(5)
*---/NUTR/ GROUPS NUTRIENT VARIABLES AT 5 DEPTHS(MAX,60 VALUES EACH)
COMMON/NUTR/ENH4(5),EN03(5),PD4(5),SIL(5),AZSW(5),PSW(5)
1,SISW(5),ENH(5),EN0(5),P0(5),SILI(5),ENHM(5,60),ENDM(5,60)
2,POM(5,60),SIM(5,60),T(60),ULIM(5)
*---/OSSI/ GROUPS OXYGEN PARAMETERS AT 5 DEPTHS(MAX,60 VALUES EACH)
COMMON/OSSI/O2(5),OSW(5),O(5),O2M(5,60),FTE(3)
1,ZRE(3),FSED(5,60),ONIT(5),FRE(3)
*---/LUCE/ GROUPS RADIATION VARIABLES,GENERAL & SOME AT 5 DEPTHS
COMMON/LUCE/BSP,ASP,CZ,RA1(5),RA2(5),RA3(5),ACMI,FOT
*---/PARAMS/ GROUPS MISCELLANEA PARAMETERS ABOUT COMPUTATION
COMMON/PARAMS/DT,NFI,NOU,FT,ITA,DAY,HR,FRAZ,FRAP,FRAS,FRAO
CHARACTER*8NOFIPA,OLP0N
CHARACTER*100PO(3)
DATA NX,NY/1,1/
CALL PRESET
```

```

WRITE(*,*)           A Q M O D L   5'
WRITE(*,*)'
WRITE(*,*)' ----- LAGONEGRO & HULL 1988 -----'
WRITE(*,*)' DATA AND PARAMETERS FILENAME-FROM AQINPT'
READ(*, '(A8)')NOFIPA
WRITE(*, '(1X,A8)')NOFIPA
OPEN(1, FILE=NOFIPA, STATUS='OLD')
REWIND 1
READ(1,*)N,NZO
READ(1,*)NZ,DT,NU
WRITE(*,*)' CONSIDERED N.OF PHYTOSPECIES ',N
WRITE(*,*)' CONSIDERED N.OF ZOOSPECIES ',NZO
WRITE(*,*)' CHOSEN LIGHT FACTOR-STEEL(1)/DITORO(2)/PLATT(3)'
READ(1,*)LUX
WRITE(*,*)' CHOSEN EXT.COEFF. FORMULA-RILEY(1)/WALSH(2)'
READ(1,*)KEST
WRITE(*, '(1X,I3,1X,I3)')LUX,KEST
KG=NX*NY*NZ
NU1=NU-1
WRITE(*,*)' CHOSEN FILENAME FOR PLOT/NO'
READ(1, '(A8)')OLPON
WRITE(*, '(1X,A8)')OLPON
IF(OLPON.EQ.'NO')GOTO 2
OPEN(2, FILE=OLPON, STATUS='NEW')
2 WRITE(*,*)' PRINTOUT ON AVAILABLE SUPPLY LIMITS-Y(1)/N(0)'
READ(1,*)JESNO
WRITE(*, '(1X,I3)')JESNO
CALL INIT(OLPON,KEST,KOPA,KG,N,NU,NU1,NZO,FIMES)
TEBASE=FLOAT(ITA)*24.
DO 3 KUK=1,NFI
TE=KUK*DT+TEBASE
TGIO=MOD(TE,24.)
FOT=.5-.125*COS(6.283*FLOAT(ITA+10)/365.)
RTOP=RADM(ITA+3)*DAYL(FOT,TGIO)
RATOP=RTOP
DO 4 KG=1,NZ
TEMP(KG)=TM(KG)-TSC(KG)*COS(6.283*(ITA+FIMES)/365.)
TFAME=EXP(0.12*TEMP(KG))
DO 5 I=1,N
IF(TEMP(KG).LE.TMAX(I)) THEN
UMAX(I)=.59*EXP(PK1(I)*TEMP(KG))/24.*EXP(PK1(I)
1*(TEMP(KG)-TMAX(I)))
ELSE
UMAX(I)=.59*EXP(PK1(I)*TMAX(I))/24.
1*EXP(-(((TEMP(KG)-TMAX(I))/DTDEC(I))**DESP(I)))
ENDIF
CONTINUE
CLA=PHP(KG)*.476
CLO=PHP(KG)
IF(KEST.EQ.1)C=.04+.054*CLA**(.6667)+.0088*CLA
IF(KEST.EQ.2)C=.16+.039*CLO**(.6667)+.0053*CLO

```

```

CZ=C*THICK(KG)
X=EXP(-CZ)
AVGLT=RTOP*(1.-X)/CZ
POIOL=ACMI*CZ/(1.-X)*EXP(-C)
IF(POIOL.LT.ACMI)POIOL=ACMI
RTOP=RTOP*X
IF(TE.GE.T(1))GOTO 6
K=1
GOTO 7
6 DO 8 K=1,NU1
  IF(TE.GE.T(K),AND,TE.LT.T(K+1))GOTO 7
8 CONTINUE
  K=NU1
7 IF(KOPA.NE.3) THEN
  CALL MESRES(TE,K,KG)
  ELSE
  CALL INPRAT(FT,K,KG)
  ENDIF
  DO 30 I=1,NZO
  ZOF(I)=CINE(O2(KG),ZO2(I))
  IF(KOPA.EQ.2)ZOF(I)=1.
30 CONTINUE
  AZO=ENH4(KG)+ENO3(KG)
  FOS=PO4(KG)
  SIS=Sil(KG)
  OSS=O2(KG)
  ROPT(1)=.7*RA1(KG)+.2*RA2(KG)+.1*RA3(KG)
  IF(ROPT(1).LT.POIOL)ROPT(1)=POIOL
  DO 28 MAH=2,N
28 ROPT(MAH)=ROPT(1)
  RA3(KG)=RA2(KG)
  RA2(KG)=RA1(KG)
  RA1(KG)=RADMX(ITA+3)*EXP(-C)
  CALL PHYTO(AVGLT,LUX,OPD,KG,N,BSP,ASP,CZ,ENH,ENO,PO,SILI,
  *O,KOPA)
  CALL UPRES(BAS,FNSI,FNP,FNO,KOPA,JESNO,FRAZ,FRAP,FRAS,K,U,
  *PHYT,AZO,FOS,SIS,DT,N,NZO,KG,FRAO,OSS,ZOOPL,ROZ,RZ,RE,TEMP,
  *CMO,D,FNH,ZNH,DPHYT,FT)
  PHP(KG)=0.
  DO 23 I=1,N
  U(I)=U(I)*BAS
  REMO=(RE(I)*FRAO+CMO(I))
  DPHYT(I)=(U(I)-REMO)*DT
23 PHP(KG)=PHP(KG)+PHYT(I,KG)
  ZOOT(KG)=0.
  DO 26 J=1,NZO
26 ZOOT(KG)=ZOOT(KG)+ZOOPL(J,KG)
  DO 24 I=1,N
  DO 25 J=1,NZO
  PHYFR(I)=FT/(FKF(J)+PHP(KG))
25 DPHYT(I)=DPHYT(I)-ZOOPL(J,KG)*PHYFR(I)*FMAX(J,I)*TFAME

```

```

PHYT(I,KG)=PHYT(I,KG)*(1.+DPHYT(I))
IF (PHYT(I,KG).LT.0.01)PHYT(I,KG)=.01
24 CONTINUE
DO 27 J=1,NZO
CALL ZOO(GNAM,J,KG,N,NZO,ITA,PHP,TFAME,ZOOT,PHYT)
DZOO(J)=(GNAM-D(J)-FRAO*ROZ(J)*EXP(RZ*TEMP(KG)))*FT
ZOOPL(J,KG)=ZOOPL(J,KG)*(1.+DZOO(J))*ZOF(J)
IF (ZOOPL(J,KG).LT.0.01)ZOOPL(J,KG)=.01
27 CONTINUE
IF (HR,EQ,0.)GOTO 4
IF (NOU,LE,(INT(12./DT+.1)))GOTO 29
KUK12=KUK+INT(12./DT)
IF (MOD(KUK12,NOU),EQ,0)CALL OUT(OLPON,KUK,BAS,C,OP0,KOPA
1,KG,N,AVGLT,NZO,RATOP)
29 IF (MOD(KUK,NOU),EQ,0)CALL OUT(OLPON,KUK,BAS,C,OP0,KOPA,KG,N
1,AVGLT,NZO,RATOP)
4 CONTINUE
HR=HR+DT
IF (HR,LT,24.)GOTO 3
IF (HR,GE,24.)DAY=DAY+1.
IF (HR,GE,24.)ITA=MOD(ITA+1,365)
HR=0.
3 CONTINUE
IF (OLPON,NE,'NO')CLOSE(2)
WRITE(*,*)' END CALCULATION'
WRITE(*,*)' RUN AQPLOT TO GET PLOTS'
CLOSE(1)
END

```

```

SUBROUTINE ZOO(GNAM,IO,KG,N,NZO,ITA,PHP,TFAME,ZOOT,PHYT)
COMMON/BESTIE/ZOOPL(3,5),FKCNV(3),FMAX(3,3),CMAX(3,3),FKF(
13),D(3),DZOO(3),ROZ(3),RZ,ZNH(3),Z02(3),ZOF(3)
DIMENSION PHP(5),ZOOT(5),PHYT(3,5)
FERB=0.
FCAR=0.
FCAP=0.
DO 1 I=1,N
1 FERB=FERB+FMAX(IO,I)*PHYT(I,KG)/(FKF(IO)+PHP(KG))
DO 2 J=1,NZO
2 FCAR=FCAR+CMAX(IO,J)*ZOOPL(J,KG)/(FKCNV(IO)+ZOOT(KG))
FCAP=FCAP+CMAX(J,IO)*ZOOPL(J,KG)/(ZOOT(KG)+FKCNV(J))
GNAM=(FERB+FCAR-FCAP)*TFAME
RETURN
END

```

```

SUBROUTINE MESRES(TE,K,KG)
COMMON/NUTR/ENH4(5),ENO3(5),P04(5),SIL(5),AZSW(5),PSW(5)
1,SISW(5),ENH(5),ENO(5),P0(5),SILI(5),ENHM(5,60),ENOM(5,60)
2,POM(5,60),SIM(5,60),T(60),ULIM(5)
COMMON/OSSI/O2(5),OSW(5),O(5),O2M(5,60),FTE(3)
1,ZRE(3),FSED(5,60),ONIT(5),FRE(3)
DET=TE-T(K)
DOT=T(K+1)-T(K)
DAT=DET+DT
DEO=DET/DOT
DAO=DAT/DOT
DNHM=ENHM(KG,K+1)-ENHM(KG,K)
DNOM=ENOM(KG,K+1)-ENOM(KG,K)
DPOM=POM(KG,K+1)-POM(KG,K)
DSIM=SIM(KG,K+1)-SIM(KG,K)
DOSS=O2M(KG,K+1)-O2M(KG,K)
DEON4=DEO*DNHM
DAON4=DAO*DNHM
DEON3=DEO*DNOM
DAON3=DAO*DNOM
DEOP=DEO*DPOM
DAOP=DAO*DPOM
DEOS=DEO*DSIM
DAOS=DAO*DSIM
DEO2=DEO*DOSS
DAO2=DAO*DOSS
ENH4(KG)=(ENHM(KG,K)+.5*(DEON4+DAON4))
ENO3(KG)=(ENOM(KG,K)+.5*(DEON3+DAON3))
P04(KG)=(POM(KG,K)+.5*(DEOP+DAOP))
SIL(KG)=(SIM(KG,K)+.5*(DEOS+DAOS))
O2(KG)=(O2M(KG,K)+.5*(DEO2+DAO2))
ENO(KG)=ENO3(KG)
ENH(KG)=ENH4(KG)
P0(KG)=P04(KG)
SILI(KG)=SIL(KG)
O(KG)=O2(KG)
RETURN
END

```

```

SUBROUTINE INPRAT(FT,K,KG)
COMMON/NUTR/ENH4(5),ENO3(5),P04(5),SIL(5),AZSW(5),PSW(5)
1,SISW(5),ENH(5),ENO(5),P0(5),SILI(5),ENHM(5,60),ENOM(5,60)
2,POM(5,60),SIM(5,60),T(60),ULIM(5)
COMMON/OSSI/O2(5),OSW(5),O(5),O2M(5,60),FTE(3)
1,ZRE(3),FSED(5,60),ONIT(5),FRE(3)
ENH4(KG)=ENHM(KG,K+1)*FT+ENH4(KG)
ENO3(KG)=ENOM(KG,K+1)*FT+ENO3(KG)
P04(KG)=POM(KG,K+1)*FT+P04(KG)

```

```

SIL(KG)=SIM(KG,K+1)*FT+SIL(KG)
O2(KG)=O2M(KG,K+1)*FT+O2(KG)
ENO(KG)=ENO3(KG)
ENH(KG)=ENH4(KG)
PO(KG)=PO4(KG)
SILI(KG)=SIL(KG)
O(KG)=O2(KG)
RETURN
END

```

```

SUBROUTINE INIT(OLP0N,KEST,KOPA,KG,N,NU,NU1,NZO,FIMSE)
COMMON/VEGET/PHYT(3,5),PKN(3),PKP(3),PKSI(3),DESP(3),DTDEC(3)
1,U(3),UMAX(3),R(3),TMAX(3),PK1(3),ROPT(3),RE(3),CMO(3),PKO(3)
2,URI(3),FNP(3),FNSI(3),DPHYT(3),PHYFR(3),FNH(3),FNO(3)
COMMON/BESTIE/ZOOPL(3,5),FKCNV(3),FMAX(3,3),CMAX(3,3),FKF(
13),D(3),OZOO(3),ROZ(3),RZ,ZNH(3),ZO2(3),ZOF(3)
COMMON/DEPTH/NZ,TH(5),TSC(5),TEMP(5),THICK(5),ZOOT(5),PHP(5)
COMMON/NUTR/ENH4(5),ENO3(5),PO4(5),SIL(5),AZSW(5),PSW(5)
1,SISW(5),ENH(5),ENO(5),PO(5),SILI(5),ENHM(5,60),ENOM(5,60)
2,POK(5,60),SIK(5,60),T(60),ULIM(5)
COMMON/OSSI/O2(5),OSW(5),O(5),O2M(5,60),FTE(3)
1,ZRE(3),FSED(5,60),GNIT(5),FRE(3)
COMMON/LUCE/BSP,ASP,CZ,RA1(5),RA2(5),RA3(5),ACMI,FOT
COMMON/PARAMS/DT,NFI,NOU,FT,ITA,DAY,HR,FRAZ,FRAP,FRAS,FRAO
CHARACTER*80LP0N
DIMENSION DESE(12)
DATA ENH414,ENO314,PO431,SIL28,O216/0.,0.,0.,0.,0.,0./
READ(1,*)ACMI
WRITE(*,*)' CHOSEN MIN.ACCL.LIGHT LEVEL-(FROM AQINPT)',ACMI
READ(1,*)ULIM(1)
WRITE(*,*)'CHOSEN SWITCHING LEVEL FROM NH4 TO NO3',ULIM(1)
WRITE(*,*)'NUTRIENTS DATA IN INPUT ARE-MEAS.CONC.(1)'
WRITE(*,*)' -RESIDUALS(2)'
WRITE(*,*)' -INPUT RATES(3)'
READ(1,*)KOPA
WRITE(*,*(1X,I3))KOPA
READ(1,*)FIMSE
WRITE(*,28)FIMSE
READ(1,*)ILA,ILO
WRITE(*,*)' DATE(MONTH,DAY)OF STARTING T FOR SIMULATION:',ILA,ILO
28 FORMAT(' INTERPOLATED (FROM AQTEMP)PHASE FOR TEMP.CURVE IS:',F6.1)
ITA=(ILA-1)*30+ILO
IF(ITA.LT.1)ITA=1
IF(ITA.GT.365)ITA=MOD(ITA,365)
DAY=1.
HR=0.
READ(1,*)OUTINT

```

```

READ(1,*)FINTIM
NFI=INT(FINTIM*24./DT+.5)
NOU=INT(OUTINT/DT+.5)
KINPU=NFI/NOU
FT=DT/24.
WRITE(*,29)FINTIM,OUTINT
29  FORMAT(' COMPUTATION TIME SPAN,IN DAYS:',F6.1/
*' TIME INTERVAL FOR OUTPUT,IN HOURS',F6.1)
IF(OLPON.EQ.'NO')GOTO 4
KINPU1=2*KINPU+1
IF(OUTINT.LE.12.)KINPU1=KINPU+1
OUT24=OUTINT/24.
WRITE(2,*)N,NZ0,NZ,KINPU1,OUT24,ITA
4   DO 7 I=1,N
READ(1,*)RE(I),CMO(I),TMAX(I),PK1(I),FRE(I),FTE(I)
WRITE(*,30)RE(I),CMO(I),TMAX(I),PK1(I),FRE(I),FTE(I)
30  FORMAT(' RESPIRATION AND MORTALITY:',2G11.4/
*' MAX.GROWTH TEMP.:',G11.4,' GROWTH C.:',G11.4/
*' O2 CONSUM.RATE:',G11.4,' O2 PRODUCTION EFF.:',G11.4)
R(I)=.027
URI(I)=(1.+R(I))
READ(1,*)PKN(I),PKP(I),PKSI(I),PKO(I)
WRITE(*,31)PKN(I),PKP(I),PKSI(I),PKO(I)
31  FORMAT(' HALF SAT.CONST. FOR N,P,SI,O2:',4G11.4)
DTDEC(I)=4.
DESP(I)=2.
RE(I)=RE(I)/24.
7   CMO(I)=CMO(I)/24.
DO 8 I=1,KG
READ(1,*)THICK(I)
READ(1,*)TM(I),TSC(I),ONIT(I)
WRITE(*,32)I,I,TM(I),TSC(I),I,THICK(I),ONIT(I)
32  FORMAT(' TM(',I3,')AND TSC(',I3,')ARE ',2F6.1/
1' AT DEPTH N.',I3,') OF METERS ',F6.1/
2' WHERE NITRIFICATION O2/NH4 RATE IS:',G11.4)
10  CONTINUE
PHP(I)=0.
DO 17 IK=1,N
READ(1,*)PHYT(IK,I)
WRITE(*,33)IK,PHYT(IK,I)
33  FORMAT(' STARTING VALUE FOR PHYTO SP.',I3,') IS ',G11.4)
17  PHP(I)=PHP(I)+PHYT(IK,I)
DO 18 IK=1,NZO
READ(1,*)ZOOPL(IK,I)
18  WRITE(*,34)IK,ZOOPL(IK,I)
34  FORMAT('          FOR ZOOSP.N.',I3,') IS ',G11.4)
CLA=PHP(I)*.476
IF(KEST.EQ.2)C=.16+.039*PHP(I)**(.6667)+.0053*PHP(I)
IF(KEST.EQ.1)C=.04+.054*CLA**(.6667)+.0088*CLA
UM=EXP(-C)

```

```

POIOL=RADMX(ITA+3)*UM
IF(POIOL.LT.ACMI)POIOL=ACMI
RA1(I)=RADMX(ITA+2)*UM
RA2(I)=RADMX(ITA+1)*UM
RA3(I)=RADMX(ITA)*UM
ROPT(1)=.7*RA1(I)+.2*RA2(I)+.1*RA3(I)
IF(ROPT(1).LT.POIOL)ROPT(1)=POIOL
DO 19 IK=1,N
ROPT(IK)=ROPT(1)
19 WRITE(*,35)IK,ROPT(IK)
35 FORMAT(' OPTIMAL I FOR PHYTO',I3,' AT START TIME:',G11,4)
WRITE(*,*)' TIME(DAYS) NH4      NO3      PO4      SI      O2 (MG/M3)'
WRITE(*,*)' THE LAST COLUMN IS O2 LOSS RATE TO SEDIMENT'
READ(1,*)T(1),ENHM(I,1),ENOM(I,1),POM(I,1),SIM(I,1),O2M(I,1)
*,FSED(I,1)
WRITE(*,36)T(1),ENHM(I,1),ENOM(I,1),POM(I,1),SIM(I,1),O2M(I,1)
*,FSED(I,1)
36 FORMAT(1X,F6,1,6(1X,G11,4))
T(1)=T(1)*24.
DO 13 K=2,NU
READ(1,*)T(K),ENHM(I,K),ENOM(I,K),POM(I,K),SIM(I,K),O2M(I,K)
*,FSED(I,K)
WRITE(*,36)T(K),ENHM(I,K),ENOM(I,K),POM(I,K),SIM(I,K),O2M(I,K)
*,FSED(I,K)
13 T(K)=T(K)*24.
ENH4(I)=ENHM(I,1)
ENO3(I)=ENOM(I,1)
PO4(I)=POM(I,1)
SIL(I)=SIM(I,1)
O2(I)=O2M(I,1)
IF(OLPON.EQ.'NO')GOTO 15
IF(KOPA.EQ.2) THEN
  ENO3(I)=0.
  GOTO 45
ENDIF
ENH414=ENH4(I)/14.
ENO314=ENO3(I)/14.
PO431=PO4(I)/31.
SIL28=SIL(I)/28.
O216=O2(I)/16.
45 TEMP(I)=TM(I)-TSC(I)*COS(6.283*(ITA+FIMESE)/365.)
WRITE(2,*)(PHYT(M,I),M=1,N),(ZOOPL(L,I),L=1,NZO),ENH414
1. ENO314,PO431,SIL28,O216,TEMP(I)
15 ENH(I)=ENH4(I)
ENO(I)=ENO3(I)
PO(I)=PO4(I)
SILI(I)=SIL(I)
9 O(I)=O2(I)
DO 25 J=1,NZO
WRITE(*,'(1X,9HZOOSP. N.,I3)')J
READ(1,*)FKF(J),FKCV(J),D(J),ROZ(J),ZRE(J),ZNH(J),ZOZ(J)

```

```

DO 26 L=1,N
  READ(1,*)FMAX(J,L)
26  WRITE(*,37)L,FMAX(J,L),FKF(J)
37  FORMAT(' GRAZ. ON PHYTO:',I3,' OXRATE:',G11.4,' H1/2:',G11.4)
  DO 27 L=1,NZO
  READ(1,*)CMAX(J,L)
27  WRITE(*,38)L,CMAX(J,L),FKCNU(J)
38  FORMAT(' PREYING ON ZOO:',I3,' OXRATE:',G11.4,' H1/2:',G11.4)
39  WRITE(*,39)D(J),ROZ(J),ZRE(J),ZOZ(J),ZNH(J)
40  FORMAT(' MORTALITY:',G11.4,' OXC RESPIRATION:',G11.4/
*' O2 CONSUM. RATE:',G11.4,' O2 H1/2 CONST.:',G11.4/
*' DEAD MATTER RECYCLING EFF.:',G11.4)
  DO 16 I=1,N
  READ(1,*)FNP(I)
  WRITE(*,40)I,FNP(I)
40  FORMAT(' PHYTO SP.',I3,' -N/P FACTOR:',G11.4)
  IF(FNP(I).NE.0.)FNP(I)=1./FNP(I)
  READ(1,*)FNSI(I)
  WRITE(*,41)I,FNSI(I)
41  FORMAT(' PHYTO SP.',I3,' -N/SI FACTOR:',G11.4)
  IF(FNSI(I).NE.0.)FNSI(I)=1./FNSI(I)
  READ(1,*)FNO(I)
  WRITE(*,46)I,FNO(I)
46  FORMAT(' PHYTO SP.',I3,' -N/O2 FACTOR:',G11.4)
  IF(FNO(I).NE.0.)FNO(I)=1./FNO(I)
  READ(1,*)FNH(I)
16  WRITE(*,44)I,FNH(I)
44  FORMAT(' PHYTO SP.',I3,' -DEAD MATTER RECYCLING RATE:',G11.4)
  WRITE(*,*)',.....'
  RETURN
END

```

```

SUBROUTINE PHYTO(AVGLT,LUX,OP0,KG,N,BSP,ASP,CZ,ENH,ENO,PO,
*SILI,O,KDPA)
COMMON/VEGET/PHYT(3,5),PKN(3),PKP(3),PKSI(3),DESP(3),DTDEC(3)
1,U(3),UMAX(3),R(3),TMAX(3),PK1(3),ROPT(3),RE(3),CMO(3),PKO(3)
2,URI(3),FNP(3),FNSI(3),DPHYT(3),PHYFR(3),FNH(3),FNO(3)
DIMENSION ENH(1),ENO(1),PO(1),SILI(1),O(1)
CHARACTER*100PO(3)
FACTL=1.
ALIMIT=1.
DO 1 IS=1,N
X=AVGLT/ROPT(IS)
WM2=AVGLT*.4845
IF(LUX,EQ.1)FACTL=(X*EXP(1.-X)-R(IS))*URI(IS)
IF(LUX,EQ.2)FACTL=2.718*(EXP(-X*EXP(-CZ))
1-EXP(-X))/CZ
IF(LUX,EQ.3)FACTL=(1-EXP(-ASP*WM2))*EXP(-BSP*WM2)

```

```

IF(KOPA.EQ.2)GOTO 4
XX=ENH(KG)+ENO(KG)
AZLIM=CINE(XX,PKN(IS))
PLIM=CINE(PO(KG),PKP(IS))
SILIM=CINE(SILI(KG),PKSI(IS))
OLIM=CINE(O(KG),PKO(IS))
IF(PKN(IS).EQ.0.)GOTO 5
ALIMIT=AZLIM
OPO(IS)='NITROGEN'
5 IF(PLIM.GT.ALIMIT.OR.PKP(IS).EQ.0.)GOTO 2
ALIMIT=PLIM
OPO(IS)='PHOSPHURUS'
2 IF(SILIM.GT.ALIMIT.OR.PKSI(IS).EQ.0.)GOTO 3
ALIMIT=SILIM
OPO(IS)='SILICON'
3 IF(OLIM.GT.ALIMIT.OR.PKO(IS).EQ.0.)GOTO 1
ALIMIT=OLIM
OPO(IS)='OXYGEN'
GOTO 1
4 ALIMIT=1.
OPO(IS)='
1 U(IS)=UMAX(IS)*FACTL*ALIMIT
RETURN
END

```

```

SUBROUTINE OUT(OLPON,KUK,BAS,C,DPO,KOPA,KG,N,ANGLT,NZO,RATOP)
COMMON/VEGET/PHYT(3,5),PKN(3),PKP(3),PKSI(3),DESP(3),DTDEC(3)
1,U(3),UMAX(3),R(3),TMAX(3),PK1(3),ROPT(3),RE(3),CMD(3),PKO(3)
2,URI(3),FNF(3),FNSI(3),DPHYT(3),PHYFR(3),FNH(3),FNO(3)
COMMON/BESTIE/ZOOPL(3,5),FKCNV(3),FMAX(3,3),CMAX(3,3),FRF(
13),D(3),DZOO(3),ROZ(3),RZ,ZNH(3),ZO2(3),ZOF(3)
COMMON/DEPTH/NZ,TM(5),TSD(5),TEMP(5),THICK(5),ZOOT(5),PHP(5)
COMMON/NUTR/ENH4(5),ENO3(5),PO4(5),SIL(5),AZSW(5),PSW(5)
1,SISW(5),ENH(5),END(5),PO(5),SILI(5),ENHM(5,60),ENOM(5,60)
2,POM(5,60),SIM(5,60),T(60),ULIM(5)
COMMON/OSSI/O2(5),OSW(5),O(5),O2M(5,60),FTE(3)
1,ZRE(3),FSED(5,60),ONIT(5),FRE(3)
COMMON/LUCE/BSF,ASP,CZ,RA1(5),RA2(5),RA3(5),ACMI,FOT
COMMON/PARAMS/DT,NFI,NOU,FT,ITA,DAY,HR,FRAZ,FRAP,FRAS,FRAD
CHARACTER*10OPO(3)
CHARACTER*8OLPON
ORA=DT*FLOAT(KUK)-(DAY-1.)*24.
WRITE(*,12)ITA,ORA
12 FORMAT(' DAY(FROM YEAR START):',I4,' HOUR:',F4.0)
13 FORMAT(' DEPTH(M):',G10.3,' T(*C):',F5.1)
14 FORMAT(' EXTINCTION COEFF. G(1/M):',G11.4)
1' OVERALL NUTRIENT LIMITATION:',F6.3)
15 FORMAT(' ZOO SP. N.',I3,'(MG/M3):',G11.4,' O2 FACT.:',F6.3)

```

```

WRITE(*,13)THICK(KG),TEMP(KG)
WRITE(*,14)C,BAS
DO 1 IS=1,N
1 WRITE(*,7)IS,U(IS),OPQ(IS)
7 FORMAT(' PHYTO',I3,' GROWTH=',G11.4,' LIM.NUT.',A10)
IF(KOPA.EQ.2)GOTO 2
WRITE(*,*)'RESIDUALS:'
WRITE(*,8)ENH4(KG),ENQ3(KG),PQ4(KG),SIL(KG),Q2(KG)
8 FORMAT(' NH4:',G11.4,' NQ3:',G11.4,' P04:',G11.4,' SI:',G11.4,
* ' Q2:',G11.4)
GOTO 3
2 WRITE(*,*)' CONSUMPTION:'
WRITE(*,9)ENH4(KG),PQ4(KG),SIL(KG),Q2(KG)
9 FORMAT(' NH4+NQ3:',G11.4,' P04:',G11.4,' SI:',G11.4,' Q2:',
*G11.4)
3 WRITE(*,10)FOT,RATOP,ROPT(1),AVGLT
10 FORMAT(' PHOTOP.:',G11.4,' ITOP:',G11.4,' IO:',G11.4,' IAV:',
*G11.4)
DO 5 IS=1,N
5 WRITE(*,11)IS,PHYT(IS,KG),DPHYT(IS)
11 FORMAT(' PHYTO SP. N.',I3,'(MG/M3):',G11.4,' VAR.RATE:',G11.4)
DO 6 IS=1,NZO
6 WRITE(*,15)IS,ZOOPL(IS,KG),ZOF(IS)
IF(OLPON.EQ.'NO')GOTO 4
ENH414=ENH4(KG)/14.
ENQ314=ENQ3(KG)/14.
PQ431=PQ4(KG)/31.
SIL28=SIL(KG)/28.
Q216=Q2(KG)/16.
WRITE(2,*) (PHYT(M,KG),M=1,N), (ZOOPL(L,KG),L=1,NZO), ENH414
1, ENQ314, PQ431, SIL28, Q216, TEMP(KG)
4 IF(KG.EQ.NZ)WRITE(*,*)'*****'
WRITE(*,*)'-----'
RETURN
END

```

```

FUNCTION CINE(X,SAT)
CINE=1.
IF((SAT+X).LE.0.)GOTO 1
CINE=X/(SAT+X)
1 RETURN
END

```

```

FUNCTION DAYL(FT,T)
GIORNO=24.
FTG3=FT*GIORNO+3.
DAYL=0.
T0=13.-FTG3/2.
T1=13.+FTG3/2.
IF(T.LT.T0.OR.T.GT.T1)GOTO 1
PIGR=3.1415
DUPIGR=6.283
CMPIGR=7.8537
AKA=DUPIGR/FTG3
FI=CMPIGR-6.283*13/FTG3
A=1.
DAYL=ABS(A*(1.+SIN(AKA*T+FI)))
RETURN
1
END

```

```

FUNCTION RADMX(JJ)
RADMX=(600.-340.*COS(6.283*FLOAT(JJ+10)/365.))* .85*.45
RETURN
END

```

```

SUBROUTINE UPRES(BAS,FNSI,FNP,FNO,KOPA,JESNO,FRAZ,FRAP,FRA5,
1K,U,PHYT,AZO,FOS,SIS,DT,N,NZO,KG,FRAO,QSS,ZOOPL,ROZ,RZ,RE,
2TEMP,CMO,D,FNH,ZNH,DPHYT,FT)
DIMENSION FNSI(1),FNP(1),UPHY(3),U(1),PHYT(3,5),ZOOPL(3,5)
1,ROZ(1),RE(1),TEMP(1),CMO(1),D(1),FNH(1),ZNH(1),DPHYT(1)
2,FNO(1)
COMMON/NUTR/ENH4(5),END3(5),P04(5),SIL(5),AZSW(5),PSW(5)
1,SISW(5),ENH(5),END(5),PD(5),SILI(5),ENHM(5,60),ENDM(5,60)
2,POM(5,60),SIM(5,60),T(60),ULIM(5)
COMMON/QSSI/Q2(5),OSW(5),O(5),O2M(5,60),FTE(3)
1,ZRE(3),FSED(5,60),ONIT(5),FRE(3)
DEMA=0.
DNH4=0.000001
FRES=0.
ZRES=0.
FOTOS=0.
DO 23 I=1,N
IF(DPHYT(I).LE.0.)GOTO 23
FOTOS=FOTOS+FTE(I)*PHYT(I,KG)*DPHYT(I)
CONTINUE
EZT=EXP(RZ*TEMP(KG))

```

23

```

DO 27 IE=1,NZO
DEMA=DEMA+ZNH(IE)*ZOOPL(IE,KG)*D(IE)*FT
27 ZRES=ZRES+ZRE(IE)*ROZ(IE)*EZT*ZOOPL(IE,KG)*FT
DO 20 IE=1,N
DEMA=DEMA+FNH(IE)*PHYT(IE,KG)*CMO(IE)*FT
FRES=FRES+FRE(IE)*RE(IE)*PHYT(IE,KG)*FT
IF(U(IE).LE.0.)GOTO 20
UPHY(IE)=U(IE)*PHYT(IE,KG)
DNH4=DNH4+UPHY(IE)*DT
20 CONTINUE
ENHNIT=ENH4(KG)*.03*EXP(.069*TEMP(KG))*FT
ENHNTO=ENHNIT*ONIT(KG)
DNH44=DNH4
DNH4=DNH4+ENHNIT
FRAZ=AZO/DNH4
IF(FRAZ.GT.1.)FRAZ=1.
DF04=0.000001
DO 21 IE=1,N
IF(U(IE).LE.0.)GOTO 21
DF04=DF04+UPHY(IE)*FNP(IE)*DT
21 CONTINUE
FRAP=FOS/DF04
IF(FRAP.GT.1.)FRAP=1.
DSIL=0.000001
DO 22 IE=1,N
IF(U(IE).LE.0.)GOTO 22
DSIL=DSIL+UPHY(IE)*FNSI(IE)*DT
22 CONTINUE
FRAS=SIS/DSIL
IF(FRAS.GT.1.)FRAS=1.
D022=.000001
DO 28 IE=1,N
IF(U(IE).LE.0.)GOTO 28
D022=D022+UPHY(IE)*FNO(IE)*DT
28 CONTINUE
D022=D2(KG)*FSED(KG,K)*FT+D022
D02=D022+ZRES+FRES+ENHNTO
FRA0=OSS/D02
IF(FRA0.GT.1.)FRA0=1.
BAS=MIN(FRAZ,FRAP,FRAS,FRA0)
IF(KOPA.NE.2)GOTO 19
BAS=1.
ENH4(KG)=DNH4
P04(KG)=DF04
SIL(KG)=DSIL
EN03(KG)=0.
O2(KG)=D02
GOTO 26
19 ENH4(KG)=ENH4(KG)-DNH44*BAS-ENHNIT*FRA0+DEMA
ENH(KG)=ENH4(KG)
EN03(KG)=EN03(KG)+ENHNIT*FRA0

```

```

IF (ENH4(KG),GE,ULIM(1)) THEN
  GOTO 9
ELSE
  ENH(KG)=0,
  ENO3(KG)=ENO3(KG)+ENH4(KG)
  IF (ENH4(KG),LT,0.)ENH4(KG)=0,
ENDIF
ENO(KG)=ENO3(KG)
IF (ENO3(KG),GE,ULIM(2)) THEN
  GOTO 9
ELSE
  ENO(KG)=0,
  IF (ENO3(KG),LT,0)ENO3(KG)=0,
ENDIF
IF (AZSW(KG),EQ,1.)GOTO 11
IF (JESNO,EQ,1)WRITE(*,*)' N NEED > AV,SUPPLY'
AZSW(KG)=1,
GOTO 11
9 AZSW(KG)=0
11 PO4(KG)=PO4(KG)-DP04*BAS
PO(KG)=PO4(KG)
IF (PO4(KG),GE,ULIM(3)) THEN
  GOTO 12
ELSE
  PO(KG)=0,
  IF (PO4(KG),LT,0.)PO4(KG)=0,
ENDIF
IF (PSW(KG),EQ,1.)GOTO 15
IF (JESNO,EQ,1)WRITE(*,*)' P NEED > AV,SUPPLY'
PSW(KG)=1,
GOTO 15
12 PSW(KG)=0,
15 SIL(KG)=SIL(KG)-DSIL*BAS
SILI(KG)=SIL(KG)
IF (SIL(KG),GE,ULIM(4)) THEN
  GOTO 16
ELSE
  SILI(KG)=0,
  IF (SIL(KG),LT,0.)SIL(KG)=0,
ENDIF
IF (SISW(KG),EQ,1.)GOTO 18
IF (JESNO,EQ,1)WRITE(*,*)' SI NEED > AV,SUPPLY'
SISW(KG)=1,
GOTO 18
16 SISW(KG)=0,
18 O2(KG)=O2(KG)-DO22*BAS-(ZRES+FRES+ENHNTD)*FRAQ+FOTDS
O(KG)=O2(KG)
IF (O2(KG),GE,ULIM(5)) THEN
  GOTO 24
ELSE
  O(KG)=0,

```

```

        IF (O2(KG).LT.0.)O2(KG)=0,
ENDIF
IF(OSW(KG).EQ.1.)GOTO 26
IF(JESNO.EQ.1)WRITE(*,*)' O2 NEED AV.SUPPLY'
OSW(KG)=1,
GOTO 26
24 OSW(KG)=0,
26 RETURN
END

```

SUBROUTINE PRESET

```

COMMON/VEGET/PHYT(3,5),PKN(3),PKP(3),PKSI(3),DESP(3),DTDEC(3)
1,U(3),UMAX(3),R(3),TMAX(3),PK1(3),ROPT(3),RE(3),CMD(3),PKO(3)
2,URI(3),FNP(3),FNSI(3),DPHYT(3),PHYFR(3),FNH(3),FNO(3)
COMMON/BESTIE/ZOOPL(3,5),FKCNV(3),FMAX(3,3),CMAX(3,3),FKF(
13),D(3),DZOO(3),ROZ(3),RZ,ZNH(3),ZO2(3),ZOF(3)
COMMON/DEPTH/NZ,TM(5),TSC(5),TEMP(5),THICK(5),ZOOT(5),PHP(5)
COMMON/NUTR/ENH4(5),EN03(5),PO4(5),SIL(5),AZSW(5),PSW(5)
1,SISW(5),ENH(5),EN0(5),PO(5),SILI(5),ENHM(5,60),ENOM(5,60)
2,POM(5,60),SIM(5,60),T(60),ULIM(5)
COMMON/OSSI/O2(5),OSW(5),O(5),O2M(5,60),FTE(3)
1,ZRE(3),FSED(5,60),ONIT(5),FRE(3)
COMMON/LUCE/BSP,ASP,CZ,RA1(5),RA2(5),RA3(5),ACMI,FOT
COMMON/PARAMS/DT,NFI,NOU,FT,ITA,DAY,HR,FRAZ,FRAP,FRAS,FRAO
DATA NFI,NPROF,NBELV,NUPAP,NUNUT/3,5,3,60,5/
FRAZ=1,
FRAP=1,
FRAS=1,
FRAO=1,
DT=0,
NFI=1
NOU=1
ITA=1
FT=0,
DAY=0,
HR=0,
DO 38 I=1,NUNUT
38 ULIM(I)=0.01
DO 36 I=1,NFIT
FTE(I)=0,
FRE(I)=0,
PKN(I)=0,
PKP(I)=0,
PKO(I)=0,
FNH(I)=0,
CMD(I)=0,
FNP(I)=0,
URI(I)=0,

```

```

PK1(I)=0.
PKSI(I)=0.
DESP(I)=0.
UMAX(I)=0.
TMAX(I)=0.
ROPT(I)=0.
FNSI(I)=0.
DTDEC(I)=0.
DPHYT(I)=0.
PHYFR(I)=0.
U(I)=0.
R(I)=0.
RE(I)=0.
36 DO 36 J=1,NPROF
   PHYT(I,J)=0.
   DO 30 I=1,NBELV
     FKCV(I)=0.
     FKF(I)=0.
     F0Z(I)=0.
     ZNH(I)=0.
     DZ00(I)=0.
     D(I)=0.
     ZRE(I)=0.
31 DO 31 J=1,NFIT
   FMAX(I,J)=0.
32 DO 32 J=1,NBELV
   CMAX(I,J)=0.
30 DO 30 J=1,NPROF
   Z00FL(I,J)=0.
   RZ=0.0693
   PA1(I)=0.
   RA2(I)=0.
   RA3(I)=0.
   DO 33 I=1,NPROF
     TM(I)=0.
     TSC(I)=0.
     PHP(I)=0.
     TEMP(I)=0.
     Z00T(I)=0.
     THICK(I)=0.
33 F0T=0.
   NZ=1
   DO 34 I=1,NPROF
     ENH4(I)=0.
     EN03(I)=0.
     AZSW(I)=0.
     SISW(I)=0.
     SILI(I)=0.
     F04(I)=0.
     SIL(I)=0.
     ENH(I)=0.

```

```

END(I)=0.
PO(I)=0.
O2(I)=0.
OSW(I)=0.
PSW(I)=0.
O(I)=0.
ONIT(I)=0.
DO 34 J=1,NUPAP
ENHM(I,J)=0.
ENOM(I,J)=0.
POM(I,J)=0.
SIM(I,J)=0.
O2M(I,J)=0.
34 FSED(I,J)=0.
ACMI=0.
BSP=0.00147/.85
ASP=0.05/.85
CZ=0.
DO 37 J=1,NUPAP
37 T(J)=0.
RETURN
END

```

Programma AQUATEMP5

Il programma AQTEMP5 calcola i parametri della curva:

$$\text{TEMP}(T) = \text{TM} - \text{TSC} * \text{COS}(6.283 * (T + F) / 365)$$

dove:

TEMP(T) e' il valore della temperatura (in gradi C) al tempo T (in giorni);

T e' la variabile indipendente tempo (in giorni);

TM e' il suo valor medio annuo (gradi C);

TSC e' la meta' della sua variazione annua, pari a meta' della distanza tra il massimo ed il minimo valore di temperatura nell'anno (in gradi C);

F e' una costante di fase che permette di centrare la curva su una data situazione reale.

L'utente fornira' al programma valori per TM, TSC ed F che gli sembrano coerenti, e vedra', mediante un grafico sul video, se la funzione che ne deriva descrive o meno il reale andamento della temperatura.

I valori finali dei tre parametri TM, TSC e F vengono dedotti dal confronto tra la curva teorica della temperatura cosi' ottenuta ed un gruppo di valori sperimentali della temperatura misurati durante un anno, considerando anche, ove disponibili, gli errori sperimentali. Questi dati misurati possono aiutare l'utente a fornire un valore di partenza sia a TM che a TSC; il programma fornisce anche, a chi non sappia dare una stima di partenza ad F, un suggerimento, che indica la data della misura del valore massimo sperimentale della temperatura (mese, giorno in numeri interi; es. il 15 di giugno e' 6,15).

Il programma quindi calcola una curva che abbia come parametri i valori dati dall'utente e la confronta con i dati misurati, fornendo anche un chi-quadrato che permette di stimare l'adattamento ed anche eventuali miglioramenti di fit, rispetto alla curva calcolata in precedenza.

I valori finali dei parametri, che soddisferanno l'utente, entreranno a far parte del data base per il modello, che sarà costruito mediante il programma AQINPT5.

```

PROGRAM AQTEMP5
COMMON SP(501,3),CC(501),AT(501),AR(501),SPU(501)
CHARACTER ATS*1,TAS*1,SIS*1,FINOM*B,MEGS*6,PEGS*5
DIMENSION DESE(12)
DATA DESE/180.,145.,110.,90.,35.,0.,-30.,-60.,-90.,-120.,
*-150.,-180./
WRITE(*,*)' Program AQTEMP-to fit a temperature data set'
15 WRITE(*,*)' Type n,points,time step(in days)-to stop type 0,0'
READ(*,*)NT,DT
WRITE(*,*)' N.of points='*NT,' time step='*DT
ATS='N'
OCHI=1.E+30
X0=0.
IF((NT*DT).EQ.0.)GOTO 14
WRITE(*,*)'Print tabulated function-Y/N'
READ(*,*(A))TAS
WRITE(*,*)' type Tm,Tsc,F for T(k)=Tm-Tsc*cos(6.283*(k+F)/365)'
WRITE(*,*)' for first estimate of F,type 999 for it'
READ(*,*)TM,TSC,F
WRITE(*,*)' Tm='*TM,' Tsc='*TSC,' F='*F
IF(F.NE.999.)GOTO 16
WRITE(*,*)' type month & day numbers of Tm measurement'
READ(*,*)DONU,DANU
DIAU=DONU-.5+DANU/30.
IF(DIAU.GT.12.)DIAU=12.
DO 17 L=1,11
IF(DIAU.GE.FLOAT(L).AND.DIAU.LE.FLOAT(L+1))GOTO 18
17 CONTINUE
L=11
18 F=(DIAU-FLOAT(L))*(DESE(L+1)-DESE(L))+DESE(L)
WRITE(*,*)' First suggestion for F!':F
16 DO 1 K=1,NT+1
X=X0+FLOAT(K-1)*DT

```

```

CC(K)=TM-TSC*COS(2*3.1415*(X+F)/365)
IF(TAS.EQ.'Y')WRITE(*,*)' time=',X,' temp(xC)=',CC(K)
1 CONTINUE
IF(ATS.EQ.'Y')GOTO 2
WRITE(*,*)'give n. of experimental triplets '
READ(*,*)NSP
WRITE(*,*)'exp. data from filename/NO'
READ(*, '(A)')FINOM
IF(FINOM.EQ.'NO')GOTO 4
WRITE(*,3)FINOM
3 FORMAT(' data from file:',A8)
OPEN(1,FILE=FINOM,STATUS='OLD')
DO 5 I=1,NSP
READ(1,*)SP(I,1),SP(I,2),SP(I,3)
IF(SP(I,3).EQ.0.)SP(I,3)=.1*SP(I,2)
IF(TAS.EQ.'Y')WRITE(*,*)SP(I,1),SP(I,2),SP(I,3)
5 CONTINUE
CLOSE(1)
2 IF(DT.NE.0)GOTO 7
SPMA=SP(1,1)
SPMI=SPMA
DO 6 I=2,NSP
IF(SP(I,1).GT.SPMA)SPMA=SP(I,1)
IF(SP(I,1).LT.SPMI)SPMI=SP(I,1)
6 CONTINUE
DT=(SPMA-SPMI+9.999999E-21)/(NT-1)
7 DO 8 I=1,NSP
8 SPU(I)=(SP(I,1)-X0)/DT
GOTO 9
4 IF(ATS.EQ.'Y')GOTO 2
OPEN(1,FILE='TEMPE.DAT',STATUS='NEW')
DO 10 I=1,NSP
WRITE(*,*)' *** give experimental triplet n.',I
READ(*,*)SP(I,1),SP(I,2),SP(I,3)
WRITE(*,*)'abscissa is ',SP(I,1)
WRITE(*,*)'function value is ',SP(I,2)
WRITE(*,*)'error estimate is ',SP(I,3)
IF(SP(I,3).EQ.0.)SP(I,3)=.1*SP(I,2)
IF(TAS.EQ.'Y')WRITE(*,*)SP(I,1),SP(I,2),SP(I,3)
10 WRITE(1,*)SP(I,1),SP(I,2),SP(I,3)
WRITE(*,*)' now exp. data are on file TEMPE.DAT'
CLOSE(1)
GOTO 2
9 DO 11 I=1,NSP
DO 12 K=1,NT-1
ILIM=X0+(K-1)*DT
IILIM=ILIM+DT
SPI1=SPU(I)*DT+X0
IF(SPI1.GE.IILIM.AND.SPI1.LE.IILIM)GOTO 11
12 CONTINUE
K=NT-1

```

```

11  AT(I)=CC(K)+(SPU(I)*DT+X0-ILIM)/DT*(CC(K+1)-CC(K))
    CHI2=0.
    MEGS='better'
    PEGS='worse'
    DO 13 I=1,NSP
    NUM=(SP(I,2)-AT(I))*2
    DENOM=SP(I,3)**2
    IF(DENOM.EQ.0.)DENOM=.000001
    CHI2=CHI2+NUM/DENOM
13  CONTINUE
    WRITE(*,*)' chi2=',CHI2,' for ',NSP-1,' DEGREES OF FREEDOM'
    IF((CHI2-QCHI).LT.0.)THEN
        WRITE(*,'(1X,A6)')MEGS
    ELSE
        WRITE(*,'(1X,A5)')PEGS
    ENDIF
    QCHI=CHI2
    CALL PROBH0(ATS,NSP,CHI2)
    CALL PLOTTA(SIS,NT,NSP)
    WRITE(*,*)'another trial-Y/N '
    READ(*,'(A)')ATS
    IF(ATS.EQ.'N')GOTO 14
    GOTO 15
14  STOP
    END

```

```

SUBROUTINE PLOTTA(SIS,NT,NSP)
COMMON SP(501,3),CC(501),AT(501),AR(501),SPU(501)
CHARACTER PLOT(21,71)
CHARACTER SIS*1
REAL MX,MN
DO 6 I=1,21
DO 6 J=1,71
6  PLOT(I,J)=' '
WRITE(*,*)'plot of functions-Y/N '
READ(*,'(A)')SIS
IF(SIS.EQ.'N')GOTO 1
RF=1.
WRITE(*,*)'any reduction factor-Y/N '
READ(*,'(A)')SIS
IF(SIS.EQ.'Y')WRITE(*,*)' then type it'
IF(SIS.EQ.'Y')READ(*,*)RF
MX=CC(1)
MN=MX
DO 2 K=2,NT
IF(CC(K).GT.MX)MX=CC(K)
IF(CC(K).LT.MN)MN=CC(K)
2  CONTINUE

```

```

DO 3 I=1,NSP
SOPRA=SP(I,2)+SP(I,3)
SOTTO=SP(I,2)-SP(I,3)
IF(SOPRA.GT.MX)MX=SOPRA
IF(SOTTO.LT.MN)MN=SOTTO
3 CONTINUE
DO 7 I=1,21
PLOT(I,1)='!'
PLOT(I,71)='!'
PLOT(1,I)='- '
7 PLOT(21,I)='- '
DO 8 I=22,71
PLOT(1,I)='- '
8 PLOT(21,I)='- '
IF(MN.GT.0.)MN=0.
FX=70./FLOAT(NT)
DY=MX-MN
IF(DY.EQ.0.)DY=1.E-09
FY=20.*RF/DY
Y0=21.+MN*FY
ITY=Y0+.1
DO 9 I=1,71
9 PLOT(ITY,I)='- '
X0=2.
ITX=X0+.1
DO 10 I=1,21
10 PLOT(I,ITX)='.'
X1=2.
Y1=21.-(CC(1)-MN)*FY
ITX=X1+.1
ITY=Y1+.1
PLOT(ITY,ITX)='*'
DO 4 K=2,NT
X=FLOAT(K-1)*FX+2.
Y=21.-(CC(K)-MN)*FY
IXT=X+.1
IYT=Y+.1
PLOT(IYT,IXT)='*'
X1=X
Y1=Y
4 CONTINUE
DO 5 I=1,NSP
X=SP(I)*FX+2.
Y1=21.-(SP(I,2)-SP(I,3))*FY
Y2=21.-(SP(I,2)+SP(I,3))*FY
Y=21.-SP(I,2)*FY
JY1=Y1+.1
JY2=Y2+.1
ITX=X+.1
ITY=Y+.1
DO 12 J=JY2,JY1

```

```

12 PLOT(J,ITX)='I'
C  PLOT(ITY,ITX)='O'
5  CONTINUE
   WRITE(*,*)'Plot to printer'
   WRITE(*,*)
   DO 13 I=1,21
13  WRITE(*,14)(PLOT(I,J),J=1,71)
14  FORMAT(1X,71A1)
   WRITE(*,*)
1  RETURN
   END

```

```

SUBROUTINE PROBH0(ATS,NSP,CHI2)
COMMON SP(501,3),CC(501),AT(501),AR(501),SPU(501)
CHARACTER ATS*1
INTEGER D
IF(ATS.EQ.'Y')GOTO 1
NPAS=500
D=NSP-1
IF(D.EQ.0)RETURN
RADN=SQRT(2.*FLOAT(D))
UL=FLOAT(D)-7.*RADN
ALI=0
IF(ALI.LT.UL)ALI=UL
IF(D.EQ.1)ALI=.0642
ALS=FLOAT(D)+7.*RADN
HH=(ALS-ALI)/FLOAT(NPAS)
AR(1)=0
IF(D.EQ.1)AR(1)=.19999
WRITE(*,*)' tabulating X2 distribution'
WRITE(*,*)' for ',D,' degrees of freedom'
X=ALI+.000001
CALL CHIDUE(D,X,E)
TI=E
DO 3 II=1,NPAS
X=X+HH
CALL CHIDUE(D,X,E)
TII=E
DIF=(TII+TI)/2.*HH
TI=TII
AR(II+1)=AR(II)+DIF
3  CONTINUE
1  IF(CHI2.LE.(0.0642).AND.D.EQ.1)GOTO 2
X=ALI+.000001
DO 4 II=1,NPAS
X=X+HH
IF(CHI2.GE.(X-HH).AND.CHI2.LE.X)GOTO 5
4  CONTINUE

```

```

PROB=.999999
  GOTO 6
2  WRITE(*,*)' prob .GT. 80%'
5  PROB=((AR(II+1)-AR(II))*(CHI2-X+HH)/HH+AR(II))
6  PROB=(1.-PROB)*100,
  WRITE(*,*)
  WRITE(*,*)' *** prob.(%)=',PROB
  WRITE(*,*)
  RETURN
END

```

```

SUBROUTINE CHIDUE(D,X,E)
  INTEGER D
  A=-.693147+(FLOAT(D)/2.-1.)*LOG(X/2.)
  B=-X/2.
  C=0.
  IF((FLOAT(D)/2.).GT.FLOAT(D/2))GOTO 1
2  DO 2 I=1,(D/2-1)
  C=C+ALOG(FLOAT(I))
  GOTO 3
1  U=SQRT(3.14159)
  I=1
  C=ALOG(U)
4  V=(FLOAT(D)/2.-FLOAT(I))
  IF(V.LT.0.)GOTO 3
  C=C+ALOG(V)
  I=I+1
  GOTO 4
3  E=EXP(A-C+B)
  RETURN
END

```

Programma AQINPT5

Questo programma permette a chi voglia usare il modello di crearsi (opzione CREATE) il data-base per il modello medesimo senza dover conoscere l'editor della macchina su cui sta lavorando. Naturalmente, se questa gli e' ben nota, egli usera' il programma o l'editor di macchina indifferentemente, specie nel fare correzioni ad una data-base gia' costruito e che vada aggiornato.

Il programma AQINPT5 permette pero' anche l'aggiornamento (opzione UPDATE) di un data-base gia' esistente, con il vantaggio di poter dare anche un nome alle grandezze da quantificare o da modificare, e non solo il loro contenuto numerico, il che costituisce un chiaro vantaggio rispetto ad un editor.

```
PROGRAM AQINPT5
PARAMETER (NUTM=732)
DIMENSION T(NUTM)
CHARACTER NS*8,CUS*1,OLPON*8
1 WRITE(*,*)' PROG.AQINPT,TO CREATE OR UPDATE A FILE FOR AQMODL'
WRITE(*,*)' TYPE PARAMETERS AND DATA FILENAME-MAX.8 CHARS'
WRITE(*,*)' TO STOP TYPE N'
READ(*,'(A8)')NS
IF(NS.EQ.'N')GOTO 3
WRITE(*,2)NS
2 FORMAT(' FILENAME FOR PARAMETERS AND DATA:',A8)
WRITE(*,*)' CREATION OR UPDATING-C/U '
READ(*,'(A)')CUS
IF(CUS.EQ.'U')CALL UPDA(T,NUTM,NS,OLPON)
IF(CUS.EQ.'C')CALL CREA(T,NUTM,NS,OLPON)
GOTO 1
3 STOP
END
```

```

SUBROUTINE CREA(T,NUTM,NS,OLPON)
DIMENSION T(NUTM)
CHARACTER NS*8,OLPON*8
OPEN(1,FILE=NS,STATUS='NEW')
REWIND 1
WRITE(*,*)' TYPE N.OF PHYTOSPECIES,N.OF ZOOSPECIES '
READ(*,*)N,NZO
WRITE(1,*)N,NZO
WRITE(*,*)' TYPE N.OF DEPTHS,TIME INTEGRATION STEP '
READ(*,*)NZ,DT
KG=NZ
WRITE(*,*)' TYPE N.OF NUTRIENTS QUADRUPLETS AT EACH DEPTH'
READ(*,*)NU
WRITE(1,*)NZ,DT,NU
WRITE(*,*)'TYPE LIGHT FACTOR- STEELE(1)/DITORO(2)/PLATT(3)'
READ(*,*)LUX
WRITE(1,*)LUX
WRITE(*,*)'TYPE EXTIN.COEFF.FORMULA-RILEY(1)/WALSH(2)'
READ(*,*)KEST
WRITE(1,*)KEST
WRITE(*,*)'TYPE PLOT-DATA FILENAME(MAX.8 CHARS)/NO'
READ(*,*(A8))' )OLPON
WRITE(1,*(A8))' )OLPON
WRITE(*,*)'TYPE OPTION FOR SUPPLY LIMITS PRINTOUT-Y(1)/N(0)'
READ(*,*)JESNO
WRITE(1,*)JESNO
WRITE(*,*)'TYPE MIN.ACCL.LIGHT LEVEL-IF UNKNOWN,TYPE 40'
READ(*,*)ACMI
WRITE(1,*)ACMI
WRITE(*,*)'TYPE SWITCHING LEVEL FROM NH4 TO NO3'
READ(*,*)ULIM
WRITE(1,*)ULIM
WRITE(*,*)'TYPE NUTRIENTS OPTION-MEASURED CONCENTRATIONS(1)'
WRITE(*,*)' -RESIDUALS (2)'
WRITE(*,*)' -DAILY INPUT RATES (3)'
READ(*,*)KOPA
WRITE(1,*)KOPA
WRITE(*,*)'TYPE PHASE CONST.FOR TEMPERATURE(FROM AQTEMP)'
READ(*,*)FIMSE
WRITE(1,*)FIMSE
WRITE(*,*)'TYPE MONTH,DAY NUMBERS OF STARTING T FOR SIMUL.'
READ(*,*)ILA,ILO
WRITE(1,*)ILA,ILO
WRITE(*,*)' TIME INTERVAL FOR OUTPUT,IN HOURS '
READ(*,*)OUTINT
WRITE(1,*)OUTINT
WRITE(*,*)' COMPUTATION TIME SPAN,IN DAYS '
READ(*,*)FINTIM
NFI=INT(FINTIM*24./DT+.5)
WRITE(1,*)FINTIM
DO 1 I=1,NU

```

```

1 WRITE(*,*)'TYPE TIME(IN DAYS) OF MEASURE N,',I
  READ(*,*)T(I)
  DO 2 I=1,N
    WRITE(*,*)'TYPE RESPIRATION AND MORTALITY FOR PHYTO SP',I
    READ(*,*)RE,RAMO
    WRITE(*,*)'TYPE MAX.GROWTH TEMP.,GROWTH CONST.FOR PHYTO SP',I
    READ(*,*)TMAX,PK1
    WRITE(*,*)'TYPE ALSO O2 CONS.RATE IN RESP.,O2 PROD.RATE'
    READ(*,*)FRE,FTE
    WRITE(1,*)RE,RAMO,TMAX,PK1,FRE,FTE
    WRITE(*,*)'TYPE HALF SAT.CONSTANTS(N,P,SI,O2)OF PHYTO SP',I
    READ(*,*)PKN,PKP,PKSI,PKO
2    WRITE(1,*)PKN,PKP,PKSI,PKO
    DO 3 I=1,KG
      WRITE(*,*)'TYPE(IN METERS)DEPTH N,',I
      READ(*,*)THICK
      WRITE(*,*)'TYPE TM('I,') AND TSC('I,')'
      READ(*,*)TM,TSC
      WRITE(*,*)'TYPE O2/NH4 RATE IN NH4 OXIDATION'
      READ(*,*)ONIT
      WRITE(1,*)THICK
      WRITE(1,*)TM,TSC,ONIT
      DO 4 J=1,N
        WRITE(*,*)'TYPE STARTING VALUE FOR PHYTO SP.',J,' AT M.',THICK
4        READ(*,*)PHYTO
        WRITE(1,*)PHYTO
        DO 5 J=1,NZO
          WRITE(*,*)'TYPE STARTING VALUE FOR ZOO SP.',J,' AT M.',THICK
5          READ(*,*)ZOO
          WRITE(1,*)ZOO
          DO 6 K=1,NU
            WRITE(*,*)'TYPE VALUES FOR NH4,NO3,P04,SI,O2(MMOL/M3),O2 CONS. '
            WRITE(*,*)'RATE BY SEDIM. AT DEPTH ',THICK,' AT TIME ',T(K)
            READ(*,*)HN4,ON3,P04,SIL,O2,FSED
            HN4=HN4*14.
            ON3=ON3*14.
            P04=P04*31.
            SIL=SIL*28.
            O2=O2*16.
6            WRITE(1,*)T(K),HN4,ON3,P04,SIL,O2,FSED
3            CONTINUE
            DO 7 J=1,NZO
              WRITE(*,*)'TYPE GRAZING H-SAT. CONST,PREYING H,SAT. CONSTANTS'
              WRITE(*,*)'FOR ZOO SP.',J
              READ(*,*)FK,CK
              WRITE(*,*)'TYPE MORTALITY AND RESPIRATION RATES '
              READ(*,*)D,B
              WRITE(*,*)'TYPE O2 CONS.RATE IN RESP.,DEAD MATT. FRACTION TO NH4'
              WRITE(*,*)'RATE AND 50% SURVIVAL O2 CONC.'
              READ(*,*)ZRE,ZNH,ZO2
              WRITE(1,*)FK,CK,D,B,ZRE,ZNH,ZO2

```

```

DO 9 I=1,N
WRITE(*,*)'TYPE GRAZING RATE OF ZOO SP.',J,' ON PHYTO SP.',I
READ(*,*)FMAX
9 WRITE(1,*)FMAX
DO 9 I=1,NZO
WRITE(*,*)'TYPE PREYING RATE OF ZOO SP.',J,' ON ZOO SP.',I
9 READ(*,*)CMAX
7 WRITE(1,*)CMAX
CONTINUE
DO 10 I=1,N
WRITE(*,*)'TYPE PHYTO SP.',I,' -NP FACTOR'
READ(*,*)DNP
WRITE(*,*)'TYPE PHYTO SP.',I,' -NSI FACTOR'
READ(*,*)DNSI
WRITE(*,*)'TYPE PHYTO SP.',I,' -NO FACTOR'
READ(*,*)DNOZ
WRITE(*,*)'TYPE DEAD MATTER FRACTION TO NH4 FOR PHYTO SP.',I
READ(*,*)FNH
WRITE(1,*)DNP
WRITE(1,*)DNSI
WRITE(1,*)DNOZ
10 WRITE(1,*)FNH
CLOSE(1)
WRITE(*,*)'DATA AND PARAMETERS ON FILE ',NS
RETURN
END

```

```

SUBROUTINE UPDA(T,NUTM,NS,OLPON)
DIMENSION T(NUTM)
CHARACTER NS*8,NNS*8,US*1,OLPON*8
WRITE(*,*)' TYPE FILENAME FOR UPDATED FILE-MAX.8 CHARS'
READ(*,'(A8)')NNS
OPEN(1,FILE=NS,STATUS='OLD')
OPEN(2,FILE=NNS,STATUS='NEW')
REWIND 1
REWIND 2
READ(1,*)N,NZO
WRITE(*,*)' N.OF PHYTOSPECIES:',N
WRITE(*,*)'N.OF ZOOSPECIES:',NZO
WRITE(*,*)'ANY CHANGE-Y/N'
READ(*,'(A)')US
IF(US.EQ.'N')GOTO 1
WRITE(*,*)' CHANGE N.PHYTOSPECIES-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE'
IF(US.EQ.'Y')READ(*,*)N
WRITE(*,*)' CHANGE N.ZOOSPECIES-Y/N '
READ(*,'(A)')US

```

```

IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)NZD
1 READ(1,*)NZ,DT,NU
WRITE(*,2)NZ,DT
2 FORMAT(' N. OF DEPTHS=',I4/' COMPUTATION TIME STEP DT=',F5.1)
WRITE(*,*)' ANY CHANGE-Y/N '
READ(*,'(A)')US
IF(US.EQ.'N')GOTO 3
WRITE(*,*)' CHANGE N.OF DEPTHS-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)NZ
WRITE(*,*)' CHANGE DT-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)DT
3 KG=NZ
WRITE(2,*)N,NZD
WRITE(*,*)' N.OF MEASURED NUTRIENT DATA GROUPS IS ',NU
WRITE(*,*)' NO CHANGE OF THIS PARAMETER IS ALLOWED'
WRITE(2,*)NZ,DT,NU
READ(1,*)LUX
WRITE(*,*)' LIGHT FACTOR OPTION IS ',LUX
WRITE(*,*)' ANY CHANGE-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW OPTION'
IF(US.EQ.'Y')READ(*,*)LUX
WRITE(2,*)LUX
READ(1,*)KEST
WRITE(*,*)' EXT.COEFF.OPTION IS ',KEST
WRITE(*,*)' ANY CHANGE-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW OPTION'
IF(US.EQ.'Y')READ(*,*)KEST
WRITE(2,*)KEST
READ(1,'(A8)')OLPON
WRITE(*,'(1X,9H PLOT FILE,A8)')OLPON
WRITE(*,*)' ANY CHANGE-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW NAME'
IF(US.EQ.'Y')READ(*,'(A8)')OLPON
WRITE(2,'(A8)')OLPON
READ(1,*)JESNO
WRITE(*,*)' SUPPLY LIMITS PRINTOUT OPTION IS ',JESNO
WRITE(*,*)' ANY CHANGE-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW OPTION'
IF(US.EQ.'Y')READ(*,*)JESNO
WRITE(2,*)JESNO
READ(1,*)ACMI
WRITE(*,*)' LIGHT ACCLIMATION LEVEL IS ',ACMI

```

```

WRITE(*,*)' ANY CHANGE-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW LEVEL'
IF(US.EQ.'Y')READ(*,*)ACMI
WRITE(2,*)ACMI
READ(1,*)ULIM
WRITE(*,*)' NH4 TO NO3 SWITCHING LEVEL IS ',ULIM
WRITE(*,*)' ANY CHANGE-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW LEVEL'
IF(US.EQ.'Y')READ(*,*)ULIM
WRITE(2,*)ULIM
READ(1,*)KOPA
WRITE(*,*)' NUTRIENT MODE OF INPUT IS ',KOPA
WRITE(*,*)' ANY CHANGE-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW MODE'
IF(US.EQ.'Y')READ(*,*)KOPA
WRITE(2,*)KOPA
READ(1,*)FIMESF
WRITE(*,*)' TEMPERATURE FUNCTION PHASE CONSTANT IS ',FIMESF
WRITE(*,*)' ANY CHANGE-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE'
IF(US.EQ.'Y')READ(*,*)FIMESF
WRITE(2,*)FIMESF
READ(1,*)ILA,ILO
WRITE(*,*)' MONTH & DAY NUMBERS FOR SIMUL. START ARE ',ILA,ILO
WRITE(*,*)' ANY CHANGE-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUES'
IF(US.EQ.'Y')READ(*,*)ILA,ILO
WRITE(2,*)ILA,ILO
READ(1,*)OUTINT
WRITE(*,*)' TIME INTERVAL FOR OUTPUT, IN HOURS ',OUTINT
WRITE(*,*)' CHANGE IT-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)OUTINT
WRITE(2,*)OUTINT
READ(1,*)FINTIM
WRITE(*,*)' COMPUTATION TIME SPAN, IN DAYS ',FINTIM
WRITE(*,*)' CHANGE IT-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)FINTIM
WRITE(2,*)FINTIM
DO 4 I=1,N
WRITE(*,*)' PHYTOSPECIES N. ',I
READ(1,*)RE,RAMO,TMAX,PK1,FRE,FTE
WRITE(*,*)' RESPIR., MORTALITY, TMAX, GROWTH C., RESP.O2 FRACT., '

```

```

WRITE(*,*)'O2 PROD.RATE ARE ',RE,RAMO,TMAX,PK1,FRE,FTE
WRITE(*,*)' ANY CHANGE-Y/N '
READ(*,'(A)')US
IF(US.EQ.'N')GOTO 5
WRITE(*,*)' CHANGE RESPIR.-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)RE
WRITE(*,*)' CHANGE MORTAL.-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)RAMO
WRITE(*,*)' CHANGE TMAX.-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)TMAX
WRITE(*,*)' CHANGE GROWTH C.-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)PK1
WRITE(*,*)' CHANGE RESP.O2 FRACTION-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW FRACTION'
IF(US.EQ.'Y')READ(*,*)FRE
WRITE(*,*)' CHANGE O2 PROD.RATE-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW RATE'
IF(US.EQ.'Y')READ(*,*)FTE
5 WRITE(2,*)RE,RAMO,TMAX,PK1,FRE,FTE
READ(1,*)PKN,PKP,PKSI,PKO
WRITE(*,*)'HALF SATURAT.CONST. OF PHYTO SP',I
WRITE(*,*)'FOR N,P,SI,O2:',PKN,PKP,PKSI,PKO
WRITE(*,*)' ANY CHANGE-Y/N '
READ(*,'(A)')US
IF(US.EQ.'N')GOTO 4
WRITE(*,*)' CHANGE N SAT.C.-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)PKN
WRITE(*,*)' CHANGE P SAT.C.-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)PKP
WRITE(*,*)' CHANGE SI SAT.C.-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)PKSI
WRITE(*,*)' CHANGE O2 SAT.C.-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE'
IF(US.EQ.'Y')READ(*,*)PKO

```

```

4  WRITE(2,*)PKN,PKP,PKSI,PKD
   DO 6 I=1,KG
   READ(1,*)THICK
   READ(1,*)TM,TSC,ONIT
   WRITE(*,7)I,THICK
7  FORMAT(' *** DEPTH N.',I4,' IS ',G11.4,' METERS')
   WRITE(*,*)' CHANGE IT-Y/N '
   READ(*,'(A)')US
   IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE FOR DEPTH '
   IF(US.EQ.'Y')READ(*,*)THICK
   WRITE(*,8)I,I,TM,TSC,ONIT
8  FORMAT(' TM(',I4,'),'TSC(',I4,') AND NH4/O2 REL. ARE ',3G11.4)
   WRITE(*,*)' ANY CHANGE-Y/N '
   READ(*,'(A)')US
   IF(US.EQ.'N')GOTO 9
   WRITE(*,*)' CHANGE AVER. TEMP.-Y/N '
   READ(*,'(A)')US
   IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
   IF(US.EQ.'Y')READ(*,*)TM
   WRITE(*,*)' CHANGE TEMP. FLUCT.-Y/N '
   READ(*,'(A)')US
   IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
   IF(US.EQ.'Y')READ(*,*)TSC
   WRITE(*,*)' CHANGE NH4/O2 RELATION-Y/N'
   READ(*,'(A)')US
   IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE'
   IF(US.EQ.'Y')READ(*,*)ONIT
9  WRITE(2,*)THICK
   WRITE(2,*)TM,TSC,ONIT
   DO 10 J=1,N
   READ(1,*)PHYTO
   WRITE(*,*)'STARTING VALUE FOR PHYTO SP.',J
   WRITE(*,*)'IS ',PHYTO
   WRITE(*,*)' CHANGE IT-Y/N '
   READ(*,'(A)')US
   IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
   IF(US.EQ.'Y')READ(*,*)PHYTO
10 WRITE(2,*)PHYTO
   DO 11 J=1,NZD
   READ(1,*)ZOO
   WRITE(*,*)'STARTING VALUE FOR ZOOSPECIES ',J
   WRITE(*,*)'IS ',ZOO
   WRITE(*,*)' CHANGE IT-Y/N '
   READ(*,'(A)')US
   IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
   IF(US.EQ.'Y')READ(*,*)ZOO
11 WRITE(2,*)ZOO
   WRITE(*,*)' ANY CHANGE IN NUTR.-Y/N '
   READ(*,'(A)')US
   IF(US.EQ.'N')GOTO 12
   DO 13 K=1,NU

```

```

READ(1,*)T(K),HN4,ON3,P04,SIL,O2,FSED
HN4=HN4/14.
ON3=ON3/14.
P04=P04/31.
SIL=SIL/28.
O2=O2/16.
WRITE(*,*)'VALUES FOR T,NH4,N03,P04,SI,O2(MMOL/M3),O2 TO SED.RATE'
WRITE(*,*)HN4,ON3,P04,SIL,O2,FSED
WRITE(*,*)' ANY CHANGE-Y/N '
READ(*,'(A)')US
IF(US.EQ.'N')GOTO 14
WRITE(*,*)' CHANGE TIME-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)T(K)
WRITE(*,*)' CHANGE NH4-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)HN4
WRITE(*,*)' CHANGE N03-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)ON3
WRITE(*,*)' CHANGE P04-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)P04
WRITE(*,*)' CHANGE SIL-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)SIL
WRITE(*,*)' CHANGE O2-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)O2
WRITE(*,*)' CHANGE O2 RATE TO SEDIMENT-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW RATE '
IF(US.EQ.'Y')READ(*,*)FSED
14  HN4=HN4*14.
    ON3=ON3*14.
    P04=P04*31.
    SIL=SIL*28.
    O2=O2*16.
13  WRITE(2,*)T(K),HN4,ON3,P04,SIL,O2,FSED
    GOTO 6
12  DO 15 K=1,NU
    READ(1,*)T(K),HN4,ON3,P04,SIL,O2,FSED
15  WRITE(2,*)T(K),HN4,ON3,P04,SIL,O2,FSED
6    CONTINUE
    DO 16 J=1,NZO

```

```

READ(1,*)FK,CK,D,B,ZRE,ZNH,ZO2
WRITE(*,17)J,FK,CK,D,B,ZRE,ZNH,ZO2
17  FORMAT(' ZOOSP.',I4/' GRAZING & PREYING SAT.CONST.',2G11.4/
1' MORTALITY ',G11.4/' RESPIRATION ',G11.4/' RESP.O2 CONS.',
2G11.4/' DEAD MATT.TO NH4 FRACTION',G11.4/' 50% SURV.CONC.',
3G11.4)
WRITE(*,*)' ANY CHANGE-Y/N '
READ(*,'(A)')US
IF(US.EQ.'N')GOTO 18
WRITE(*,*)' CHANGE GRAZING S.C.-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)FK
WRITE(*,*)' CHANGE PREYING S.C.-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)CK
WRITE(*,*)' CHANGE MORTALITY-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)D
WRITE(*,*)' CHANGE RESPIRATION-Y/N '
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)B
WRITE(*,*)' CHANGE O2 RESP.CONSUM.-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)ZRE
WRITE(*,*)' CHANGE DEAD MATTER TO NH4 FRACTION-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)ZNH
WRITE(*,*)' CHANGE O2 CONC. FOR 50% SURVIVAL-Y/N'
READ(*,'(A)')US
IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
IF(US.EQ.'Y')READ(*,*)ZO2
18  WRITE(2,*)FK,CK,D,B,ZRE,ZNH,ZO2
DO 19 I=1,N
READ(1,*)FMAX
WRITE(*,20)FMAX,J,I
20  FORMAT(' GRAZING RATE=',G11.4,' OF ZOOSP.',I4,' ON PHYTO.',I4)
WRITE(*,*)' ANY CHANGE-Y/N '
READ(*,'(A)')US
IF(US.EQ.'N')GOTO 19
WRITE(*,*)' TYPE NEW VALUE '
READ(*,*)FMAX
19  WRITE(2,*)FMAX
DO 21 I=1,NZO
READ(1,*)CMAX
WRITE(*,22)CMAX,J,I

```

```

22  FORMAT(' PREYING RATE:',G11.4,' OF ZOOSP.',I4,' ON ZOOSP.',I4)
    WRITE(*,*)' ANY CHANGE-Y/N '
    READ(*,'(A)')US
    IF(US.EQ.'N')GOTO 21
    WRITE(*,*)' TYPE NEW VALUE '
    READ(*,*)CMAX
21  WRITE(2,*)CMAX
16  CONTINUE
    DO 23 I=1,N
    READ(1,*)DNP
    READ(1,*)DNSI
    READ(1,*)DNO2
    READ(1,*)FNH
    WRITE(*,*)' PHYTO SP.',I,'-NP FACTOR ',DNP
    WRITE(*,*)' CHANGE IT-Y/N '
    READ(*,'(A)')US
    IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
    IF(US.EQ.'Y')READ(*,*)DNP
    WRITE(*,*)' PHYTO SP.',I,'-NSI FACTOR ',DNSI
    WRITE(*,*)' CHANGE IT-Y/N '
    READ(*,'(A)')US
    IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
    IF(US.EQ.'Y')READ(*,*)DNSI
    WRITE(*,*)' PHYTO SP.',I,'-NO2 FACTOR ',DNO2
    WRITE(*,*)' CHANGE IT-Y/N'
    READ(*,'(A)')US
    IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
    IF(US.EQ.'Y')READ(*,*)DNO2
    WRITE(*,*)' PHYTO SP.',I,' DEAD MATTER TO NH4 FRACTION ',FNH
    WRITE(*,*)' CHANGE IT-Y/N'
    READ(*,'(A)')US
    IF(US.EQ.'Y')WRITE(*,*)' TYPE NEW VALUE '
    IF(US.EQ.'Y')READ(*,*)FNH
    WRITE(2,*)DNP
    WRITE(2,*)DNSI
    WRITE(2,*)DNO2
23  WRITE(2,*)FNH
    CLOSE(1)
    CLOSE(2)
    WRITE(*,*)'UPDATED DATA ON FILE ',NNS
    RETURN
    END

```

Programma AQPLOTS

Questo programma fornisce un semplice grafico su video (a bassa risoluzione) dell'andamento delle quantità biologiche e chimiche calcolate dal modello e scritte su file dalla subroutine QUT (mentre INIT vi ha scritto i valori iniziali).

Le grandezze possono essere viste assieme o singolarmente od a sottogruppi, come si vuole. La prima opzione riguarda la scelta se vedere le quantità biologiche (fito e zoospecie) o quelle chimiche (nutrienti). Nel primo caso si deve ancora da scegliere se si vuol vedere le curve delle fitospecie o delle zoospecie.

Il grafico prodotto è un tipico grafico di lavoro, un po' rozzo ma veloce, adatto anche alle macchine che non possono effettuare la riproduzione del video sulla stampante (dump graphics). Tutte le quantità calcolate dal modello vengono però stampate in formato compatto, se richiesto, così che l'utente potrà poi farsi il grafico fino, di quelle che saranno considerate più interessanti. Se il computer su cui si lavora ha un pacchetto grafico accessibile al FORTRAN e mezzi per riprodurre i grafici ad alta risoluzione su carta, sarà sua cura scrivervi una subroutine adatta allo scopo. Noi non l'abbiamo fatto perché i pacchetti grafici sono in genere molto legati alla particolare macchina, a differenza del FORTRAN.

Ad un certo punto il programma domanda se nei risultati del modello, scritti sul file per la grafica, vi siano una serie di doppie scritture o no. Si risponde Y (yes) se l'intervallo di scrittura imposto al modello tramite il data-base è superiore alle 12 ore. N (no) nel caso contrario. Il numero dei punti che dovranno comporre il grafico è già sul file proveniente dal

modello; tra le quantità che possono essere stampate vi è anche la temperatura.

```
PROGRAM AQPLOTS
PARAMETER(KF=5,KZ=5,KT=100,KD=3)
DIMENSION D(KF,KT,KD),Z(KZ,KT,KD),T(KT),TE(KT,KD)
DIMENSION AM(KT,KD),AZ(KT,KD),PO(KT,KD),SI(KT,KD),O(KT,KD)
DIMENSION P(KT,5)
COMMON NF,NZ,ND,NT,DETI,ATA
CHARACTER LE(4)*3,DS*8,US*1,DD*1
2 WRITE(*,*)' PROGRAM AQPLOT'
WRITE(*,*)' TYPE DATA INPUT FILENAME-MAX 8 CHARS'
WRITE(*,*)' TO STOP,TYPE:NO'
READ(*,'(A8)')DS
IF(DS.EQ.'NO')GOTO 3
OPEN(1,FILE=DS)
REWIND 1
READ(1,*)NF,NZ,ND,ANT,DETI,ATA
NT=ANT+.1
WRITE(*,*)' DOUBLE-TIME OUTPUT FROM MODEL-Y/N'
READ(*,'(A)')DD
WRITE(*,*)' PRINT DATA-Y/N '
READ(*,'(A)')US
DO 1 I=1,NT
DO 1 J=1,ND
READ(1,*)(D(K,I,J),K=1,NF),(Z(K,I,J),K=1,NZ),AM(I,J),AZ(I,J)
*,PO(I,J),SI(I,J),O(I,J),TE(I,J)
1 CONTINUE
CLOSE(1)
DO 4 I=1,NT
IF(DD.EQ.'N')GOTO 44
AVAN=MOD(I,2)
IF(AVAN.EQ.0.)T(I)=INT(I/2)*DETI+.5
IF(AVAN.NE.0.)T(I)=T(I-1)+.5
GOTO 4
44 T(I)=I*DETI
4 CONTINUE
CALL PLOT(KF,KT,KZ,KD,D,Z,T,TE,AM,AZ,PO,SI,O,US,VS,LE,DS,P)
GOTO 2
3 STOP
END
```

```

SUBROUTINE PLOT(KF,KT,KZ,KD,D,Z,T,TE,AM,AZ,PO,SI,O,US,VS,LE,
1DS,P)
DIMENSION D(KF,KT,KD),Z(KZ,KT,KD),T(KT),TE(KT,KD)
DIMENSION AM(KT,KD),AZ(KT,KD),PO(KT,KD),SI(KT,KD),O(KT,KD)
DIMENSION P(KT,5),IDX(5)
COMMON NF,NZ,ND,NT,DETI,ATA
CHARACTER LE(5)*3,DS*8,US*1,VS*2,LE2(4)*3
CHARACTER DISE(21,71)*1
REAL MX,NX,MY,NY
1 WRITE(*,*)' PLOT OF CURVES AT DEPTH N.(TO STOP TYPE O)'
READ(*,*)KY
IF(KY,EQ,0)GOTO 2
WRITE(*,*)' PLOT OF PHYTO+ZOO OR NUTRIENTS-PZ/NU'
READ(*,'(A)')VS
WRITE(*,*)' HOW MANY CURVES WANTED-MAX.5'
READ(*,*)NCU
IF(VS,EQ,'NU')GOTO 4
WRITE(*,*)' PHYTO OR ZOO-P/Z '
READ(*,'(A)')VS
IF(VS,EQ,'Z')GOTO 3
DO 5 IK=1,NCU
WRITE(*,*)' TYPE PHYTO COMPONENT N.',IK
5 READ(*,*)IDX(IK)
IF(US,EQ,'Y')WRITE(*,6)
6 FORMAT(' DAYS',2X,'PHYTO SP. COMPONENTS, FROM LEFT TO RIGHT')
WRITE(*,7)(IDX(IK),IK=1,NCU)
7 FORMAT(' PHYTO SP.:',5I4)
WRITE(*,*)' TEMP. AT DEPTH N.',KY
WRITE(*,*)
DO 8 IK=1,NCU
KI=IDX(IK)
DO 8 I=1,NT
8 P(I,IK)=D(KI,I,KY)
GOTO 9
9 DO 10 IK=1,NCU
WRITE(*,*)' TYPE ZOO COMPONENT N.',IK
10 READ(*,*)IDX(IK)
IF(US,EQ,'Y')WRITE(*,11)
11 FORMAT(' DAYS',2X,'ZOO SP. COMPONENTS, FROM LEFT TO RIGHT')
12 FORMAT(' ZOO SP.:',5I4)
WRITE(*,12)(IDX(IK),IK=1,NCU)
WRITE(*,*)' TEMP. AT DEPTH N.',KY
WRITE(*,*)
DO 13 IK=1,NCU
KI=IDX(IK)
DO 13 I=1,NT
13 P(I,IK)=Z(KI,I,KY)
9 MX=T(1)
NX=MX
MY=P(1,1)
NY=MY

```

```

DO 14 I=1,NT
IF(US.EQ.'N')GOTO 15
GIO=T(I)+ATA-1,
WRITE(*,16)GIO,TE(I,KY),(P(I,IK),IK=1,NCU)
16  FORMAT(1X,2F6.1,5(2X,G11.4))
15  IF(T(I),GT,MX)MX=T(I)
    IF(T(I),LT,NX)NX=T(I)
    DO 17 J=1,NCU
      IF(P(I,J),GT,MY)MY=P(I,J)
      IF(P(I,J),LT,NY)NY=P(I,J)
17  CONTINUE
14  CONTINUE
    GOTO 18
4   LE(1)='NH4'
    LE(2)='NO3'
    LE(3)='PO4'
    LE(4)='SI'
    LE(5)='O2'
    DO 19 IK=1,NCU
      WRITE(*,*)' TYPE NUTRIENT COMPONENT N.'*IK
19  READ(*,*)IDX(IK)
20  FORMAT(' DAYS',3X,'NUTRIENTS FROM LEFT TO RIGHT')
    IF(US.EQ.'Y')WRITE(*,20)
    DO 21 IK=1,NCU
      KI=IDX(IK)
      LE2(IK)=' '
21  LE2(IK)=LE(KI)
      WRITE(*,*)' (MMOL/M3) AT DEPTH N.'*KY
      WRITE(*,22)(LE2(IK),IK=1,NCU)
22  FORMAT(5(8X,A3))
    DO 23 IK=1,NCU
      K=IDX(IK)
      GOTO(24,25,26,27,46)*K
      GOTO 32
24  DO 28 I=1,NT
28  P(I,IK)=AM(I,KY)
      GOTO 23
25  DO 29 I=1,NT
29  P(I,IK)=AZ(I,KY)
      GOTO 23
26  DO 30 I=1,NT
30  P(I,IK)=PO(I,KY)
      GOTO 23
27  DO 31 I=1,NT
31  P(I,IK)=SI(I,KY)
      GOTO 23
46  DO 47 I=1,NT
47  P(I,IK)=O(I,KY)
23  CONTINUE
32  MX=T(1)
    NX=MX

```

```

MY=P(1,1)
NY=MY
DO 33 I=1,NT
IF(US.EQ.'N')GOTO 34
GIO=T(I)+ATA-1.
WRITE(*,35)GIO,(P(I,IK),IK=1,NCU)
35 FORMAT(1X,F6.1,5(2X,G11.4))
34 IF(T(I).GT.MX)MX=T(I)
IF(T(I).LT.NX)NX=T(I)
DO 36 J=1,NCU
IF(P(I,J).GT.MY)MY=P(I,J)
IF(P(I,J).LT.NY)NY=P(I,J)
36 CONTINUE
33 CONTINUE
18 DO 37 I=1,21
DO 37 J=1,71
37 DISE(I,J)=' '
DO 38 I=1,21
DISE(I,1)=' '
38 DISE(I,71)=' '
DO 39 I=1,71
DISE(1,I)='- '
39 DISE(21,I)='- '
IF(MX.LT.0)MX=0.
IF(NX.GT.0)NX=0.
IF(MY.LT.0)MY=0.
IF(NY.GT.0)NY=0.
IF((MX-NX).EQ.0)MX=NX+1.E-10
IF((MY-NY).EQ.0)MY=NY+1.E-10
FX=70/(MX-NX)
FY=20/(MY-NY)
IY0=21+NY*FY
IX0=-NX*FX+2
DO 40 I=1,21
40 DISE(I,IX0)=' ,'
DO 41 I=1,71
41 DISE(IY0,I)='- '
DO 42 J=1,NCU
DO 42 I=1,NT
IXP=(T(I)-NX)*FX+1
IYP=21-(P(I,J)-NY)*FY+.5
42 DISE(IYP,IXP)=CHAR(16+IDX(J))
WRITE(*,*)' DEPTH N,',KY
WRITE(*,*)
DO 43 I=1,21
43 WRITE(*,44)(DISE(I,J);J=1,71)
44 FORMAT(1X,71A1)
45 GOTO 1
2 RETURN
END

```

INDICE

INTRODUZIONE	2
IL MODELLO	4
LA STRUTTURA DEL PROGRAMMA AQMODL5	5
Main Program	6
SUBROUTINE INIT	10
" MESRES	16
" INPRAT	19
" PHYTO	21
" UPRES	23
" ZOO	32
" OUT	34
FUNZIONE DAYL	36
" RADMAX	38
" CINE	38
" MISCELLANEA	38
ESEMPIO ESPLICATIVO	39
SUMMARY	45
RIFERIMENTI BIBLIOGRAFICI	46
APPENDICE (Listati dei Programmi FORTRAN)	47

