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

MELISSA

Memorandum of Understanding
ECT/FG/MMM/97.012
CONTRACT ESA-ESTEC/ADERSA
Purchase Order n° 161 399 of 14/06/1996

TECHNICAL NOTE : 35.3

Improvement of the Spirulina Biomass Quality software

Version : 1
Issue : 0

J.-J. LECLERCQ

August 1999

10, rue de la Croix Martre
91873 PALAISEAU Cedex
Téléphone : (33) 1 60 13 53 53
Télécopie : (33) 1 69 20 05 63
Courriel : adersa@adersa.asso.fr

adersa

Document Change Log

Version	Issue	Date	Observation
0	0	July 1999	Draft
1	0	August 1999	Original version

TABLE OF CONTENTS

1. INTRODUCTION

Since the last TN 28.4 of July 1997, relating to the same subject, an up to date is now necessary for a few reasons.

First the software of prediction of the Spirulina biomass quality has to take into account the modifications done in Version 2 of Photosim.

Then this study takes also into account the comments of the TN 37.120 of UAB.
And finally, it is the opportunity to level the transferability with a view to the new structure of the Global Purpose Station.

At the meeting ESA/LGCB/ADERSA on May 12th 1999, it was decided to give an official name to a few software's :

name	software
LSPC	Light Spirulina Production Control
LRPC	Light Rhodobacter Production Control
SBQ	Spirulina Biomass Quality
MCS	Melissa Control Software (Level 2 of the hierarchical control)

So, according to this decision, the main function realizing the prediction of the Spirulina biomass quality is called *sbq* .

2. LIST OF MODIFICATIONS

In comparison with the previous version of TN 28.4, the following modifications are done in the software :

- the illuminated surface fraction, f_I, is a parameter modifiable by the user and is an argument of the function *sbq* . It is fixed now in the main programme *sbq_main* and no more in the include *parmodel.h* ;
- the global EPS volumetric production rate REPS is calculated only by one way (the approach entitled « non structured » in TN 19.2 is preferred because it is independent of the photoreactor geometry) ;
- the half saturation constant for nitrate limitation has been re-identified from the results obtained in continuous cultures by UAB in order to properly predict the nitrate concentration in the output flow of the reactor (less than 10% of mean deviation). The new value is 5.10^{-4} kg/m^3 ;
- the real EPS volumetric production rate rEPS is calculated from a new direct equation which derives from the definitions of XT and XV, that is :
  rEPS = rXT - rXV = RXA + REPS - rXV ;

Remark : In comparison with the software of Spirulina production control, the parameter, dt, which is here an integration step, is not modifiable by the user. Therefore, its value is fixed inside the function *sbq* via the include *pargene.h* .

3. COMPARISON WITH PHOTOSIM V2.0

In order to compare this new version of *sbq* with *Photosim V2.0* , the two tests at low and high dilution rate (0.025 and 0.035 h^{-1}) of UAB (figures 6 and 7, p. 12 of TN 37.120) are done with the illuminated surface fraction, f_I, set to 0.60 .

The corresponding simulations are simulated with the main programme *sbq_main* given in annex.

The evolution of the total biomass is shown in the upper left hand side graph of the figures 1 and 2. The meaning of the variables is the same as in TN 28.4 .

The numerical values of these simulations at steady state functioning are compared with those of *Photosim V2.0* in Tables 1 and 2. As it can be noted, the results of *sbq V2.0* are quite near those of *Photosim V2.0* .

dilution rate (h^{-1})	total biomass (g/l) at the simulation start	total biomass (g/l) at the simulation end
0.025	0.93	1.44
0.035	0.58	0.93

Table 1 : Simulation with *sbq V2.0*

dilution rate (h^{-1})	total biomass (g/l) at the simulation start	total biomass (g/l) at the simulation end
0.025	0.91	1.42
0.035	0.52	0.91

Table 2 : Simulation with *Photosim V2.0*
(Data from JF Cornet of LGCB)

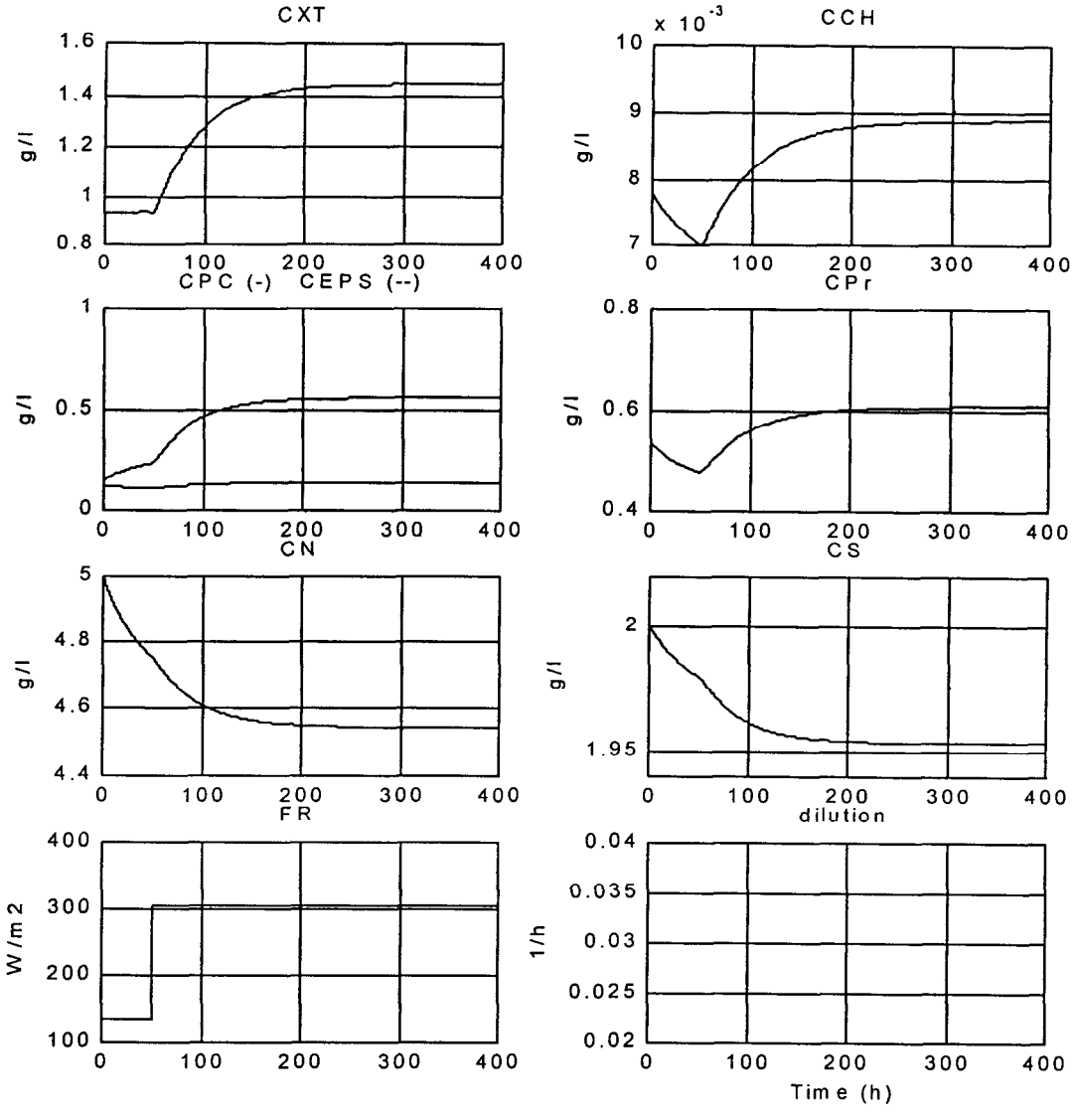

Figure 1 : Step of light flux, F_R, from 133 to 305 W/m^2 at dilution rate of 0.025 h^{-1} (simulation with *sbq V2.0*)

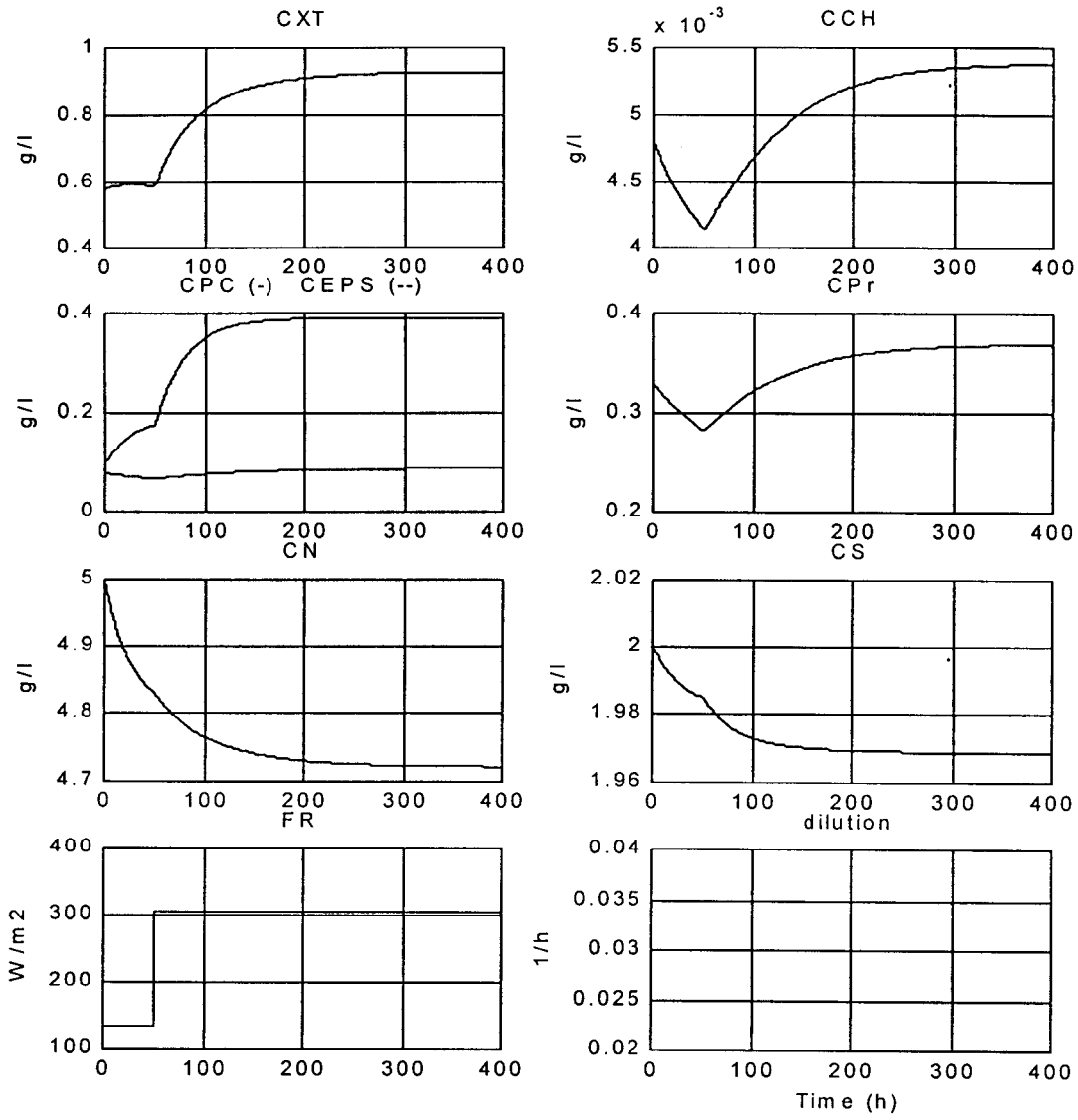

Figure 2 : Step of light flux, F_R, from 133 to 305 W/m^2 at dilution rate of 0.035 h^{-1} (simulation with *sbq V2.0*)

4. CONCLUSION

This version 2.0 of the prediction of the Spirulina Biomass Quality software is in agreement with the *Photosim V2.0* software.

At the meeting ESA/LGCB/ADERSA on May 12th 1999, it was decided to run this programme off line. So it is no more necessary to insert it in the GPS software. In the future structure of the GPS, this programme will run off line too.

This study is a first step towards a multivariable control placed at level 2 of the hierarchical control strategy defined in TN 24.1 of ADERSA.

REFERENCES

ALBIOL J., VERNEREY A., GODIA F. " Control laws of photosynthetic compartment. Integration of biomass composition prediction model ". ESTEC contract 11549/95/NL/FG, January 1998, TN 37.120 .

CORNET J.F., DUSSAP C.G., GROS J.B. " Modelling of physical limitations in photobioreactors. Applications to simulation and control of the Spirulina compartment of the MELISSA artificial ecosystem " ESA contract PRF 130820, 1993, TN 19.3 .

FULGET N. " Study for the non linear Base Model predictive Control of Spirulina compartment using knowledge model ". ESA/ESTEC contract PRF 142356, February 1995, TN 24.1.

LECLERCQ J.J. " Software of computation of the biomass composition in the photoautotrophic compartment ". ESTEC contract 151491 of 10/05/1995, July 1997, TN 28.4 .

ANNEX

This annex contains the following files :

- sbq_main.c programme for testing *sbq*
- sbq.c main function for the computation of the biomass composition
- pargene.h include of general parameters (dimension of arrays and parameters of integration)
- parmodel.h include of the parameters of the model

```
/*
sbq_main.c

Main programme for the estimation of the Spirulina Biomass Quality

J.J. Leclercq
ADERSA
July 1999
Version 2.0 (according to TN 35.3)

*/

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

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 fI, 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
      4_ the illuminated surface fraction (dimensionless)

*/
   if (choisim == 1)
   {
      /* ========================================================
         Simulation according to TN37.120   p 12 ; Figure 6
         ======================================================== */

      /* 1_ initialization of the duration of the simulation */
      tsim = 400.;        /* 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] = .78;              /* active biomass */
    Cr0[9] = .15;              /* exopolysaccharide */
    Cr0[6] = 5.;               /* 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 */
    i2 = (int)(50. / dt);      /* incident flux step at time = 50 h */
    for (i=0; i<=i2; i++)
          FRH[i] = 133.;
    for (i=i2+1; i<=H; i++)
          FRH[i] = 305.;           /* incident flux step */
    for (i=0; i<=H; i++)
          dilH[i] = 0.025;     /* dilution rate constant */
    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 */
    }

    /* 4_ illuminated surface fraction */
    fI = 0.60;
}
else if (choisim == 2)
{
    /* ========================================================
       Simulation according to TN37.120   p 12 ; Figure 7
       ======================================================== */

    /* 1_ initialization of the duration of the simulation */
    tsim = 400.;        /* 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] = .48;              /* active biomass */
    Cr0[9] = .10;              /* exopolysaccharide */
    Cr0[6] = 5.;               /* 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 */
    i2 = (int)(50. / dt);      /* incident flux step at time = 50 h */
    for (i=0; i<=i2; i++)
          FRH[i] = 133.;
    for (i=i2+1; i<=H; i++)
          FRH[i] = 305.;            /* incident flux step */
    for (i=0; i<=H; i++)
          dilH[i] = 0.035;     /* dilution rate constant */
    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 */
    }

    /* 4_ illuminated surface fraction */
    fI = 0.60;
}
else if (choisim == 3)
{
    /* ========================================================
       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 */
    }

    /* 4_ illuminated surface fraction */
    fI = 1.0;
}
else if (choisim == 4)
{
     /* ========================================================
        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 */
    }

   /* 4_ illuminated surface fraction */
   fI = 1.0;
}
else if (choisim == 5)
{
    /* ======================================================
       Simulation according to TN19.3 , Appendix 7 , Figure 9
       ====================================================== */

    /* 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] = 25.;            /* incident flux step */
    for (i=0; i<=i1; i++)
            dilH[i] = 0.;
    for (i=i1+1; i<=H; i++)
            dilH[i] = 0.026;         /* optimal 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 */
        }

      /* 4_ illuminated surface fraction */
      fI = 1.0;
   }

/*
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
      */
      sbq(fI, 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\n",tH[i],FRH[i],dilH[i],CrH[i][1],CrH[i][2],CrH[i][3],CrH[i][4],CrH[i][5],CrH
[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],chonspH[i][2],chonspH[i][3],chonspH[i][4],chonspH[i][5
]);
        }
        fclose(pf);

}
```

```
/*
sbq.c

Main programme for the estimation of the Spirulina Biomass Quality
based on the first principles model built by LGCB (TN 19.1 and 19.2)

J.J. Leclercq
ADERSA
July 1999
Version 2.0 (according to TN 35.3)

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

Synopsis:
=========
sbq(fI, H, Cr0, FRH, dilH, CiH, CrH, fmasbioH, chonspH)
Input arguments :
-----------------
fI : illuminated surface fraction
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  sbq(fI, H, Cr0, FRH, dilH, CiH, CrH, fmasbioH, chonspH)
double fI, 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(fI, 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(fI, FR, dil, Ci, Cr)
double fI, 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(fI,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] = ri[1] - ri[8];                                       /* 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(fI,CPC,CCH,CXV,FR,RXA,REPS)
double  fI, 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 * fI * kstep;

    /*
    REPS1 = 2. * mupMEPS * CPC * sEPS * fI * 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.;
    */
    /* Modified on May 11th 1999 */
    *REPS = 2. * mupMEPS * CPC * sEPS * fI * kstep;

}
```

```
/*
        pargene.h

        general parameters

        July 1999
        Version 2.0 (according to TN 35.3)
*/

# define nT 2000    /* number max of time steps (modifiable by the user) */
# define nZ 1000    /* number max of integration steps along the radius (modifiable
by 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

        July 1999
        Version 2.0 (according to TN 35.3)
*/

#define RT          .045    /* reactor radius (m) */
#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.e-4   /* half satur. cste for nitrate limitation (kg/m3) */
#define KS          2.5e-4  /* half satur. cste for sulfate limitation (kg/m3) */
#define KPC         .06     /* half satur. cste for phycocyanin limit. (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) */
```