ASSOCIATION POUR LE DÉVELOPPEMENT DE L'ENSEIGNEMENT
ET DE LA RECHERCHE EN SYSTÉMATIQUE APPLIQUÉE

MELISSA

Memorandum of Understanding
ECT/FG/CB/95.205

CONTRACT ESA-ESTEC/ADERSA
Purchase Order n° 151 491 of 10/05/95

TECHNICAL NOTE : 28.4

Software of computation of the biomass composition in the photoautotrophic compartment

Version : 1
Issue : 0

LECLERCQ J.-J.

July 1997

10, rue de la Croix Martre
91873 PALAISEAU Cedex
Téléphone : (33) 01 60 13 53 53
Télécopie : (33) 01 69 20 05 63
E-Mail : adersa@adersa.worldnet.net

adersa

Document Change Log

Version	Issue	Date	Observation
0	0	April 1997	Draft
1	0	July 1997	Original version

TABLE OF CONTENTS

1. INTRODUCTION

The aim of this technical note is to furnish to the Universitat Autonoma de Barcelona a software giving a predicted estimation of the composition of the biomass in the photoautotrophic compartment. This software is based on the first principles model built by J.F. Cornet (LGCB Clermont-Ferrand) for simulation of cultures of the cyanobacterium Spirulina Platensis cultivated in cylindrical and radially illuminated photobioreactors.

This model (TN19.2 versions at July 1993 and January 1997) takes into account limitations by light and minerals (nitrate, sulfate) and is defined by the equations giving the mean mass volumetric rate $<r_i>$ of the 9 considered compounds i (recalled in table 1). As the reactor is assumed to be perfectly mixed, the concentration of a compound i is uniform in all the volume of the reactor and the system of the mass conservation equations is :

$$\frac{dC_i^r}{dt} = < r_i > + D\,(C_i - C_i^r) \qquad 1 \le i \le 9 \qquad (1)$$

whose variables have the following meaning, for each compound i :

C_i^r : concentration in the reactor

$<r_i>$: mean mass volumetric rate

D : dilution rate (ratio of the volumetric flow rate on the total volume of the reactor)

C_i : concentration in the incoming flow of the reactor

The system (1) is quite general and can describes :

- batch cultures, when D = 0 ,
- steady state continuous cultures, when $\frac{dC_i^r}{dt} = 0$ for each i ;
- dynamic continuous cultures, otherwise.

The present software is written in C language and is composed of 2 parts, a main programme, called *qbs.c*, and a main function, called *modspiru.c*. The main function, *modspiru.c*, is to be implemented in the General Purpose Station (GPS) at the highest level for a long term prediction. The purpose of *qbs.c* is only to allow tests of *modspiru.c*. An example of *qbs.c* which goes with this TN allows to test *modspiru.c*. in the operating conditions described in the TN 19.3 Appendix 7 Figure 8.

2. DESCRIPTION OF THE MAIN FUNCTION

A general description of the main fuction *modspiru.c* is given in the figure 1.

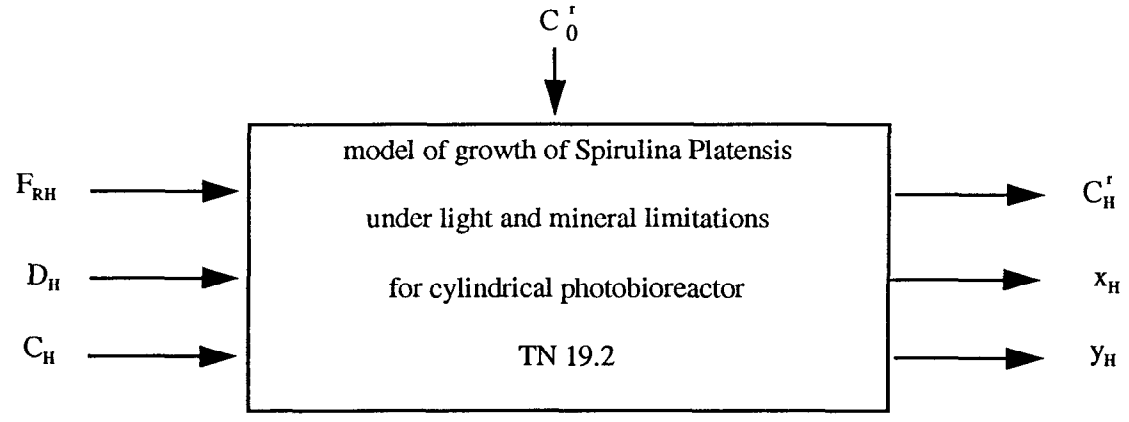


Figure 1

The variable C_0^r represents the initial conditions of the simulation (i.e. the concentration of each compound i in the reactor at the initial time t_0).

The other variables, F_{RH}, D_H, C_H, C_H^r , x_H and y_H represent the variations of the inputs and outputs of the model from the initial time t_0 to the final time t_H. If t_0 is the present moment, so t_H is a moment in the future and then the simulation is presumed to predict the future values of C_H^r, x_H and y_H according to the assumed future values of F_{RH}, D_H and C_H.

The meanings of the variables are :

C_0^r : initial concentrations in the reactor of the 9 considered compounds (kg/m^3)

F_{RH} : incident radiant energy flux (W/m^2)

D_H : dilution rate (1/h)

C_H : concentrations in the incoming flow of the 9 compounds (kg/m^3)

C_H^r : concentrations in the reactor of the 9 compounds (kg/m^3)

x_H : composition of the total biomass expressed in mass fraction (dimensionless)

y_H : global formula of the total biomass expressed with the chemical elements C, H, O, N, S and P.

The components of the vector of concentrations C_0^r are given in Table 1. At a given moment, the variables C_H and C_H^r are vectors of concentrations and the meaning of their components are, of course, the same as C_0^r.

Compound			i
XT	:	total biomass	1
XA	:	active biomass	2
CH	:	chlorophyll a	3
PC	:	phycocyanin	4
P	:	protein	5
N	:	nitrate	6
S	:	sulfate	7
XV	:	vegetative biomass	8
EPS	:	exopolysaccharide	9

Table 1 : storage of the 9 compounds in the variables C_0^r, C_H and C_H^r

The composition of the total biomass (expressed in mass fraction) is given according to the table of TN 19.2 p11 and recalled hereafter :

<table>
<tr><td colspan="4">Active Biomass (XA)</td><td rowspan="3">Glycogen

(G)</td><td rowspan="3">Exopoly-saccharide

(EPS)</td></tr>
<tr><td colspan="2">Proteins (P)</td><td rowspan="2">Chlorophylls

(CH)</td><td rowspan="2">Biomass

(B)</td></tr>
<tr><td>Phyco - cyanins (PC)</td><td>Other Proteins (OP)</td></tr>
<tr><td colspan="5">Vegetative biomass (XV)</td></tr>
<tr><td colspan="6">Total biomass (XT)</td></tr>
</table>

The 6 components $x_H(j)$ of x_H are the mass fractions (dimensionless) of the 6 constituents j of the total biomass (Table 2)

compound			j
PC	:	phycocyanins	1
OP	:	other proteins	2
CH	:	chlorophylls	3
B	:	biomass	4
G	:	glycogen	5
EPS	:	exopolysaccharide	6

Table 2 : storage of the constituents of the total biomass in x_H.

The 5 components y_H (k) of y_H are respectively the coefficients a, b, c, d and e of the global formula $CH_a O_b N_c S_d P_e$ of the total biomass.

The computation of y_H is based on the formula of the 3 following compounds :

active biomass : $CH_{1.566} O_{0.405} N_{0.192} S_{0.0052} P_{0.0063}$

polysaccharide : $CH_{1.65} O_{0.95} S_{0.015}$

glycogen : $CH_{1.67} O_{0.711} S_{0.0007}$

The connection between the variables of figure 1 and the arguments of *modspiru.c* is given in the table 3.

variables of Figure 1	arguments of *modspiru.c*
C_0^r	Cr0
F_{RH}	FRH
D_H	dilH
C_H	CiH
C_H^r	CrH
x_H	fmasbioH
y_H	chonspH

Table 3 : Connection with the arguments of *modspiru.c*

3. METHODS OF INTEGRATION

3.1. Calculation of the specific growth rates

The general expressions of the specific growth rates are (notations of TN 19.2) :

$$\langle R_{XA} \rangle = C_{PC} \cdot \frac{2\mu_M'}{R^2} \int_0^R f \cdot r \cdot dr \qquad (2)$$

$$\langle R_{EPS} \rangle = C_{PC} \cdot \frac{2\mu_M'^{EPS}}{R^2} \int_0^R f_{EPS} \cdot r \cdot dr \qquad (3)$$

with $$f = \frac{4\pi J_r}{K_j + 4\pi J_r}$$

$$f_{EPS} = \frac{4\pi J_r}{K_j^{EPS} + 4\pi J_r}$$

With the change of variable defined by

$$Z = \frac{r}{R} \qquad (Z = \text{dimensionless abscissa})$$

the expressions (2) and (3) becomes

$$\langle R_{XA} \rangle = 2\mu'_M \cdot C_{PC} \int_0^1 f \cdot Z \cdot dZ \tag{4}$$

$$\langle R_{EPS} \rangle = 2\mu_M^{'EPS} \cdot C_{PC} \int_0^1 f_{EPS} \cdot Z \cdot dZ \tag{5}$$

The expression of $4\pi J_r$ is recalled hereafter (with the notations of TN 19.2) :

$$4\pi J_r = \frac{F_R}{Z} \cdot \frac{2\cosh(\delta Z)}{\cosh(\delta) + \alpha.\sinh(\delta)}$$

In the present software, the computation of $<R_{XA}>$ and $<R_{EPS}>$ is done trough the expressions (4) and (5) in which the corrective factor wiv (working illuminated volume) is added.

When $4\pi J_r \leq F_{min}$, then f and f_{EPS} are null.

The expression of $I = \int_0^1 g.\ Z.dZ$ is still equal to :

$$I = \int_0^\varepsilon g.\ Z.dZ + \int_\varepsilon^1 g.\ Z.dZ$$

where g means f or f_{EPS} depending on the expression of $<R_{XA}>$ or $<R_{EPS}>$.

When ε tends to 0,

the term $I_1 = \int_0^{\varepsilon} g.Z.dZ$ tends to $\frac{1}{2} \cdot \varepsilon^2$

and $I \cong \int_{\varepsilon}^{1} g.\ Z.dZ$

In the present software, $\varepsilon = \frac{10^{-5}}{R}$.

The integration along the interval [ε, 1] is done by the trapezium method. This interval is divided into 500 equal parts.

3.2. Calculation of the concentrations in the reactor

Several methods of integration (Euler, Runge-Kutta, Gear-Adams) were evaluated on the Simulink® programme elaborated by N. Fulget (ADERSA ; TN 21.2) from the software Photosim written by J.F.Cornet (LGCB ; TN 19.3). All these methods are equivalent for the integration of the system (1).

So the simplest method is choosen : Euler's method with an integration step of 0.5 hour.

4. EXAMPLE OF SIMULATION

This example is described in TN 19.3 ; Appendix 7 ; Figure 8.

The initial concentrations in the photobioreactor are (in kg/m^3) :

$C_{XA0} = 0.1$
$C_{EPS0} = 0.02$
$C_{N0} = 0.8$
$C_{S0} = 0.2$
$C_{XV0} = C_{XA}$
$C_{XT0} = C_{XA} + C_{EPS}$
$C_{CH0} = 0.01 * C_{XA}$
$C_{PC0} = 0.162 * C_{XA}$
$C_{Pr0} = 0.684 * C_{XA}$

The time sequences are (t means time) :

- from t = 0 to t = 50h :

  $F_{RH} = 50\ W/m^2$
  $D_H = 0$

- from t = 50h to t=250h, the reactor is feeded with a solution of nitrate and sulfate, without change of the incident radiant energy flux :

$$\begin{vmatrix} F_{RH} & = & 50\ W/m^2 \\ D_H & = & 0.05\ h^{-1} \\ C_N & = & 0.8\ kg/m^3 \\ C_s & = & 0.2\ kg/m^3 \end{vmatrix}$$

- from t = 250h to t = 500 h, the incident radiant energy flux is increased, without any other change

  $F_{RH} = 100\ W/m^2$

The results of the simulation are saved in 3 ASCII files under a matrix from. The first column of this matrix is the vector of time, each row is the vector of concentrations in the reactor or of biomass composition or of global formula at a given moment.

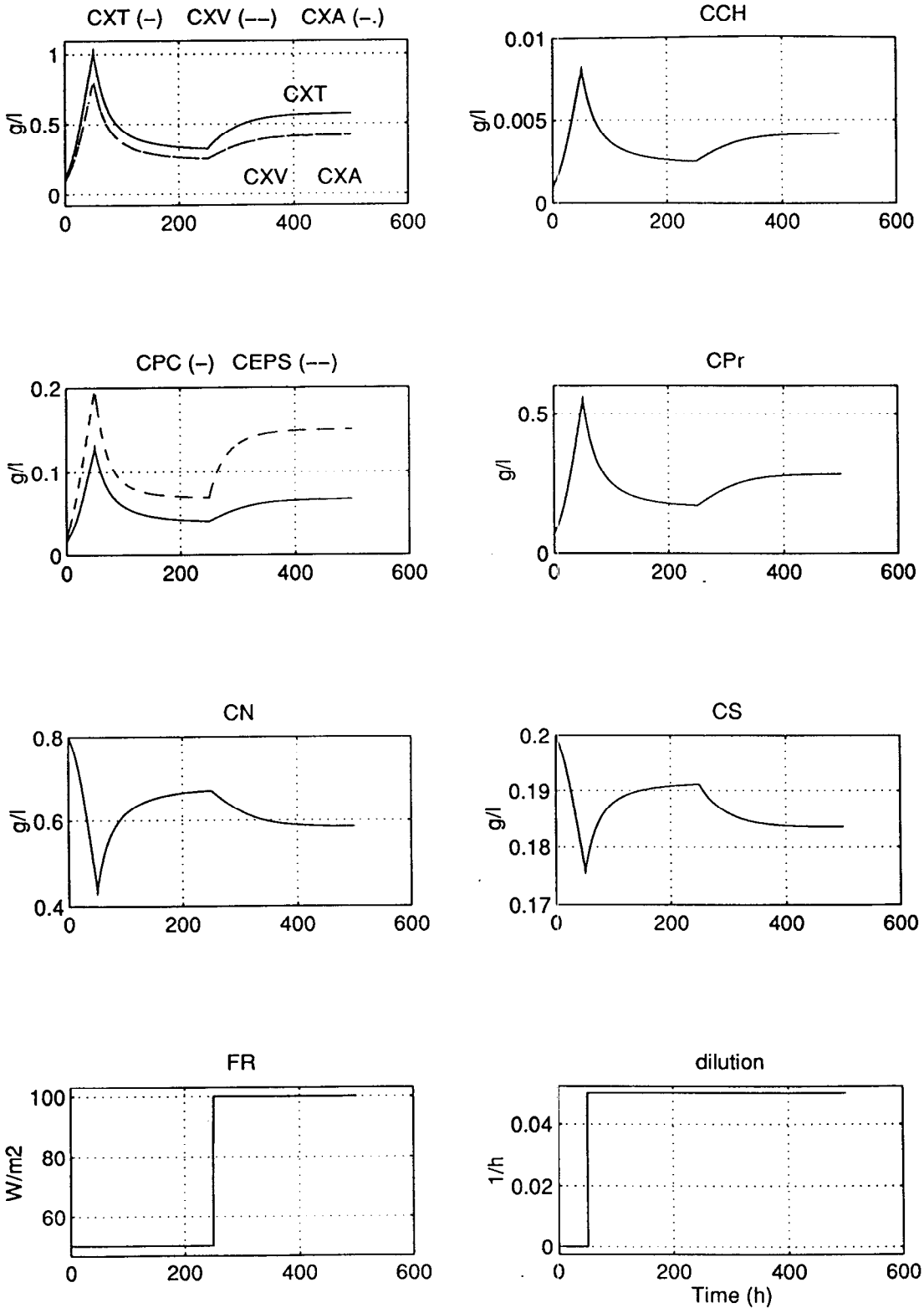
They are plotted by means of a commercial drawing software.

The figure 2 shows the concentrations of the 9 considered compounds in the reactor. They are identical to those of TN 19.3 ; Appendix 7 ; Figure 8.

The figures 3 and 4 give the time evolutions of the mass composition and global formula of the total biomass.

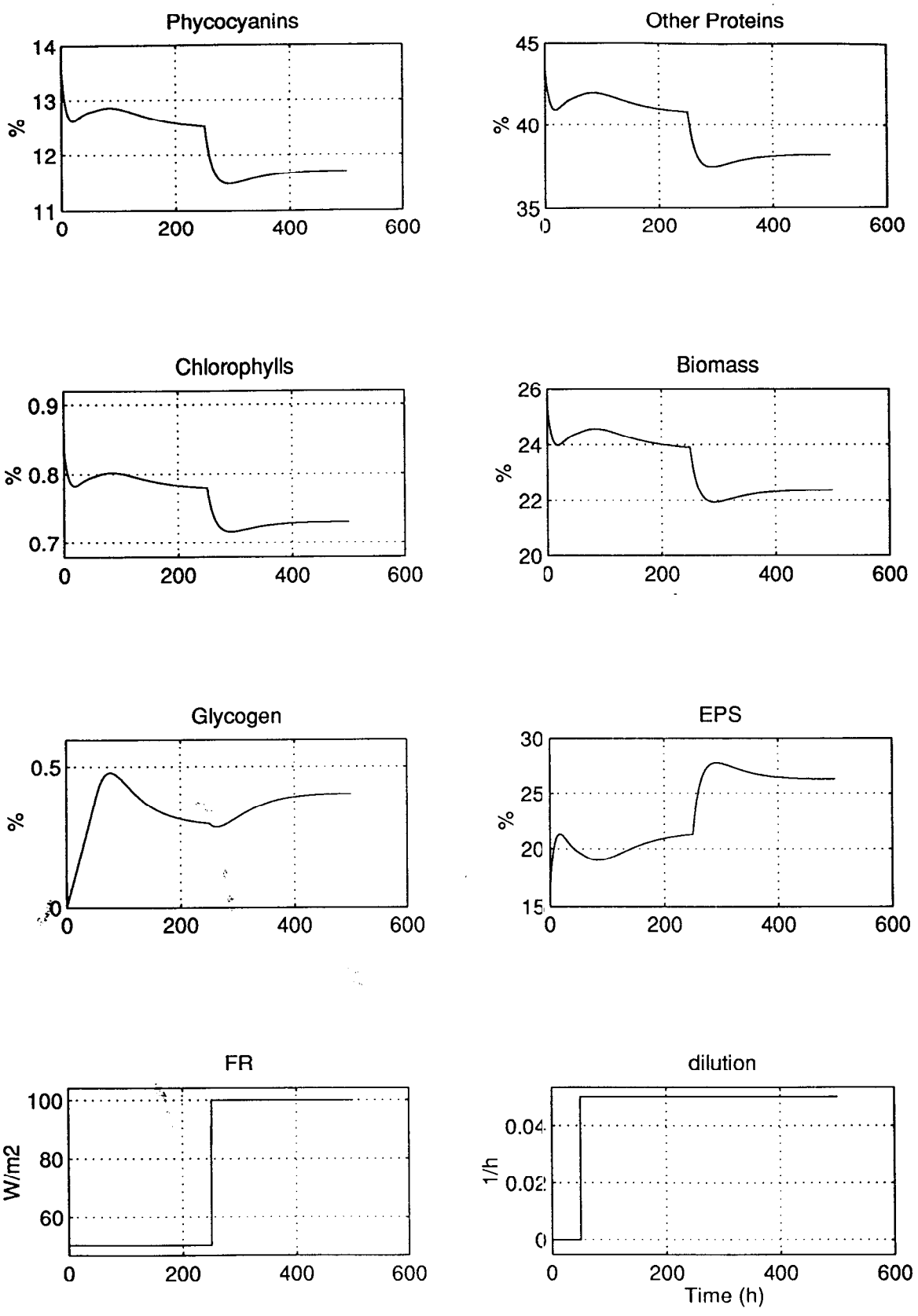
At the beginning of the simulation (during the batch culture where the dilution is null), the nitrate concentration (CN graph of figure 2) is decreasing and the nitrogen composition of the total biomass (N graph of figure 4) is decreasing too. This is not a paradox. Indeed, when the nitrogen composition of the total biomass is decreasing, the concentration of the total biomass (CXT graph of figure 2) is increasing. And all along the batch (from t = 0 to t = 50 h), the total nitrogen concentration in the reactor is constant as it is shown in figure 5 : the total nitrogen concentration is equal to 0.1924 ± 0,0001 g/l.

The same checking was done for the sulphur concentration in the reactor during the batch (fig. 5 too, where the total sulphur concentration is equal to (6.770 ± 0,001) . 10^{-2} g/l).



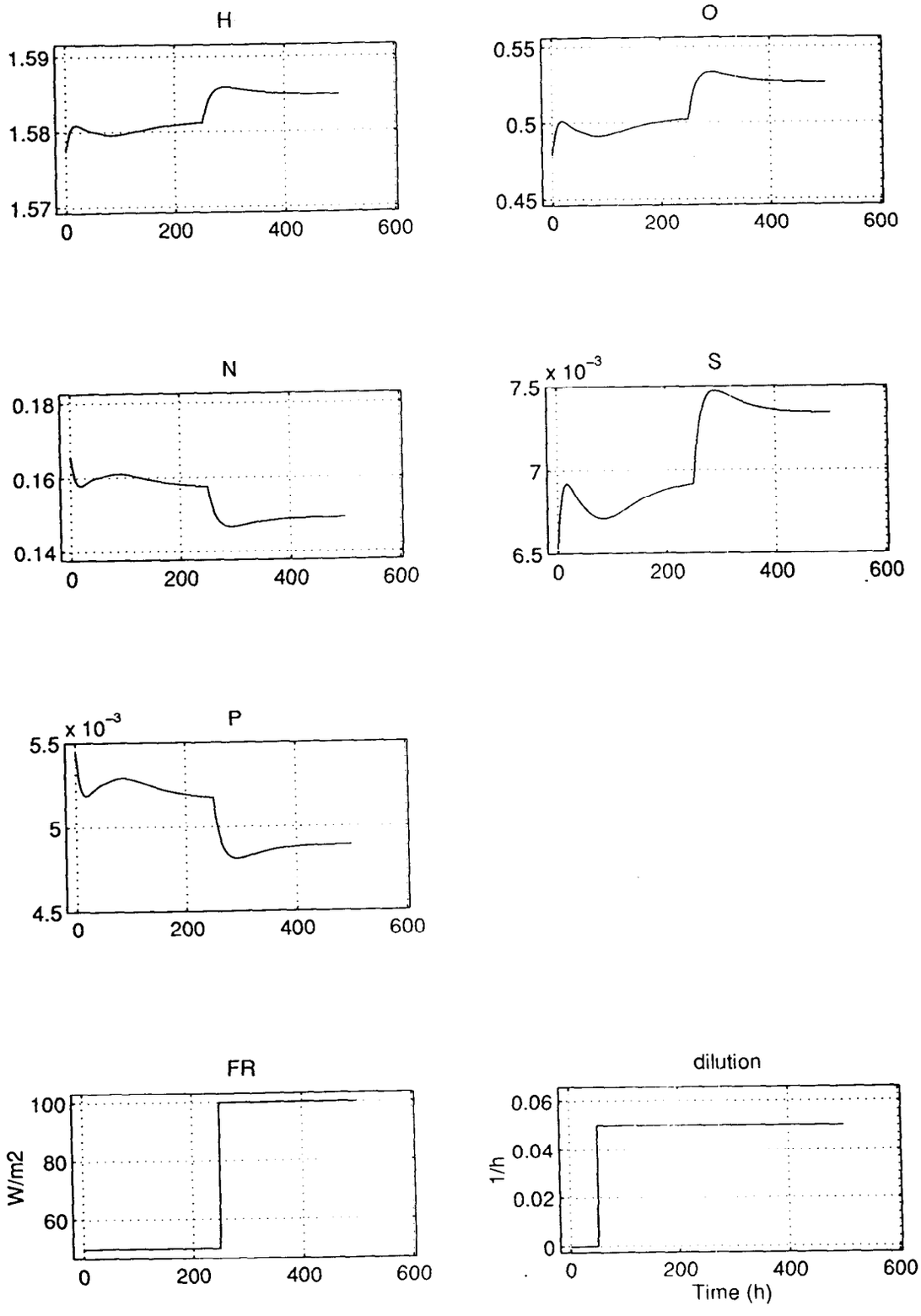
Concentrations in the reactor

Figure 2



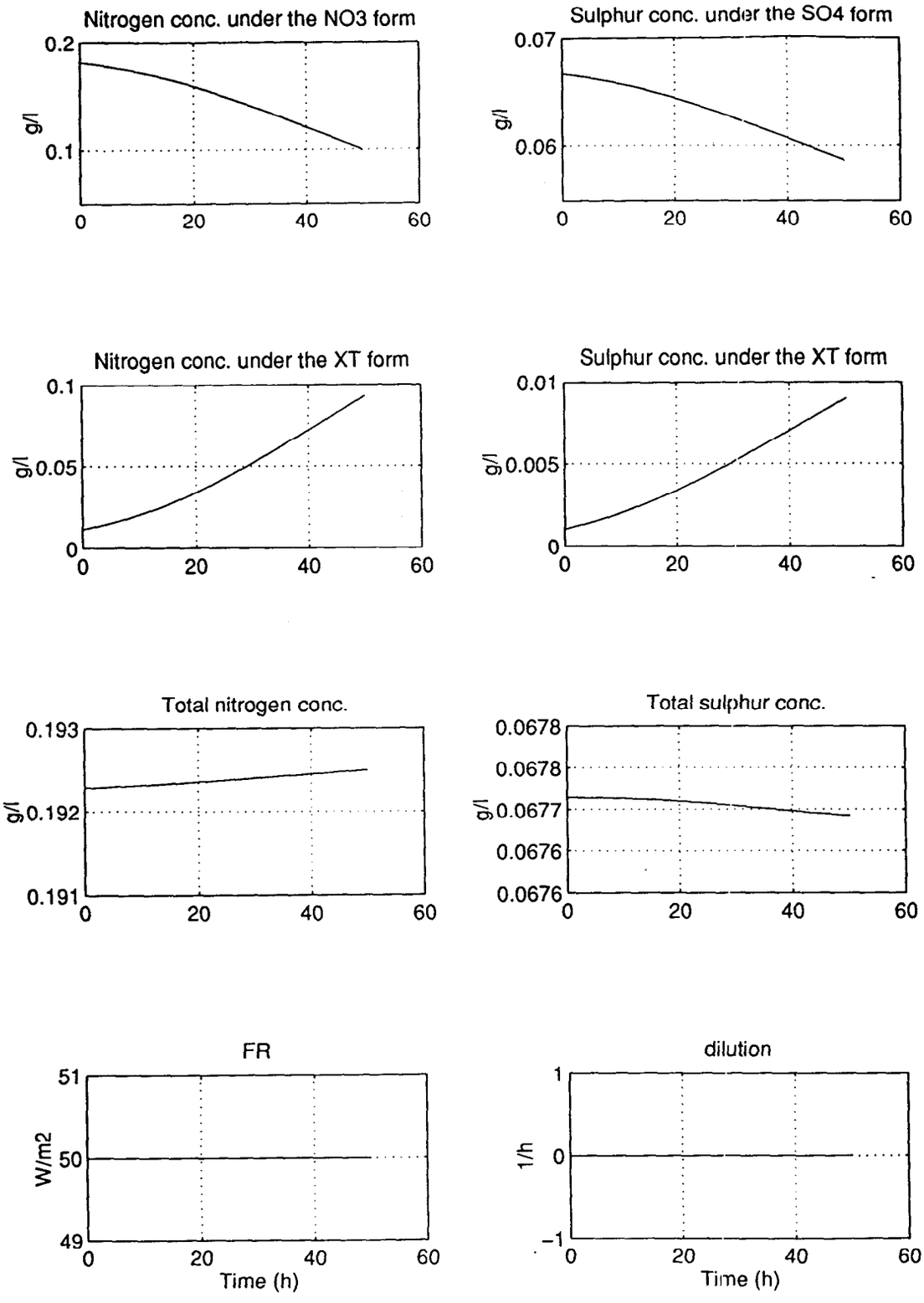
Biomass : mass composition (in %)

Figure 3



Biomass : global formula

Figure 4



Checking of the total concentrations of Nitrogen and Sulphur

Figure 5

5. CONCLUSION

The aim of the programme, described in this technical note and given in annex, is to help to predict the quality of the total biomass by mean of 2 groups of values :

1. the global chemical formula (relative molar fractions of the elements C, H, O, N, S and P)

2. the mass composition of the total biomass in the following compounds :

   - phycocyanins
   - other proteins
   - chlorophylls
   - biomass
   - glycogen
   - exopolysaccharide.

The prediction of this quality lay on the first principles model of growth of cyanobacterium Spirulina Platensis cultivated in cylindrical and radially illuminated photobioreactors (built by J. F. Cornet at LGCB).

It needs not only the measurement of the initial concentration in the reactor of each of the 9 considered compounds (total biomass, active biomass, chlorophylla, phycocyanin, protein, nitrate, sulfate, vegetative biomass and exopolysaccharide) but also the estimated concentration of each of these compounds in the incoming flow all over the prediction horizon.

These prediction may reasonably be done over a period of 1 to 3 days and the programme is supposed to be called off line once a day.

REFERENCES

CORNET J.F., DUSSAP C.G., GROS J.-B., 1993a. Modelling of physical limitations in photobioreactors. Adaptation of the light energy transfer model to cylindrical geometries. ESA contract PRF 130-820, Technical Note 19.1.

CORNET J.F., DUSSAP C.G., GROS J.-B., 1993b. Modelling of physical limitations in photobioreactors. Modelling of exopolysaccharide synthesis in cultures of *Spirulina platensis*. ESA contract PRF 130-820, Technical Note 19.2.

CORNET J.F., DUSSAP C.G., GROS J.-B., 1993c. Modelling of physical limitations in photobioreactors. Applications to simulation and control of the Spirulina Compartment of the MELISSA artifical ecosystem. ESA contract PRF 130-820, Technical Note 19.3.

FULGET N., "MELISSA. First approach of model based predictive control of Spirulina compartment". Contract ESA-ESTEC/ADERSA PRF n°132443. December 1994. TN 21.2.

ANNEX

This annex contains the following files :

::: • qbs.c	programme for testing modspiru.c
: • modspiru.c	main fonction for the computation of the composition of the total biomass
: • pargene.h	include of general parameters (dimensions of variables and parameters of the integration methods).
: • parmodel.h	include of the parameters of the model. The values of these parameters are those of TN 19.2 (versions of July 1993 and January 1997). Note that the working illuminated volume wiv is fixed to 1 as in TN 19.3. These values have to be updated, if necessary.

```
        /*
        qbs.c

        Main programme for the estimation of the quality of the biomass of
        the photoautotrophic (spirulina) compartment

        J.J. Leclercq
        ADERSA
        April 1997

        */

#include "parmodel.h"
#include "pargene.h"
#include "math.h"
#include <stdio.h>
extern void modspiru();

main()
{
        double FRH[nT+1], dilH[nT+1], CiH[nT+1][nsig+1], CrH[nT+2][nsig+1];
        double fmasbioH[nT+1][ncomp+1], chonspH[nT+1][ncoef+1], tH[nT+1];
        double Cr0[nsig+1];
        double H, tsim;
        short i, i1, i2, choisim;
        FILE *pf;

/*      Choice of the example of simulation */
        choisim = 1;

/*
In this part of the main programme, the user has to define
        1_ the duration of the simulation : tsim (expressed in hours)
        2_ the initial concentrations in the reactor (in kg/m3)
        3_ the time variations of the inputs along the horizon of simulation H :
                FRH : incident radient energy flux (W/m2)
                dilH : dilution rate (1/h)
                concentration (kg/m3) in the incoming flow of :
                CiH(1) : total biomass
                CiH(2) : active biomass
                CiH(3) : chorophyll
                CiH(4) : phycocyanins
                CiH(5) : proteins
                CiH(6) : nitrate
                CiH(7) : sulfate
                CiH(8) : vegetative biomass
                CiH(9) : exopolysaccharide

*/
   if (choisim == 1)
   {
        /* ======================================================
           Simulation according to TN19.3 , Appendix 7 , Figure 8
           ====================================================== */

        /* 1_ initialization of the duration of the simulation */
        tsim = 500.;            /* duration of the simulation (in hours) */
        H = tsim / dt;          /* length of simulation (in sampling periods)
                                   dt : sampling period defined in pargene.h */
        if(nT < H)      /* dimension test */
          {
          printf("******* The simulation duration is too long\n");
          printf("versus the dimension nT. Increase nT in pargene.h *******\n");
          exit(1);
```

```
        }

        /* 2_ initial concentrations in the reactor */
        Cr0[2] = .1;                    /* active biomass */
        Cr0[9] = .02;                   /* exopolysaccharide */
        Cr0[6] = .8;                    /* nitrate */
        Cr0[7] = .2;                    /* sulfate */
        Cr0[8] = Cr0[2];                /* vegetative biomass */
        Cr0[1] = Cr0[2] + Cr0[9];       /* total biomass */
        Cr0[3] = .01 * Cr0[2];          /* chlorophyll */
        Cr0[4] = .162 * Cr0[2];         /* phycocyanin */
        Cr0[5] = .684 * Cr0[2];         /* protein */

        /* 3_ time variation of the inputs */
        i1 =(int)(50. / dt);    /* dilution rate step at time = 50 h */
        i2 = (int)(250. / dt);  /* incident flux step at time = 250 h */
        for (i=0; i<=i2; i++)
                FRH[i] = 50.;
        for (i=i2+1; i<=H; i++)
                FRH[i] = 100.;          /* incident flux step */
        for (i=0; i<=i1; i++)
                dilH[i] = 0.;
        for (i=i1+1; i<=H; i++)
                dilH[i] = 0.05;         /* dilution rate step */
        for (i=0; i<=H; i++)
        {
                /* constant concentration (kg/m3) in the incoming flow of : */
                CiH[i][1] = .0;         /* total biomass */
                CiH[i][2] = .0;         /* active biomass */
                CiH[i][3] = .0;         /* chorophyll */
                CiH[i][4] = .0;         /* phycocyanins */
                CiH[i][5] = .0;         /* proteins */
                CiH[i][6] = Cr0[6];     /* nitrate */
                CiH[i][7] = Cr0[7];     /* sulfate */
                CiH[i][8] = .0;         /* vegetative biomass */
                CiH[i][9] = .0;         /* exopolysaccharide */
        }
}
else if (choisim == 2)
{
        /* ======================================================
           Simulation of a batch (dilution rate is null)
           ====================================================== */

        /* 1_ initialization of the duration of the simulation */
        tsim = 50.;             /* duration of the simulation (in hours) */
        H = tsim / dt;          /* length of simulation (in sampling periods)
                                   dt : sampling period defined in pargene.h */
        if(nT < H)      /* dimension test */
          {
          printf("******* The simulation duration is too long\n");
          printf("versus the dimension nT. Increase nT in pargene.h *******\n");
          exit(1);
          }

        /* 2_ initial concentrations in the reactor */
        Cr0[2] = .1;                    /* active biomass */
        Cr0[9] = .02;                   /* exopolysaccharide */
        Cr0[6] = .8;                    /* nitrate */
        Cr0[7] = .2;                    /* sulfate */
        Cr0[8] = Cr0[2];                /* vegetative biomass */
        Cr0[1] = Cr0[2] + Cr0[9];       /* total biomass */
        Cr0[3] = .01 * Cr0[2];          /* chlorophyll */
        Cr0[4] = .162 * Cr0[2];         /* phycocyanin */
```

```
        Cr0[5] = .684 * Cr0[2];         /* protein */

        /* 3_ time variation of the inputs */
        for (i=0; i<=H; i++)
                FRH[i] = 50.;
        for (i=0; i<=H; i++)
                dilH[i] = 0.;
        for (i=0; i<=H; i++)
        {
                /* constant concentration (kg/m3) in the incoming flow of : */
                CiH[i][1] = .0;         /* total biomass */
                CiH[i][2] = .0;         /* active biomass */
                CiH[i][3] = .0;         /* chorophyll */
                CiH[i][4] = .0;         /* phycocyanins */
                CiH[i][5] = .0;         /* proteins */
                CiH[i][6] = Cr0[6];     /* nitrate */
                CiH[i][7] = Cr0[7];     /* sulfate */
                CiH[i][8] = .0;         /* vegetative biomass */
                CiH[i][9] = .0;         /* exopolysaccharide */
        }
    }

/*
End of the initialization done by the user
*/

        if(nZ < nstep)  /* dimension test */
          {
          printf("******* The number of integration steps is too big\n");
          printf("versus the dimension nZ. Increase nZ in pargene.h *******\n");
          exit(1);
          }

        /*
        Computation on the time horizon H of :
        . concentrations of the compounds in the reactor : 'CrH'
        . mass fraction of the biomass : fmasbioH
        . global formula (CHONSP) of the biomass : chonspH
        */
        modspiru(H, Cr0, FRH, dilH, CiH, CrH, fmasbioH, chonspH);

        /*
        Saving results into 3 files *.res :
        1_ concentrations of the compounds in the reactor : conc.res
        2_ mass fraction of the biomass : compo.res
        3_ global formula (CHONSP) of the biomass : glob.res
        */
        for (i=0; i<=H; i++)
                tH[i] = i * dt; /* vector of time */
        pf = fopen("conc.res", "w");
        for (i=0; i<=H; i++)
        {
        fprintf(pf,"%10.2f %12.5e %12.5e %12.5e %12.5e %12.5e %12.5e %12.5e %12.5e %12.5e
%12.5e %12.5e\n",tH[i],FRH[i],dilH[i],CrH[i][1],CrH[i][2],CrH[i][3],CrH[i][4],CrH[i][5],C
H[i][6],CrH[i][7],CrH[i][8],CrH[i][9]);
        }
        fclose(pf);

        pf = fopen("compo.res", "w");
        for (i=0; i<=H; i++)
        {
        fprintf(pf,"%10.2f %12.5e %12.5e %12.5e %12.5e %12.5e %12.5e\n",tH[i],fmasbioH[i]
1],fmasbioH[i][2],fmasbioH[i][3],fmasbioH[i][4],fmasbioH[i][5],fmasbioH[i][6]);
        }
```

```
        fclose(pf);

        pf = fopen("glob.res", "w");
        for (i=0; i<=H; i++)
        {
        fprintf(pf,"%10.2f %12.5e %12.5e %12.5e %12.5e %12.5e\n",tH[i],chonspH[i][1],chor
pH[i][2],chonspH[i][3],chonspH[i][4],chonspH[i][5]);
        }
        fclose(pf);

}
```

```
/*
modspiru.c

Subroutine for the estimation of the quality of the biomass
of the photoautotroph (spirulina) compartment
based on the first principles model built by LGCB (TN 19.1 and 19.2)

J.J. Leclercq
ADERSA
April 1997

-------------------------------------------------------------------------

Synopsis:
=========
modspiru(H, Cr0, FRH, dilH, CiH, CrH, fmasbioH, chonspH)
Input arguments :
-----------------
H : length of simulation (expressed in number of sampling period dt)
Cr0 : initial concentrations in the reactor (kg/m3)
FRH : time variation of the incident radiant energy flux (W/m2)
dilH : time variation of the dilution (1/h)
CiH : time variation of the concentrations in the incoming flow (kg/m3)
Ouput arguments :
-----------------
CrH : time variation of the concentrations in the reactor (kg/m3)
fmasbioH : time var. of the mass fraction of the biomass (dimensionless)
chonspH : time var. of the global formula of the biomass (dimensionless)

Storage in Cr0, CiH and CrH (at a given moment) :
        (1) : total biomass
        (2) : active biomass
        (3) : chorophyll
        (4) : phycocyanins
        (5) : proteins
        (6) : nitrate
        (7) : sulfate
        (8) : vegetative biomass
        (9) : exopolysaccharide

Storage in fmasbioH (at a given moment) :
        (1) : phycocyanins
        (2) : other proteins
        (3) : chorophylls
        (4) : biomass
        (5) : glycogen
        (6) : exopolysaccharide

Storage in chonspH (at a given moment)
of the global formula C Ha Ob Nc Sd Pe :
        chonspH(1) = a
        chonspH(2) = b
        chonspH(3) = c
        chonspH(4) = d
        chonspH(5) = e
*/

#include "parmodel.h"
#include "pargene.h"
#include "math.h"

extern void pro_auto();
extern void forglob();
extern void compo();
```

```
void  modspiru(H, Cr0, FRH, dilH, CiH, CrH, fmasbioH, chonspH)
double H;
double Cr0[nsig+1], FRH[nT+1], dilH[nT+1], CiH[nT+1][nsig+1], CrH[nT+2][nsig+1];
double fmasbioH[nT+1][ncomp+1], chonspH[nT+1][ncoef+1];
{
        short i, j;
        double FR, dil, Ci[nsig+1], Cr[nsig+1];
        double chonsp[ncoef+1], fmasbio[ncomp+1];

        for (j=1; j<=nsig; j++)
                CrH[0][j] = Cr0[j];

        for (i=0; i<=H; i++)
        {
                FR = FRH[i];
                dil = dilH[i];
                for (j=1; j<=nsig; j++)
                {
                        Ci[j] = CiH[i][j];
                        Cr[j] = CrH[i][j];
                }

                /* global formula of the biomass (CHONSP) */
                forglob(Cr, chonsp);
                for (j=1; j<=ncoef; j++)
                        chonspH[i][j] = chonsp[j];

                /* mass fraction of the biomass (dimensionless) */
                compo(Cr, fmasbio);
                for (j=1; j<=ncomp; j++)
                        fmasbioH[i][j] = fmasbio[j];

                /* concentration (in kg/m3) of each compound in the reactor */
                pro_auto(FR, dil, Ci, Cr);
                for (j=1; j<=nsig; j++)
                        CrH[i+1][j] = Cr[j];

        }
}

void  compo(Cr, fmasbio)
double Cr[nsig+1], fmasbio[ncomp+1];
{
        fmasbio[1] = Cr[4] / Cr[1];
        fmasbio[2] = (Cr[5] - Cr[4]) / Cr[1];
        fmasbio[3] = Cr[3] / Cr[1];
        fmasbio[4] = (Cr[2] - Cr[3] - Cr[5]) / Cr[1];
        fmasbio[5] = (Cr[8] - Cr[2]) / Cr[1];
        fmasbio[6] = Cr[9] / Cr[1];
}

void  forglob(Cr, chonsp)
double Cr[nsig+1], chonsp[ncoef+1];
{
        double abm, glym, epsm, abn, glyn, epsn, abf, glyf, epsf, ntot;

        /* mass fraction of active biomass */
        abm = Cr[2] / Cr[1];
        /* mass fraction of glycogen */
        glym = (Cr[8] - Cr[2]) / Cr[1];
        /* mass fraction of exopolysaccharide */
        epsm = Cr[9] / Cr[1];

        /* molar ratio of active biomass */
```

```
        abn = abm / Mab;
        /* molar ratio of glycogen */
        glyn = glym / Mgly;
        /* molar ratio of exopolysaccharide */
        epsn = epsm / Meps;

        ntot = abn + glyn + epsn;
        /* molar fraction of active biomass */
        abf = abn / ntot;
        /* molar fraction of glycogen */
        glyf = glyn / ntot;
        /* molar fraction of exopolysaccharide */
        epsf = epsn / ntot;

        /* hydrogen coefficient */
        chonsp[1] = 1.566*abf + 1.67*glyf + 1.65*epsf;
        /* oxygen coefficient */
        chonsp[2] = .405*abf + .711*glyf + .95*epsf;
        /* nitrogen coefficient */
        chonsp[3] = .192*abf;
        /* sulphure coefficient */
        chonsp[4] = .0052*abf + .0007*glyf + .015*epsf;
        /* phophorus coefficient */
        chonsp[5] = .0063*abf;
}

extern void rx_auto();
void pro_auto(FR, dil, Ci, Cr)
double FR, dil, Ci[nsig+1], Cr[nsig+1];
{
        short i;
        double RXA, REPS;
        double xCCH, xCPC, xCN, xCS, xCXV;
        double aa, bb, cc, dd, ee;
        double ri[nsig+1], dCr[nsig+1];


        xCCH = Cr[3];
        xCPC = Cr[4];
        xCN  = Cr[6];
        xCS  = Cr[7];
        xCXV = Cr[8];

        aa = xCN / (KN + xCN);
        bb = xCS / (KS + xCS);
        cc = xCPC / (KPC + xCPC);
        dd = KN / (KN + xCN);
        ee = KS / (KS + xCS);

        /* calculation of RXA et REPS */
        rx_auto(xCPC,xCCH,xCXV,FR,&RXA,&REPS);

        /* calculation of the 9 mean volumic growth rates */
        ri[1] = RXA + REPS;                                    /* rXT  */
        ri[2] = RXA * aa * bb;                                 /* rXA  */
        ri[3] = zCH * ri[2];                                   /* rCH  */
        ri[4] = zPC * RXA * (aa*bb - (dd+ee));                 /* rPC  */
        ri[5] = zP * RXA * (aa*bb - qq*ee);                    /* rP   */
        ri[6] = -YNXA * ri[2];                                 /* rN   */
        ri[7] = -YSXA * ri[2] - YSEPS * REPS * aa * bb;        /* rS   */
        ri[8] = RXA * (aa*bb + cc*(dd+ee));                    /* rXV  */
        ri[9] = REPS * aa * bb + (ri[1] - ri[8]) * (dd + ee);  /* rEPS */

        /* calculation of derivatives and integration */
```

```
        /* integration step for Euler method  = dt */
        for (i=1; i<=nsig; i+=1)
        {
                dCr[i] = ri[i] + dil * (Ci[i] - Cr[i]);
                Cr[i] = Cr[i] + dCr[i] * dt;
        }

}

void  rx_auto(CPC,CCH,CXV,FR,RXA,REPS)
double  CPC, CCH, CXV, FR, *RXA, *REPS;
{
        /* internal variables */
        double yy[nZ+1];
        double a1, a2, alpha, delta, sXA, sEPS, pijz, z, REPS1, REPS2, A, PE;
        double z0, kstep;
        short i;

        a1 = Ea * (CPC+CCH);
        a2 = Ea * (CPC+CCH) + Es*CXV;
        alpha = sqrt(a1 / a2);
        delta = (sqrt(a1 * a2)) * RT;

        /*
        Computation of RXA et REPS1
        Integration interval : [z0, 1]
        This interval is divided into 'nstep' equal parts
        Integration : trapezium method
        */
        z0 = 1.e-5 / RT;
        kstep = (1. - z0) / nstep;
        i = 0;
        for (z=z0; z<=1.; z+=kstep)
            {
            i += 1;
            yy[i] = 2*FR/z*cosh(delta*z)/(cosh(delta)+alpha*sinh(delta));
            }
        sXA = 0.;
        sEPS = 0.;
        i = 0;
        for (z=z0; z<1.; z+=kstep)
            {
            i += 1;
            pijz = (yy[i] + yy[i+1]) / 2.;
            if (pijz>=Fmin)
               {
               sXA += z * pijz / (Kj+pijz);
               sEPS += z * pijz / (KjEPS+pijz);
               }
            }
        *RXA = 2. * mupM * CPC * sXA * wiv * kstep;

        REPS1 = 2. * mupMEPS * CPC * sEPS * wiv * kstep;
        A = 4*FR * alpha * sinh(delta) / RT / (cosh(delta) + alpha*sinh(delta));
        PE = 1.222e-5 * A + 1.267;
        REPS2 = 29.33 * (2.874*PE - 3.568) * *RXA / 23.096 / (3.33-1.92*PE);
        *REPS = (REPS1 + REPS2) / 2.;

}
```

```
/*
        pargene.h

   general parameters
   ---------------- */

# define nT 5000          /* number max of time steps (modifiable by the user) */
# define nZ 1000          /* number max of integration steps along the radius (modifiable b
 the user) */
# define nsig 9           /* number of compounds of the model (non modifiable) */
# define ncoef 5          /* number of coefficients of the global formula (non modifiable)
/
# define ncomp 6          /* number of compounds of the biomass(non modifiable) */

/* parameters of the integration methods */
#define nstep     500.    /* number of integration steps along the radius */
#define dt        0.5     /* sampling period (h) */
```

```
/*
        parmodel.h

   first principles model parameters of photoautotroph compartment
   ------------------------------------------------------------- */

#define RT          .045   /* reactor radius (m) */
#define wiv         1.     /* working illuminated volume (dimensionless) */
#define Ea          872.   /* global absorption mass coefficient (m2/kg) */
#define Es          200.   /* global scattering mass coefficient (m2/kg) */
#define mupM        .45    /* max growth rate for biomass (1/h) */
#define mupMEPS     1.852  /* max growth rate for biomass phycocyanins (1/h) */
#define Kj          20.    /* half satur. cste for energy to biomass (W/m2) */
#define KjEPS       750.   /* half satur. cste for energy to EPS (W/m2) */
#define Fmin        1.     /* min incident radiant energy flux (W/m2) */

#define KN          5.3e-3 /* half satur. cste for nitrate conc. (kg/m3) */
#define KS          2.5e-4 /* half satur. cste for sulfate conc.(kg/m3) */
#define KPC         .06    /* half satur. cste for phycocyanin conc.(kg/m3) */
#define zCH         .01    /* mass biotic fraction of CH (dimensionless) */
#define zPC         .162   /* mass biotic fraction of PC (dimensionless) */
#define zP          .684   /* mass biotic fraction of P (dimensionless) */
#define YNXA        .516   /* mass conver. yield of NO3 in XA (dimensionless) */
#define YSXA        .022   /* mass conver. yield of SO4 in XA (dimensionless) */
#define YSEPS       .049   /* mass conver. yield of SO4 in EPS (dimensionless) */
#define qq          .55    /* coefficient of proportionality (dimensionless) */

#define Mab         23.096 /* molar mass of active biomass    (g/C_mole) */
#define Mgly        25.07  /* molar mass of glycogen          (g/C_mole) */
#define Meps        29.33  /* molar mass of exopolysaccharide (g/C_mole) */
```