UNIVERSITE
BLAISE PASCAL
CLERMONT-FERRAND
LGCB
Laboratoire de Génie Chimique Biologique
63177 Aubière Cedex, FRANCE
Tel: (33).04.73.40.74.30
Fax: (33) 04.73.40.78.29
email: lgcb@gecbio.univ-bpclermont.fr
sky@gecbio.univ-bpclermont.fr

MELiSSA

Memorandum of Understanding
ECT/FG/CB/95.205
ESA/ESTEC P.O.: 161 081

TECHNICAL NOTE 32.2

NitriSim -version 2.3-
Fixed Bed Nitrifying Column Simulation Program
-
Presentation and utilisation of the software

version 1
Issue 0

L. Poughon

October 1997

Document change log

Version	Issue	Date	Observations
0	0	July 1997	Draft version NitriSim software v2.3
1	0	October 1997	Final version NitriSim software v2.3

CONTENT

INTRODUCTION 1

I - BASIS OF THE NITRIFYING MODEL 2

I.1 - COLUMN DESIGN, MODELLING AND HYDRODYNAMIC DESCRIPTION 2
I.2 - BIOFILM MODEL 4
I.3 - BIOLOGICAL MODEL 5
I.4 - STEADY STATE ALGORITHM 6
I.5 - MODEL VARIABLES AND PARAMETERS 9

II - NITRISIM 2.3 PROGRAM DESIGN 12

II.1 - OVERVIEW AND SPECIFIC REQUIREMENTS 12
II.1.1 - Architectural design of the program *13*
II.2 - SUBROUTINES INVOLVED 16
II.2.1 - Subroutines description *16*
II.2.2 - Subroutines hierarchy *30*
II.2.3 - Subroutines listing *31*
II.3 - IMPORTANT REMARKS 31
II.3.1 - Differences between DOS and UNIX versions. *31*
II.3.2 - Common variables *31*

III - NITRISIM V2.3 USER MANUAL 33

III.1 - INSTALL PROCEDURE 33
III.2 - MAIN MENU 34
III.3 - MODEL PARAMETERS MANAGEMENT MENU 35
III.3.1 - Option 1: Design of the column *35*
III.3.2 - Option 2: stoichiometric equations *36*
III.3.3 - Option 3: Biological kinetics *38*
III.3.4 - Option 4: Compounds involved and their physico-chemical associated constants *38*
III.4 - BIOFILM DIFFUSION SIMULATION 39
III.5 - NITRIFYING PROCESS SIMULATION MENU 40
III.5.1 - Complete process (option 1 and 2) *41*
III.5.2 - Steady state process *44*
III.6 - SIMULATIONS AND RESULTS STORAGE FILES 46
III.6.1 - Names *46*
III.6.2 - Examples *46*

Appendices

T.N. 32.3: NitriSim

L. Poughon.
Laboratoire de Génie Chimique Biologique
63177 AUBIERE Cedex. France.

Introduction

NitriSim is a FORTRAN 77 software for the simulation of a nitrifying fixed bed column. The program is based on models developed in TN 27.1, 27.2, 27.3 and 32.1, which involved mass-balanced description of the nitrification process, biological kinetics (including saturation constants and inhibitory constants), physico-chemical equilibria (pH, gas-liquid, liquid-biofilm) and hydrodynamic description of the column.

This note is divided into 3 parts.

The first reviews the basis of the nitrifying column model, in order to present the different compounds, and parameters involved in the NitriSim model.

In a second part the NitriSim program and its subroutines are described.

The third part is dedicated to all users of the NitriSim software. It details all the options and the steps to correctly use the software.

This note must be released with the floppy disk containing the NitriSim software version 2.3 and its sources. It must be underlined that at the present time, the simulations have not been confronted to experimental data. The kinetic parameters (μmax, saturation and inhibitory constants) had to be verified. It must be kept in mind that the present model is for an autotrpohic nitrification in a Biostyr beads fixed bed reactor.

This section reviews the basis of the model established for the development of the NitriSim program - Nitrifying fixed bed column Simulation program -. The detailed model can be found in TN 27.1, 27.2, 27.3 and 32.1.

I - Basis of the nitrifying model

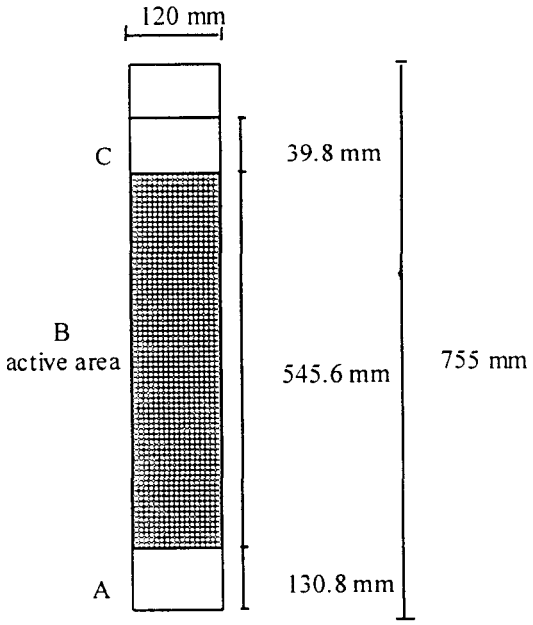
The nitrifying model developed is based on a fixed bed column process. It is first design to oxidise NH_3 and NO_2^- using the autotrophic ability of 2 strains *Nitrosomonas europea* and *Nitrobacter winogradskyi*. The model take into account the pH equilibria of every compound involved, as well as the gas-liquid equilibria, and in some cases the biofilm diffusion limitation.

I.1 - Column design, modelling and hydrodynamic description

The nitrifying fixed bed column is described by using a N-tanks in series model, including back-mixing between each tank. Back-mixing is a way to represent the perfect mixing behaviour of the column, while the number of tanks is a way to represent the plug-flow behaviour of the column.

The column is sub-divided into 3 parts. Part A correspond to the bottom of the column, part B is the fixed bed ,packed with biostyr beads, and represented by N-tanks, and part C is the top of the column.

Column:
Diameter: 120 mm.
Height: 716.2 mm occupied by beads + liquid + gas (calculated from the occupied volume of 8,1 l).
755 mm total (calculated from the total volume of 8.53 l)
Volume: 8.1 l (experimental occupied volume measured at UAB Laboratory).
8.53 l (total volume calculated from the dimensions of the UAB column)
Void fraction ε^{col}: 0.52
Liquid fraction ε_L^{col}: 0.475
Gas fraction ε_G^{col}: 0.045

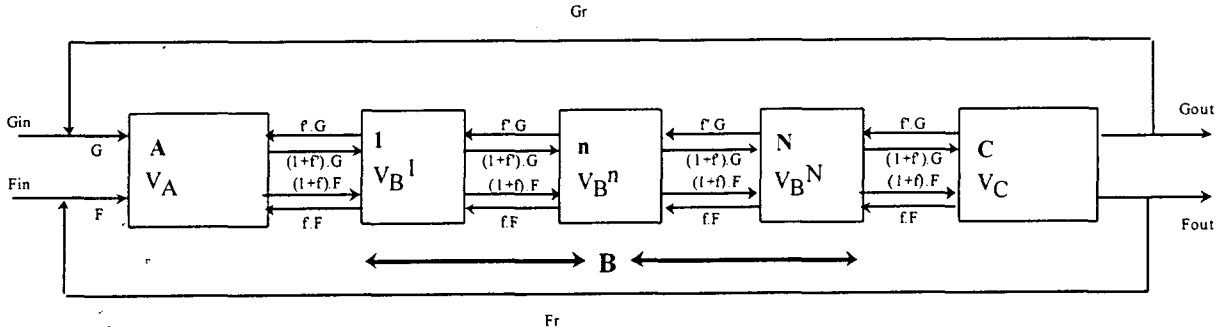


Part A
Volume: 1.48 l

Part B (active fixed bed area)
Volume: 6.17 l
ε: 0.37
ε_L: 0.33
ε_G:0.04

Part C
Volume: 0.45 l

The column can be represented by the following scheme, where the hydrodynamic parameters (e.g., liquid and gas flow rates, liquid and gas back-mixing, inputs and outputs flow rates) are reported.



The balance equations modelling the column are defined for each part of the column for the liquid and the gas phase.

Part A

$$\frac{\varepsilon_L}{\varepsilon} V_A \frac{dC_{Si}^A|_L}{dt} = F_{in}.C_{Si}^{in}|_L + F_r.C_{Si}^{out}|_L - (f+1).F.C_{Si}^A|_L + f.F.C_{Si}^1|_L + \frac{\varepsilon_L}{\varepsilon} V_A.\phi_{Si}^A|_{GL}$$

$$\frac{\varepsilon_G}{\varepsilon} V_A \frac{dC_{Si}^A|_G}{dt} = G_{in}.C_{Si}^{in}|_G + G_r.C_{Si}^{out}|_G - (f'+1).G.C_{Si}^A|_G + f'.G.C_{Si}^1|_G - \frac{\varepsilon_L}{\varepsilon} V_A.\phi_{Si}^A|_{GL}$$

Part B (fixed bed)

$$\varepsilon_L.V_B^n.\frac{dC_{Si}^n|_L}{dt} = (1+f).F.C_{Si}^{n-1}|_L + f.F.C_{Si}^{n+1}|_L - (f+1).F.C_{Si}^n|_L - f.F.C_{Si}^n|_L + \varepsilon_L.V_B^n.\phi_{Si}^n|_{GL} + \varepsilon_L.V_B^n.\phi_{Si}^n|_{LB}$$

$$\varepsilon_G.V_B^n.\frac{dC_{Si}^n|_G}{dt} = (1+f').G.C_{Si}^{n-1}|_G + f'.G.C_{Si}^{n+1}|_G - (f'+1).G.C_{Si}^n|_G - f'.G.C_{Si}^n|_G - \varepsilon_L.V_B^n.\phi_{Si}^n|_{GL}$$

Part C

$$\frac{\varepsilon_L}{\varepsilon} V_C \frac{dC_{Si}^C|_L}{dt} = (f+1).F.C_{Si}^N|_L - f.F.C_{Si}^C|_L - F_r.C_{Si}^C|_L - F_{out}.C_{Si}^C|_L + \frac{\varepsilon_L}{\varepsilon} V_C.\phi_{Si}^C|_{GL}$$

$$\frac{\varepsilon_G}{\varepsilon} V_C \frac{dC_{Si}^C|_G}{dt} = (f'+1).G.C_{Si}^N|_G - f'.G.C_{Si}^C|_G - G_r.C_{Si}^C|_G - G_{out}.C_{Si}^C|_G - \frac{\varepsilon_G}{\varepsilon} V_C.\phi_{Si}^C|_{GL}$$

n being the number of the tank, $1<n<N$, $\phi_{Si}^n|_{GL}$ the gas-liquid transfer term (mol/unit volume. unit time) and $\phi_{Si}^n|_{LB}$ the liquid-biofilm transfer term (mol/unit volume. unit time)

The parameters involved in these equations can be classified in a first approach in three categories: flow rates variables, column design variables and transfer rate terms

Three categories of flow rate can be defined in the model:

- the input flow rates: F_{in} (liquid) and G_{in} (gas)
- the recycling flow rates represented by the recycling ratio: ($\frac{\text{Inlet flow rate}}{\text{Recycling flow rate}}$)

  R_L (liquid) and R_G (gas)
- the back-mix flow fractions: f (liquid) and f' (gas) (see the previous scheme)

The two first kinds can be manipulated while the back-mix flow fractions depend on the column design and on the flow rate inside the column. A relation exists between f for N-stirred tank model and the axial dispersion term E_X in the plug flow model (TN 27.1), but even if E_X can be approximated in two phases fluidized and fixed beds (TN 27.1), there is no evidence that these relations can be used for the nitrifying column.

From DTS experiments, a linear relation was determined between f (liquid back-mixing) and N (TN 23.7). If the NitriSim user changes the number a tanks for the model, he can use this relation to estimate a value for the back-mixing and to change this parameter in the model (cf. section III..3.1).

I.2- Biofilm model

The model chosen for mass transfer in a biofilm is based on the following assumptions (TN 27.1, TN27.3):

1 - Steady state transfer limitation in the biofilm
2 - No transfer resistance between the biofilm and the bulk phase
3 - A plane geometry is supposed for the biofilm

The model is represented, at every time t, by the following system:

$$\left.\frac{d^2C_{Si}|_B}{db^2}\right|_b = -\frac{1}{D_{Si}|_B}\left[r_{Si}^{Ns}|_B + r_{Si}^{Nb}|_B\right]$$

with the boundary conditions

$$\left.\frac{dC_{Si}^{\alpha}|_B}{db_B}\right|_{b=R_0} = 0$$

$$C_{Si}^{\alpha}|_B = C_{Si}^{\alpha}|_L \qquad \text{at} \qquad b = R_0 + h_b^n$$

where R_O is the bead radius, h_b the biofilm thickness and b denotes for the the abscissa throughout the biofilm.

The system is solved using a Runge Kutta Merson algorithm of the 4th order. The algorithm developed for solving the biofilm diffusion is presented in TN 27.3.

In fact, the results obtained have shown that with the **autotrophic growth** conditions actually used for the simulations of the process, the oxygen diffusion limitation in the biofilm (the most important and the first limitation that can appear in the fixed bed nitrifying process) was far to be reached. For this reason, the biofilm diffusion model, which greatly increases the computation time needed to solve the model, is not included in the dynamic model of the process. Nevertheless, an option is available in the NitriSim program in order to calculate the biofilm concentration profile for a defined column concentration profile at a given time t.
Thus, for the biofilm simulation, the liquid-biofilm transfer term previously described is defined as:

$$\phi_{Si}^{n}|_{LB} = -D_{Si}|_{LB} \left.\frac{\partial^2 C_{Si}^{n}|_B}{\partial b^2|_B}\right|_{B=hb+Ro}$$

and for the model without biofilm limitation (e.g. dynamic simulation in the NitriSim program), the liquid -biofilm term is defined as:

$$\phi_{Si}^{n}|_{LB} = r_{Si}^{Ns} + r_{Si}^{Nb}$$

I.3 - Biological model

The biological model of the nitrification is based on a structured stoichiometric description of the biological nitrification by *Nitrosomonas* and *Nitrobacter* (TN 23.2 and 32.1) and on kinetic model for the growth, the substrate consumption/production and for the maintenance.

The growth yields used in the kinetic model reported below, are calculated from the stoichiometries (TN 23.2, TN 32.1 and section 1.1.5). The kinetic parameters, the maximum growth rate, maintenance coefficients, the half saturation constant and the inhibitory constants are taken from the literature and are reported in section 1.1.5. This kinetic parameters are for autotrophic growth and were not validate with real experiments of autotrpphic nitrification in the MELiSSA fixed bed column. The saturation constants for CO_3^{2-}, HPO_3^{2-} and SO_4^{2-} are set to 10^{-8} mol/l. This arbitrary value was chosen enough low to avoid limitations phenomena.

Thus the program can not be used its current version 2.3 to simulate the biological limitation of carbon, phosphorus and sulfur.

The kinetic growth model is based on the following relations (detailed in TN 27.1):

$$r_X^{Ns} = \mu^{Ns}.C_{X-Ns}\big|_B + Y_{X/Smt}^{Ns}.m^{Ns}.\left(\frac{\mu^{Ns}}{\mu_{max}^{Ns}} - 1\right).C_{X-Ns}\big|_B$$

$$r_X^{Nb} = \mu^{Nb}.C_{X-Nb}\big|_B + Y_{X/Smt}^{Nb}.m^{Nb}.\left(\frac{\mu^{Nb}}{\mu_{max}^{Nb}} - 1\right).C_{X-Nb}\big|_B$$

with

$$\mu = \mu_{max}.\prod_{\substack{\text{Limiting}\\ \text{substrates } j}} \frac{C_j\big|_B}{(Ks_{Sj} + C_j\big|_B).Ki} \quad \text{where} \quad Ki = \prod_{\substack{\text{Inibitory}\\ \text{Substrate } k}} \left(1 + \frac{C_k\big|_B}{I_k}\right)$$

The biomass released in the liquid from the biofilm on the beads is represented by:

$$r_X^{Ns-free} = K_{wo}.r_X^{Ns}$$
$$r_X^{Nb-free} = K_{wo}.r_X^{Nb}$$

The consumption/production rates of substrates are expressed by the following relations:

$$r_{Si}^{Ns} = \frac{1}{Y_{X/Si}^{Ns}}.r_X^{Ns} + \frac{1}{Y_{Smt/Si}^{Ns}}r_m^{Ns}$$

$$r_{Si}^{Ns} = \frac{1}{Y_{X/Si}^{Ns}}.r_X^{Ns} + \frac{1}{Y_{Smt/Si}^{Ns}}r_m^{Ns}$$

The description and the current setting values of the parameters involved in the biological model are reported in tables of section 1.1.5

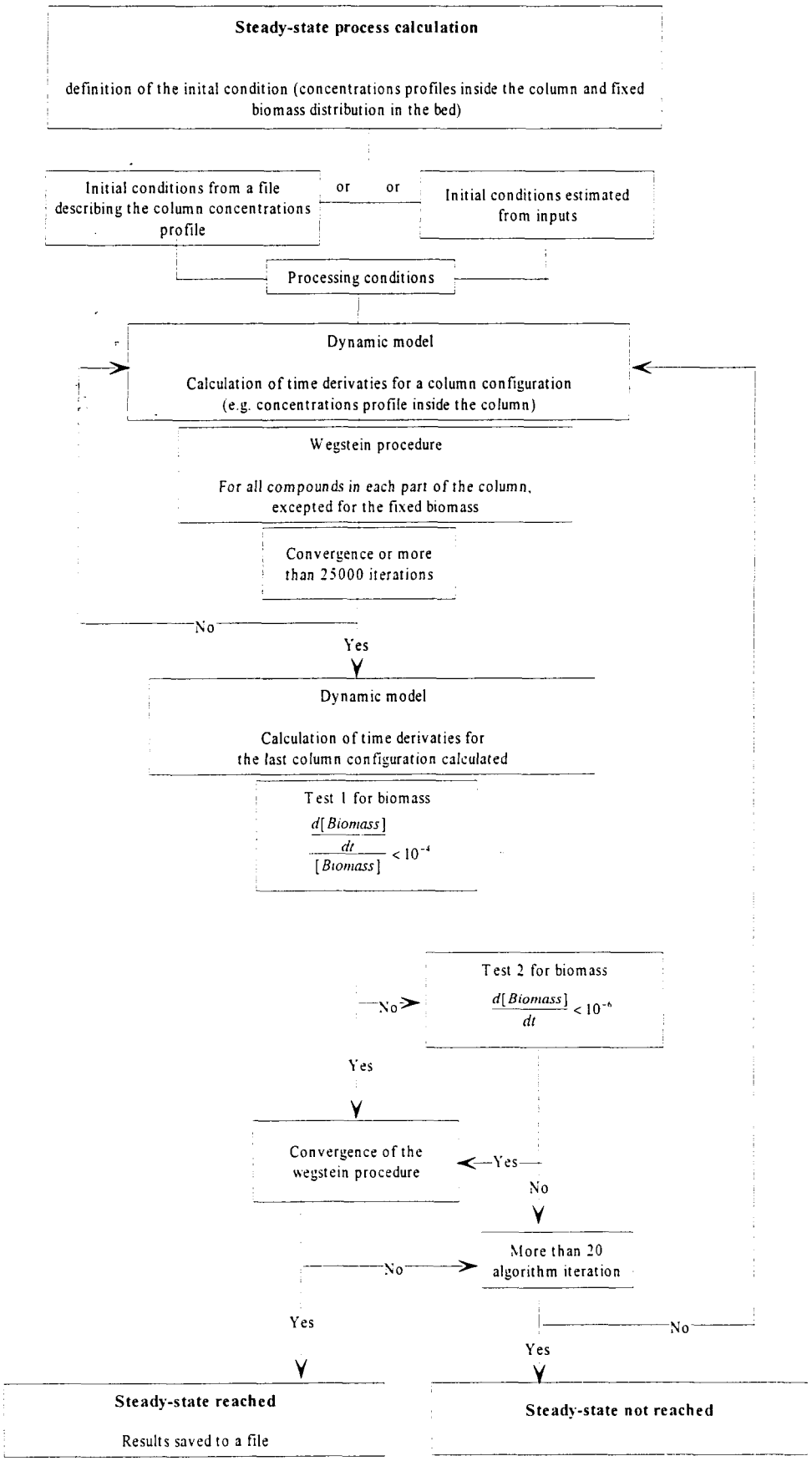
I.4 - Steady state algorithm

From the previous modelling works on the nitrifying fixed bed column, an algorithm for the determination of the steady-state of the column for defined process condition was developed.

The algorithm is based on two successive iterative methods:

- a Wegstein procedure for the calculation of the steady-state of the compounds in the gas and in the liquid phases
- a pool of tests to determined the steady-state for the fixed biomass

The algorithm for steady-state calculation is represented by the following scheme:



The sensitivity of the 2 iterative procedures for the determination of the steady-state can be chosen by the user. They have the following default setting values:

- 10^{-3} for the Wegstein procedure. This means in the algorithm that steady state is reached when for each compound if:

$$\frac{\frac{d[compound]}{dt}}{[compound]} < 2.10^{-4} \text{ or } 0.02\%$$

- 10^{-6} for the tests on biomass. This mean that steady-state is assumed for the fixed biomass when one of these two conditions is satisfy:

$$\frac{\frac{d[Biomass]}{dt}}{[Biomass]} < 100.10^{-6} \quad \text{i.e.} \quad 10^{-4} \text{ or } 0.01\%$$

$$\frac{d[Biomass]}{dt} < 10^{-6}$$

I.5 - Model variables and parameters

List of the compounds involved in the model and their physico-chemical constants

Compound (non ionic form / first dissociated form / second dissociated form / third dissociated form)	pH equilibrium K_A (25°C) [Adimensional]	Gas-liquid equilibrium k_i (25°C) [Adimensional]
NH_3 ⇆ NH_4^+	1.762 10^{-5} [TN 23.1.]^c	1.173 10^{-2} [TN 23.1.]^c
HNO_2 ⇆ NO_2^-	Complete dissociation	
HNO_3 ⇆ NO_3^-	Complete dissociation	
CO_2 ⇆ HCO_3^- ⇆ CO_3^{2-}	4.320 10^{-7} [TN 17.1.]c 4.557 10^{-11} [TN 17.1.]^c	1635 [TN 17.1]^c
O_2		4.272 10^4 [TN 17.1]^c
H_3PO_4 ⇆ $H_2PO_4^-$ ⇆ HPO_4^{2-} ⇆ PO_4^{3-}	6.918 10^{-3} [TN 27.1.] 6.166 10^{-8} [TN 27.1.] 4.780 10^{-13} [TN 27.1.]	
H_2SO_4 ⇆ HSO_4^- ⇆ SO_4^{2-}	Complete dissociation (10^{10}) 1.047 10^{-2} [TN 27.1.]	
H_2O ⇆ H+/O ⇆ H^-	10^{-14}	Po=0.031 atm ^{c}
Biomass *Nitrosomonas*		
Biomass *Nitrobacter*		
Default (no equilibrium)	0.	0.

c= calculated value

List and values of the kinetic parameters

			Reference	Remarks
Growth rates				
μ_{max}^{Ns}	$5.7\ 10^{-2}\ h^{-1}$		[TN 27.2]	mean values calculated
μ_{max}^{Nb}	$3.6\ 10^{-2}\ h^{-1}$		[TN 27.2]	from several
m^{Ns}	3.38 10^{-3}		[TN 27.2]	continuous cultures
m^{Nb}	7.92 10^{-3}		[TN 27.2]	
Limiting substrate [Si]	K_{Si}^{Ns}	K_{Si}^{Nb}		
NH_3	6.625 10^{-5} mol/l	-	[TN 27.2]	Model parameter
NO_2^-	-	3.6 10^{-4} mol/l	[TN 27.2]	values for a fixed bed
O_2	5.05 10^{-6} mol/l	1.7 10^{-5} mol/l	[TN 27.2]	of carragenan beads
HCO_3^-	10^{-8} mol/l	10^{-8} mol /l		no carbon limitation
HPO_4^{2-}	10^{-8} mol/l	10^{-8} mol /l		no phosphate limitation
SO_4^{2-}	10^{-8} mol/l	10^{-8} mol /l		no sulphur limitation
Other (non limiting)	0.	0.		
Inhibitory substrate Si	I_{Si}^{Ns}	I_{Si}^{Nb}		
NO_2^-	-	0.159	[TN 27.3]	
NO_3^-	-	0.188	[TN 27.3]	
Others (no inhibition)	10^{10}	10^{10}		
Growth Yield of substrate [Si] (from Stoichiometric coefficients)	$Y_{X/Si}^{Ns}$ [Stoic. Coef.]	$Y_{X/Si}^{Nb}$ [Stoic. Coef.]		g biomass / mol substrate Si Algebraic value
NH_3	-6.0271 [-3.8400]	109.8448 [-0.2107]	[TN 32.1]	
NO_2^-	6.3772 [3.6292]	1.4611 [-15.8398]	[TN 32.1]	
NO_3^-	-	1.4611 [15.8398]	[TN 32.1]	
O_2	5.3020 [-4.3652]	3.3830 [-6.8413]	[TN 32.1]	
HCO_3^-	23.1443 [-1]	23.1443 [-1]	[TN 32.1]	
HPO_4^{2-}	-1701.7867 [-0.0136]	-1701.7867 [-0.0136]	[TN 32.1]	
SO_4^{2-}	-5644.9512 [-0.0041]	-5644.9512 [-0.0041]	[TN 32.1]	
H^+	6.4401 [3.5938]	65.3794 [-0.3540]	[TN 32.1]	
OH^-	23.1443 [1]	23.1443 [1]	[TN 32.1]	
C-molar weight of Biomass	23.1443	23.1443	[TN 32.1]	g/C-mol
Maintenance Yields of Substrate Si	$Y_{Smt/Si}^{Ns}$	$Y_{Smt/Si}^{Nb}$		mol maintenance substrate / mol Si Algebraic value
NH_3	-1			(Smt for Ns)
NO_2^-	1	-1		(Smt for Nb)
NO_3^-		1		
H_2O	1			
H+	1			
O_2	-0.5	-1.5		

Column and flow rates (standard values)

Column	Height: 716.2 mm diameter:120 mm Volume part A:1.48 l Volume part B:0.45 l Pressure: 1 atm Temperature: 25°C Liquid void fraction: 0.33 Gas void fraction:0.04
Fixed bed (active area)	Particle diameter: 4.1 mm N: 5 (Number of tanks equivalent for the fixed bed- part B) f: 155% (liquid back-mixing) f': 0% (gas back-mixing)
Input flow rates	Fin: 2.8 ml/min Gin: 0.03 l/min
Recycling ratio	R_L: 6.42 R_G: 99
Gas composition	CO_2: 0.004% O_2: 21% H_2O: 0%
Liquid composition	NH_3: 7.14 mmol/l (100 mg N-NH_3/l) H_3PO_4: 0.1 mmol/l (no limiting) H2SO3: 0.1 mmol/l (no limiting)
Gas-Liquid transfer parameters	$K_La\|_{O2}$: 51 h^{-1} (0.014 s^{-1}) $K_La\|_{CO2}$: 51 h^{-1} (0.014 s^{-1}) $K_La\|_{H2O}$: 500 h^{-1} $K_La\|_{NH3}$: 0 h^{-1} (no gas-liquid transfer) [Default values: 0. (No gas-liquid transfer)]
Kinetics parameters	K_{wo}: 0% (Biomass releasing term)
Biofilm	No biofilm limitation (in the dynamic model only)

II - NitriSim 2.3 program design

II.1 - Overview and specific requirements

The NitriSim program is written in FORTRAN 77, and can be compiled as well under DOS and UNIX operating systems. The program is composed of 13 subroutines (see section 1.2.3) contained in 8 source files for this NitriSim version 2.3:

MAIN23.FOR
- MAIN program

CALCOL23.FOR
- CALCOL subroutine
- DERIV subroutine

FILM23.FOR
- BIOFILM subroutine
- DERIVBIO subroutine
- RKMER2 subroutine

EDCONF23.FOR
- EDITCONF subroutine

CONFSI23.FOR
- CONFIGSIM subroutine

SAVECF23.FOR
- SAVECONF subroutine

SAVPAR23.FOR
- SAVEPARA subroutine

SAVSIM23.FOR
- SAVESIM subroutine

SKYMER23.FOR
- SKYMER subroutine

WEG23.FOR
- WEG subroutine

The NitriSim binary executable has a length of 270 Ko. It requires the presence under the same directory of specific data files **(This data files are the default files which are automatically modified when the user change on the command line one of the parameter that they contain.)**:

PHYTRANS.DAT : Default physical parameters for the compounds
PHYPH.DAT : Data file containing constants for pH calculation
CORPS.DAT : Data file containing the compounds names
CINET.DAT : Default kinetic parameters
STOIC.DAT : Data file containing stoichiometric coefficients
FLOWCOL.DAT : Default column design (flow parameters)
CARCOL.DAT : Default column design

The length of the files into which the results are stored depends on the number of values that the user has chosen to keep during a simulation. An average of 100-200 Ko can be assumed for the storage of 60 concentrations profiles at different times for a column of 5-tanks equivalent for the fixed bed.

To install and properly use NitriSim a minimum of free disk space of 500 Ko is required. Then the program can be run on 3.5 inch floppy disk if needed.

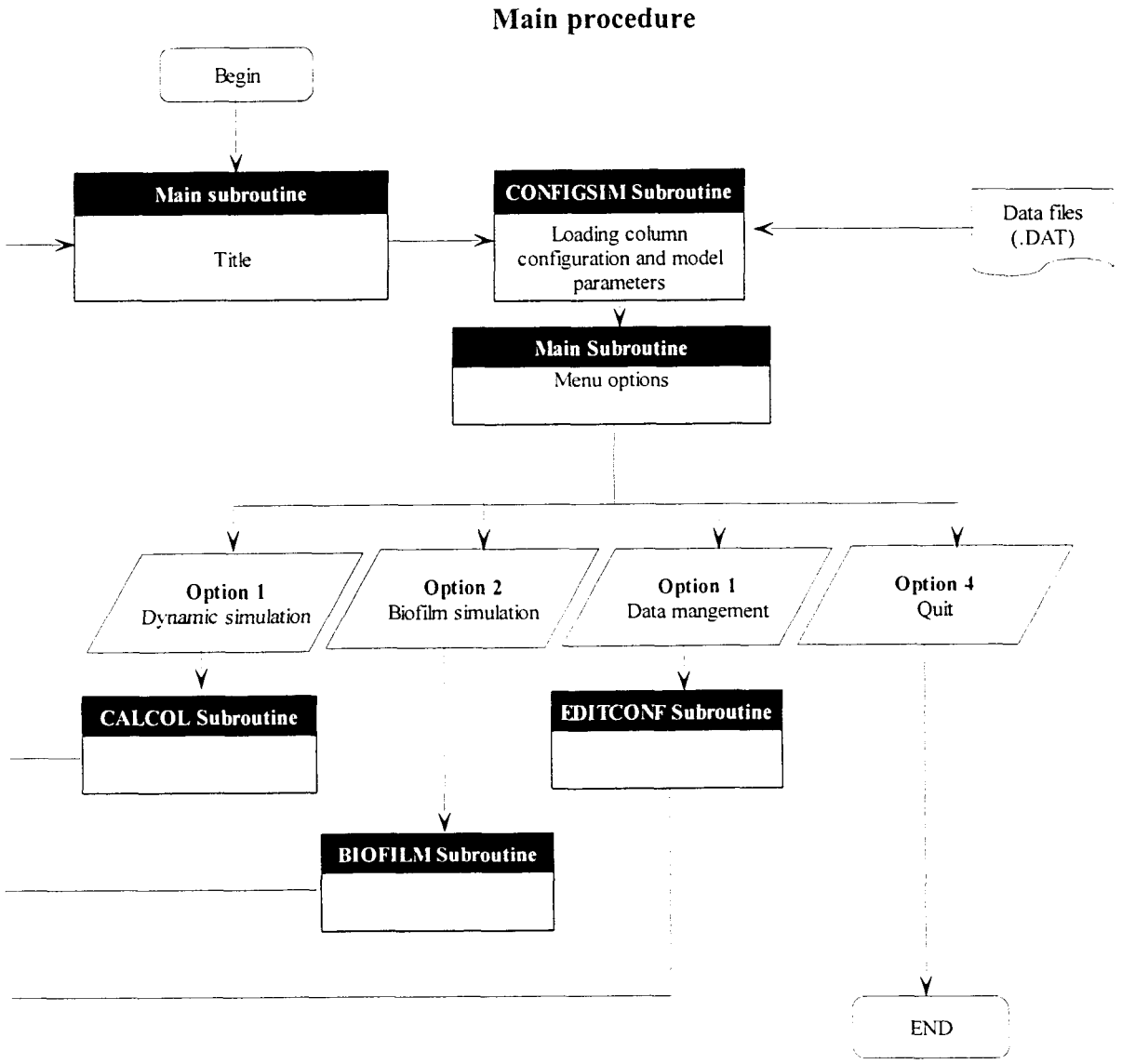
The computationnal time required for a simulation greatly depends on:

- the number of equivalent tanks of the fixed bed for dynamic simulations (without biofilm diffusion)
- the sensibilities chosen for the test in the steady-state simulation (see section 1.1.4 and section 2.5.2).

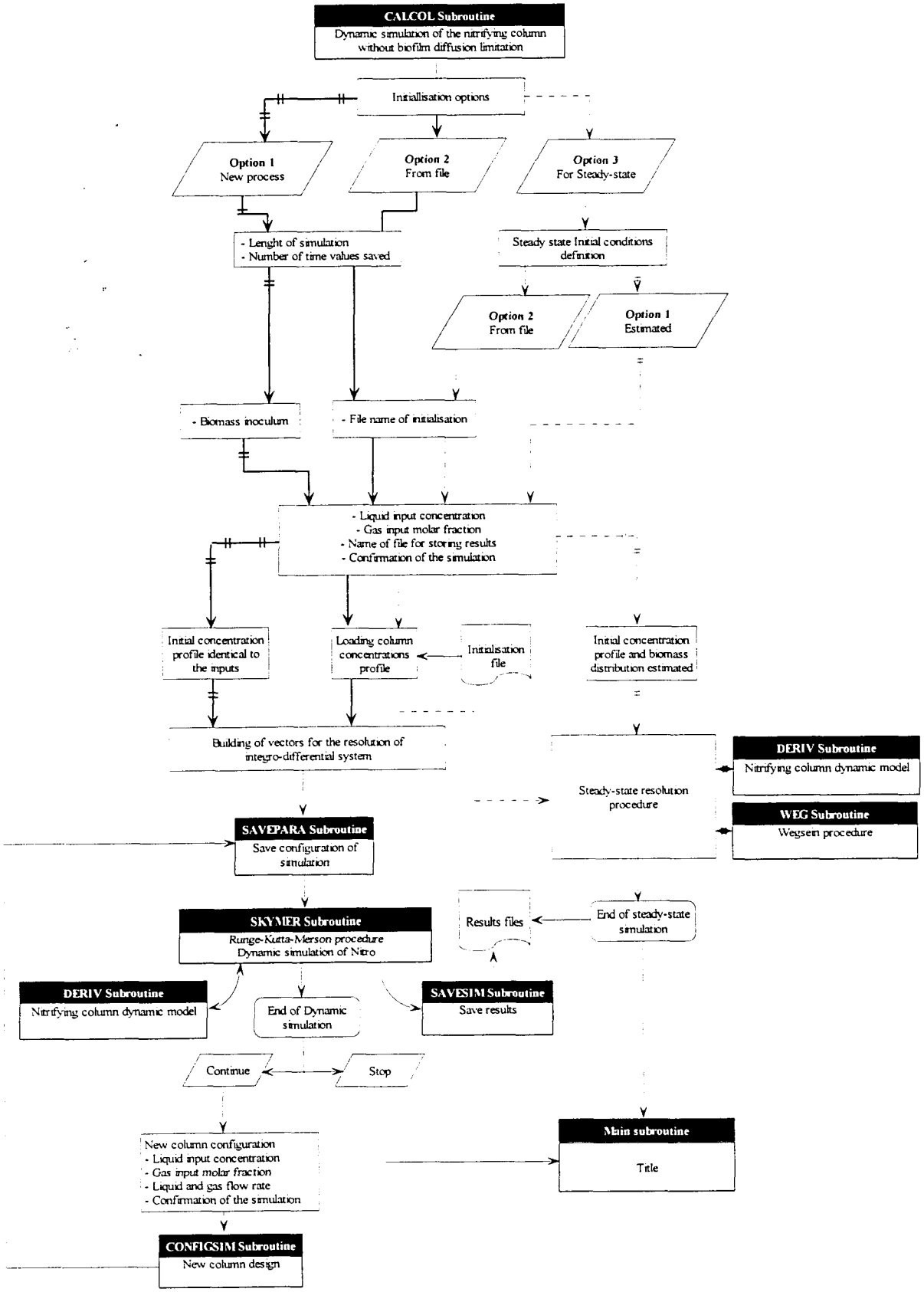
With a Pentium based CPU at a frequency of 100MHz, the dynamic simulation of a process of 100 hours with a column of 5-tanks equivalent for the fixed bed, takes around 50 minutes. The running time is roughly doubled for a 10 tanks configuration.

II.1.1 - Architectural design of the program

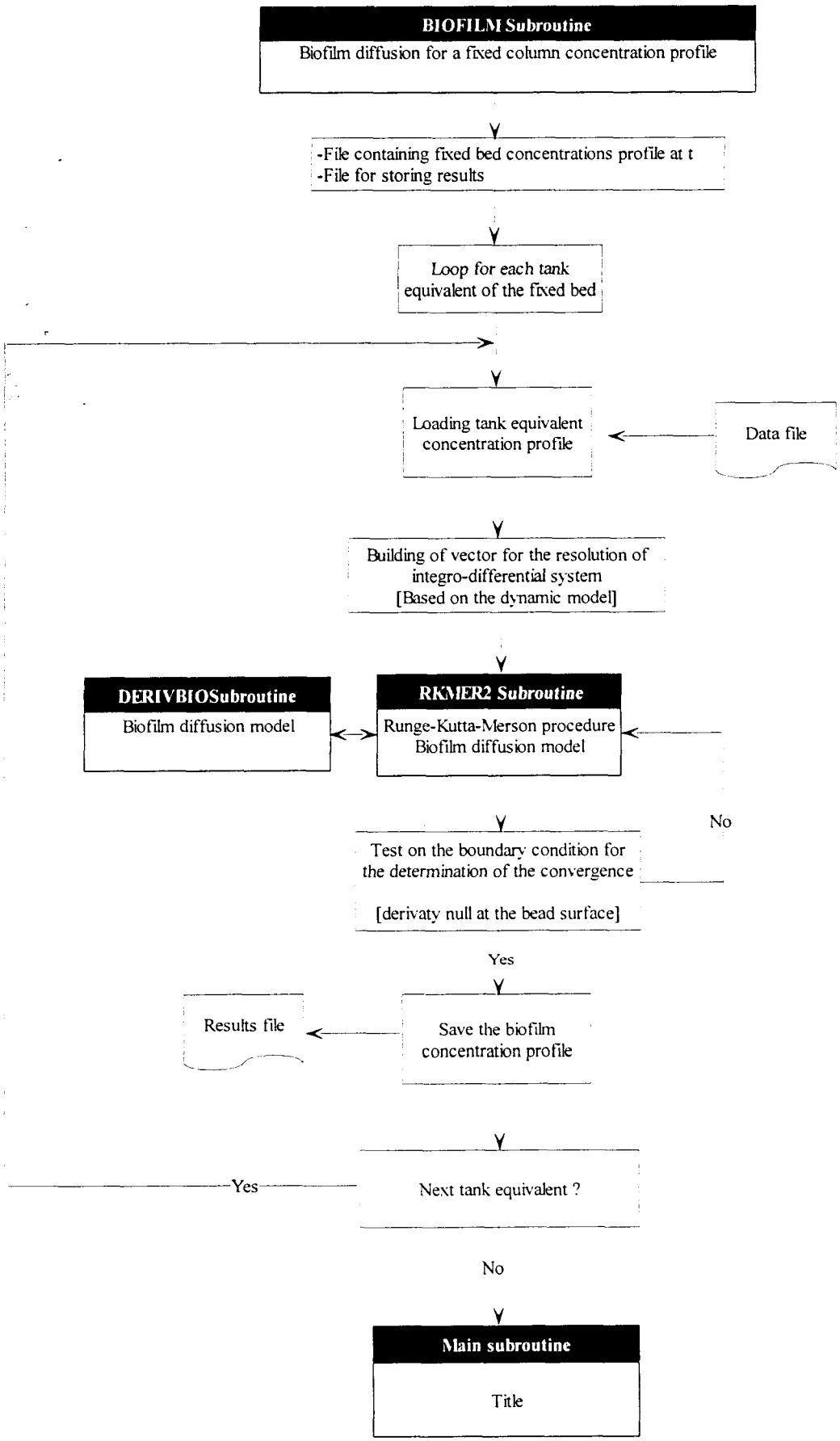
The following diagrams resume the architectural design of the program and gives the hierarchical call of the different subroutines.



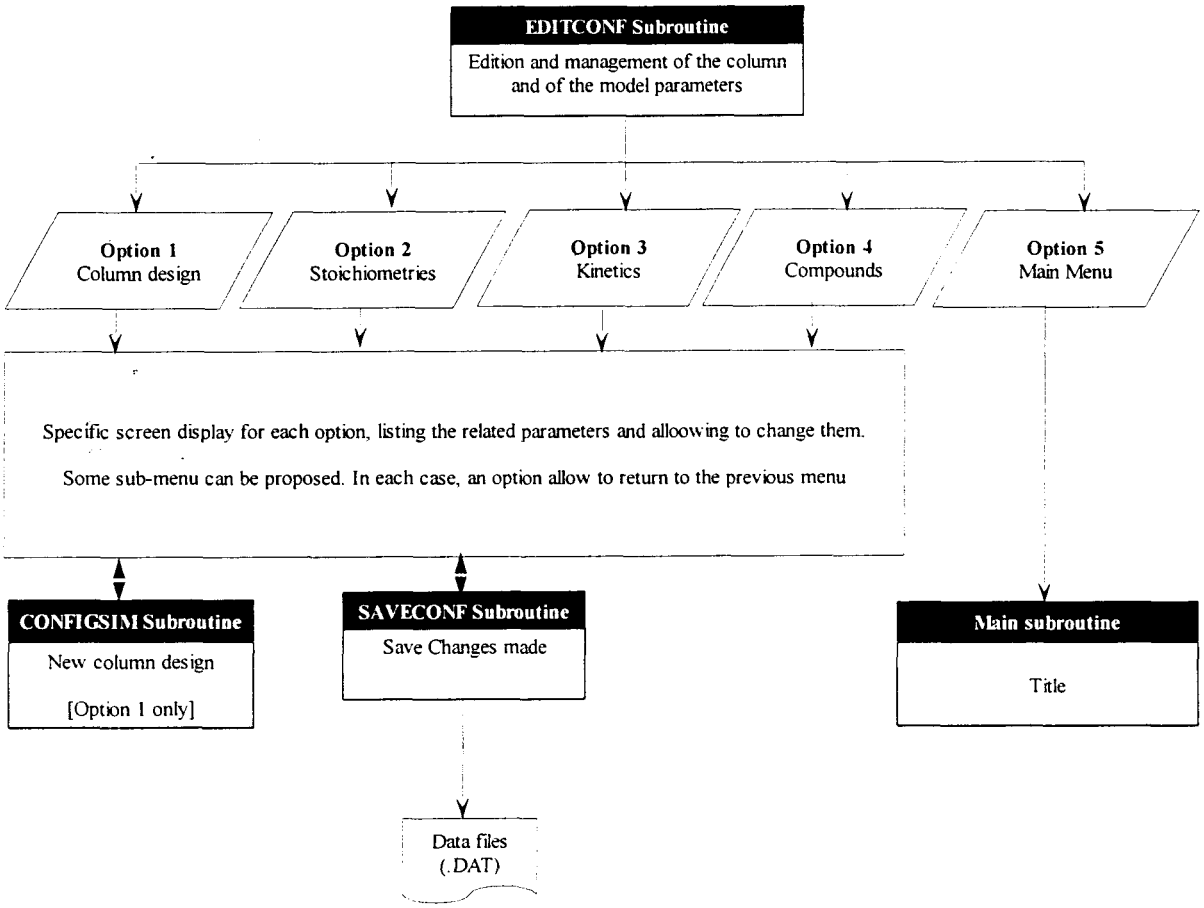
Option 1: Dynamic simulation



Option 2: Biofilm diffusion



Option 3: Management



II.2 - Subroutines involved

II.2.1 - Subroutines description

The following table give an overview of the different subroutines involved in the program. They detail their functions, their interrelations with the other subroutines and their entry/output variables. It must be outlined that most part of the variables used in each subroutine are shared between the different subroutines via a "COMMON" statement . The list of the "COMMON" is reported in section 1.2.4.

Subroutine: CALCOL
Function
Simulation of the nitrifying fixed bed without a biofilm diffusion limitation term in the dynamic model. Simulation of transient and steady-state processes
Entry variables
None
Output variables
None
Entry-output variables
None
Called by
Main program
Subroutines called
SKYMER SAVEPARA CONFIGSIM

Subroutine: CONFIGSIM
Function
Loading of data files Column design Kinetic parameters compounds and constants Calculations for the column (void fraction, pressure drop...)
Entry variables
MENU: determine the origin of the call
Output variables
None
Entry-output variables
None
Called by
Main program CALCOL EDITCONF
Subroutines called
None

Subroutine: EDITCONF
Function
Edition and management of all the model parameters and of the column design loaded in data files
Entry variables
None
Output variables
None
Entry-output variables
None
Called by
Main program
Subroutines called
CONFIGSIM SAVECONF

Subroutine: BIOFILM
Function Calculation of the concentrations in the biofilm for a fixed concentration profile inside the column.
Entry variables None
Output variables None
Entry-output variables None
Called by Main Program
Subroutines called DERIVBIO RKMER2

Subroutine: MAIN program
Function
Definition of program variables and common variables Main options of the program
Entry variables
None
Output variables
None
Entry-output variables
None
Called by
None
Subroutines called
CALCOL CONFIGSIM BIOFILM EDITCONF

Subroutine: SAVECONF
Function Save changes made by the parameters management in the corresponding data file
Entry variables Menu: origin of the call. Determine which file must be changed
Output variables None
Entry-output variables None
Called by EDITCONF
Subroutines called None

Subroutine: SAVEPARA
Function Save configuration of the column (design, flow rates, inlet concentrations) for the current simulation in .CNF file
Entry variables None
Output variables None
Entry-output variables None
Called by CALCOL
Subroutines called None

Subroutine: SAVESIM
Function
Save results of the current simulation for a time t
Entry variables
X: Time of simulation for which the results are stored Y: Vector containing concentrations in every segment of the column K: Number of call of the subroutine - (number of saved results during the simulation is fixed in dynamic simulation)
Output variables
None
Entry-output variables
Called by
CALCOL SKYMER
Subroutines called

Subroutine: SKYMER
Function Solve a system of N-differential equation of the first order by the Rung-Kutta-Merson method of the 4th order with variable step (error estimated at each step). Derivatives are calculated by the call of subroutine DERIV
Entry variables X0: lower integrative boundary XF: upper integrative boundary N: number of differential equations ITAB: Number of values to save (i.e. number of call of SAVESIM subroutine)
Output variables
Entry-output variables Y0: vector of initial conditions (concentration) in X0 in input; Vector of initial conditions to continue the simulation from XF.
Called by CALCOL
Subroutines called SAVESIM

Subroutine: WEG

Function

WEGSTEIN procedure: iterative method to solve simultaneously a system of N-equations f(x)=x.

Entry variables

Y: vector of the N function f(x)
PRE: sensibility for the convergence test
Xmin: Vector of the minimum value for x
Xmax: Vector of the maximum value for x
NR: Number of equations

Output variables

NC: convergence indicator
SCRIT: Greatest criteria calculated
NCRIT: Index of the equation for which was calculated the greatest criteria
INDIC: Number of value which have reach convergence

Entry-output variables

X: vector of the x values
XA: temporary working vector for x values
YA: temporary working vector for f(x) values

Called by

CALCOL

Subroutines called

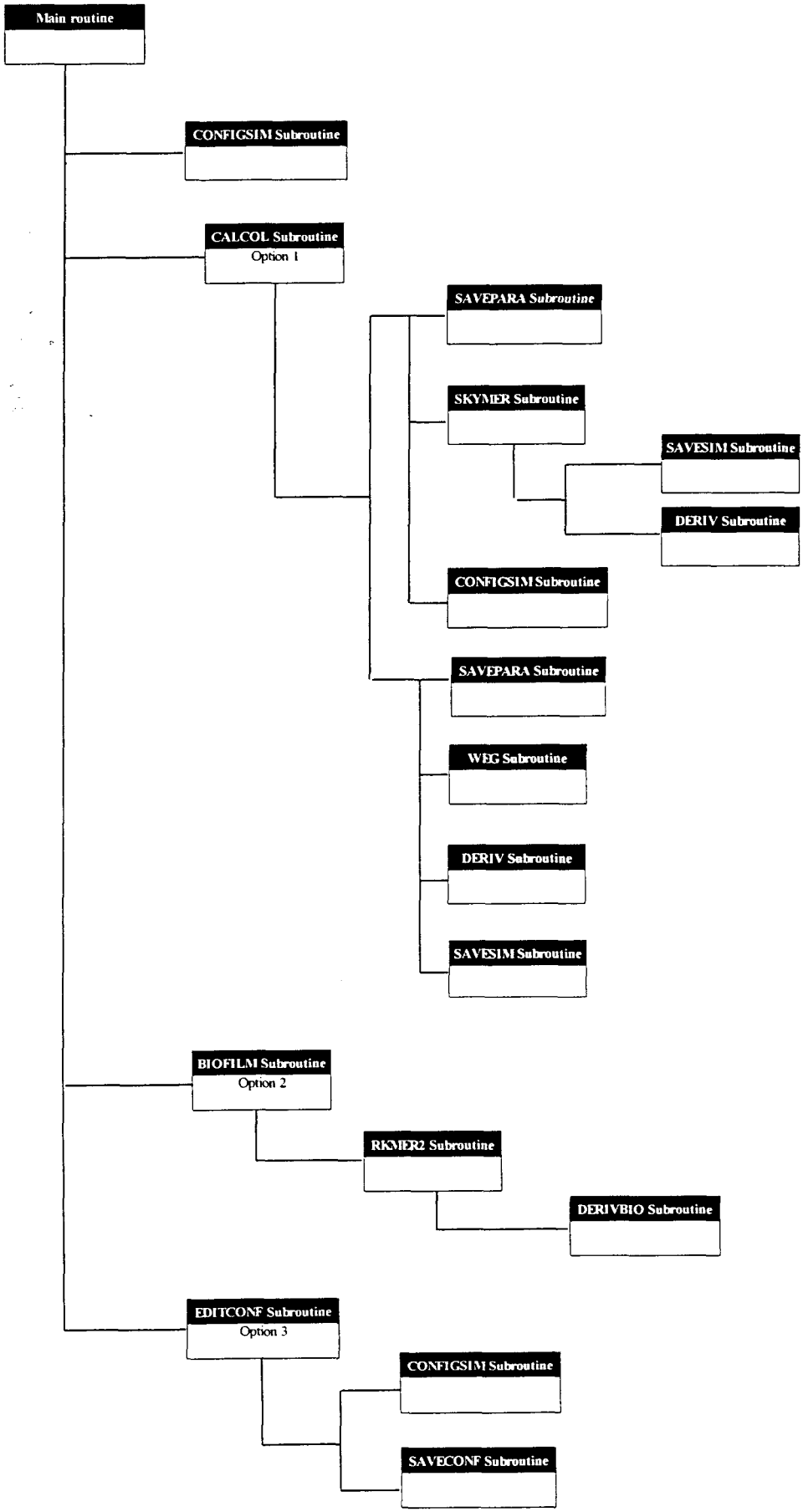
None

Subroutine: DERIV
Function Subroutine containing the integro-differential model for the dynamic simulation of the nitrifying column (without the biofilm diffusion term).
Entry variables X: time of current simulation
Output variables FCT: vector of derivatives of the system
Entry-output variables Y: vector of concentrations in every segment of the column
Called by SKYMER
Subroutines called None

Subroutine: DERIVBIO
Function Subroutine containing the integro-differential model for the biofilm diffusion model. Integration on the biofilm depth
Entry variables X: Biofilm depth for the current simulation
Output variables FCT: vector of derivatives of the system
Entry-output variables Y: vector of concentrations in the biofilm
Called by RKMER2
Subroutines called None

Subroutine: RKMER2
Function Solve a system of N-differential equations of the first order by the Rung-Kutta-Merson method of the 4th order with variable step (error estimated at each step). Derivaties are calculated by the call of subroutine DERIVBIO
Entry variables X0: upper integrative boundary (thickness of biofilm) XF: lower integrative boundary (0.) N: number of differential equations ITAB: Number of values to save Y0: vector containing the initial concentrations at X0.
Output variables XTAB: vector in which are stored the ITAB results of concentrations. YTAB: vector in which are stored the ITAB results for derivaties.
Entry-output variables
Called by BIOFILM
Subroutines called DERIVBIO

II.2.2 - Subroutines hierarchy



II.2.3 - Subroutines listing

Report to appendix 1

II.3 - Important remarks

II.3.1 - Differences between DOS and UNIX versions.

The only difference between the DOS and the UNIX version of the program concerns the screen display. The graphical characters used under the DOS operating system do not exist under the UNIX operating system. Then, these characters have been changed. The problems concern the "FORMAT()" descriptors which are different between in the FORTRAN language under DOS and UNIX:

the continuation line descriptor, allowing to write at the end of a previouly written line is "\" under DOS and "$" under UNIX system.

the descriptor allowing to stay on the same line and to replace a message by a new one is "+" under DOS and doesn't exists under UNIX. When running under UNIX, all new message are displayed on a new line on the screen. For this reason, the display of the advancement of a simulation (for dynamic simulation and biofilm diffusion simulation) is different in the DOS and in the UNIX version (report to the Users manual).

II.3.2 - Common variables

Most of the variables used in the different subroutines are shared by using a "COMMON" statement. All these "COMMON are described in the Main routine:

COMMON/REACBIO2/MUMAX,KS,KI,STO
Common for the biological kinetic parameters

COMMON/PHYINI/T,P,PHINI
Common for physical parameters (temperature, pressure, regulated pH)

COMMON/PHYPH/KA
Common for the pH equilibrium constants

COMMON/PHYPH2/PHA,PHC,PHB
Common for the real pH in each segment of the column (not used, the pH is assumed to be regulated at PHINI)

COMMON/PHYTRANS/D,KLBIO,KLGAZ,ASPGAZ,ASPBIO
Common for liquid-biofilm and gas-liquid transfert

COMMON/PHYTRANS2/CSAT

Common for the gas-liquid equilibrium constant

COMMON/BILLE/RO,HBIO,RNBIO,BWO
common for the beads description

COMMON/COLON/EPSL,EPSG,EPS,FINP,GINP,RL,RG,VA,VB,VC,FBAK,F
BAKPRIM
COMMON/COLON2/REHL,REHG,LONG,DCOL,SECT,DELTAP,DH
Commons for the hydrodynamic design of the column

COMMON/TAB/N,CORP,TMPS,REAC
Common for the definition of the dimension of vectors

COMMON/TAB2/NOM$
Common containing the names of the compounds involved

COMMON/CINI/CINL,CING
Common containing the inlet concentration on the column

COMMON/RKMERY/XIM1,YIM1
Common specific of the RKMER2 and SKYMER subroutines

III - NitriSim v2.3 User manual

III.1- Install procedure

The NitriSim v2.3 install patch (floppy disk) is composed of the following files:

NITRISIM.EXE : Binary executable

In the DATA directory :

PHYTRANS.DAT : Default physical parameters for the compounds
PHYPH.DAT : Data file containing constant for pH calculation
CORPS.DAT : Data file containing the compounds name
CINET.DAT : Default kinetic parameters
STOIC.DAT : Data file containing stoichiometric coefficients
FLOWCOL.DAT : Default column design (flow parameters)
CARCOL.DAT : Default column design

In the SOURCES directory

MAIN23.FOR
CALCOL23.FOR
FILM23.FOR
EDCONF23.FOR
CONFSI23.FOR
SAVECF23.FOR
SAVPAR23.FOR
SAVSIM23.FOR
SKYMER23.FOR
WEG23.FOR

To install the NitriSim v2.3 program, copy in the **same directory** the binary NITRISIM and the .DAT files, or better, run the install procedure (reports to INSTALL.TXT or appendix 3).

Important notes:

Read Install.txt notes on the floppy disk for installation of the program.
The column design, the biological and physical parameters are stored in the data files of the same directory of the binary NitriSim program. If the user modified some of these parameters, they are automatically stored as new default values in these files. To avoid the loss of the values previously stored as default for the NitriSim v2.3 software (reported in the different tables of this TN), it is suggested that in the NitriSim directory you create a Data_v2_3 directory containing the data files released with the software. This operation is not required if you use the automatic install procedure under DOS operating system (cf INSTALL.TXT notes)

To run the NitriSim program

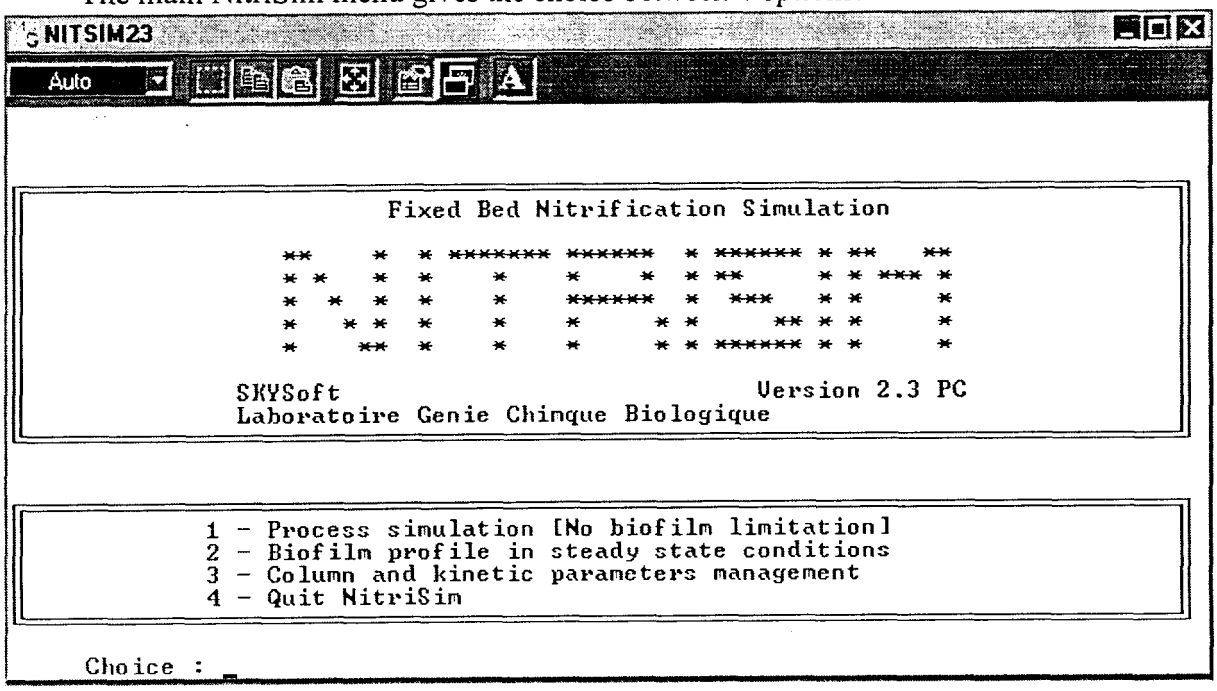
- Type "NITRISIM" on the NitriSim directory command line (DOS or UNIX operating system)

or

- Double click on the NITRISIM program icon (Windows95 or X-Windows)

III.2 - Main menu

The main NitriSim menu gives the choice between 4 options:



Option 1: Run simulations of the model developed in TN 23.1, 23.2 and 23.3 without the biofilm diffusion limitation term.

Option 2: Run simulation of the biofilm profile. This is not a complete dynamic simulation. It gives only the concentration profile inside the biofilm for a defined liquid and gas concentration profile inside the column.

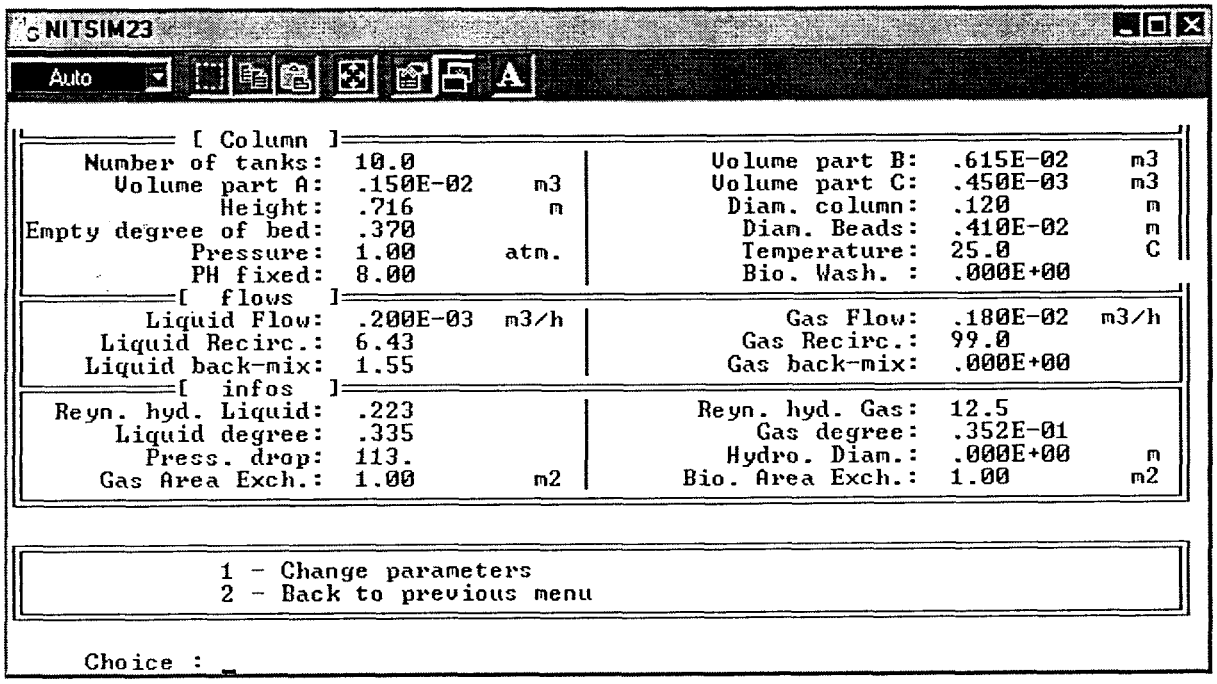
Option 3: List all the parameters of the models and allow to change the default values stored in data files.

Option 4: Quit NitriSim

III.3 - Model parameters management Menu

Most of the model parameters are loaded in the program associated data files.

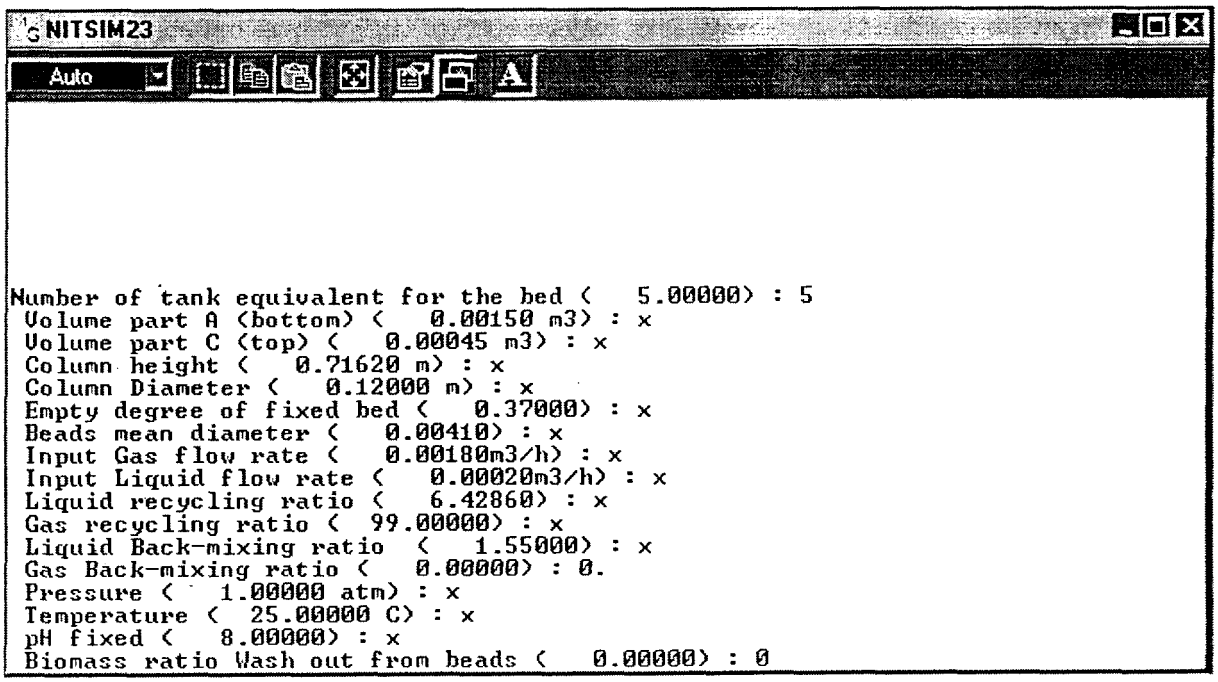
III.3.1- Option 1: Design of the column



The two first parts displayed (Column and flow) can be modified using the option 1.

The "infos" part displays results of calculations in the CONFSIM subroutine and can not be modified by using the program interface. These informations are only indicative values. The hydraulic Reynolds number for gas and liquid where calculated using a formula for two phases systems, and are probably not adapted to the three phase process of the nitrifying column. The voidage for gas and liquid (respectively called gas and liquid void degree) are those determined by UAB experiments. Pressure drop is calculated by the ERGUN relation. Hydraulic diameter is not calculated. The exchange area are not calculated and set to 1. The gas-liquid and liquid biofilm transfer coefficients must be directly given as Kla .

By choosing the option 1, you are asked to modify all the parameters listed in "column" and "flows".



When you answer with a number, the new value is recorded and stored as a new default value in the data files.

When you answer using another character than a number, the default value (displayed in bracket) is conserved.

When changes are terminated, the window "Column parameters" is displayed again, and you can verify your modifications.

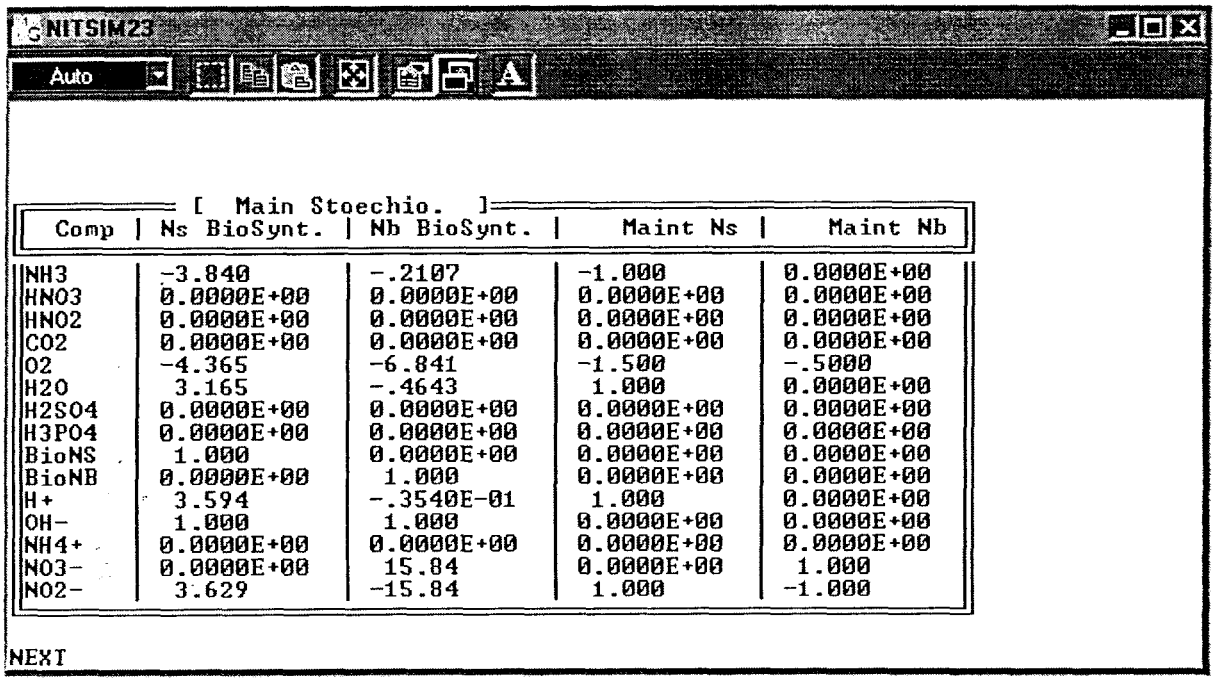
III.3.2 - Option 2: stoichiometric equations

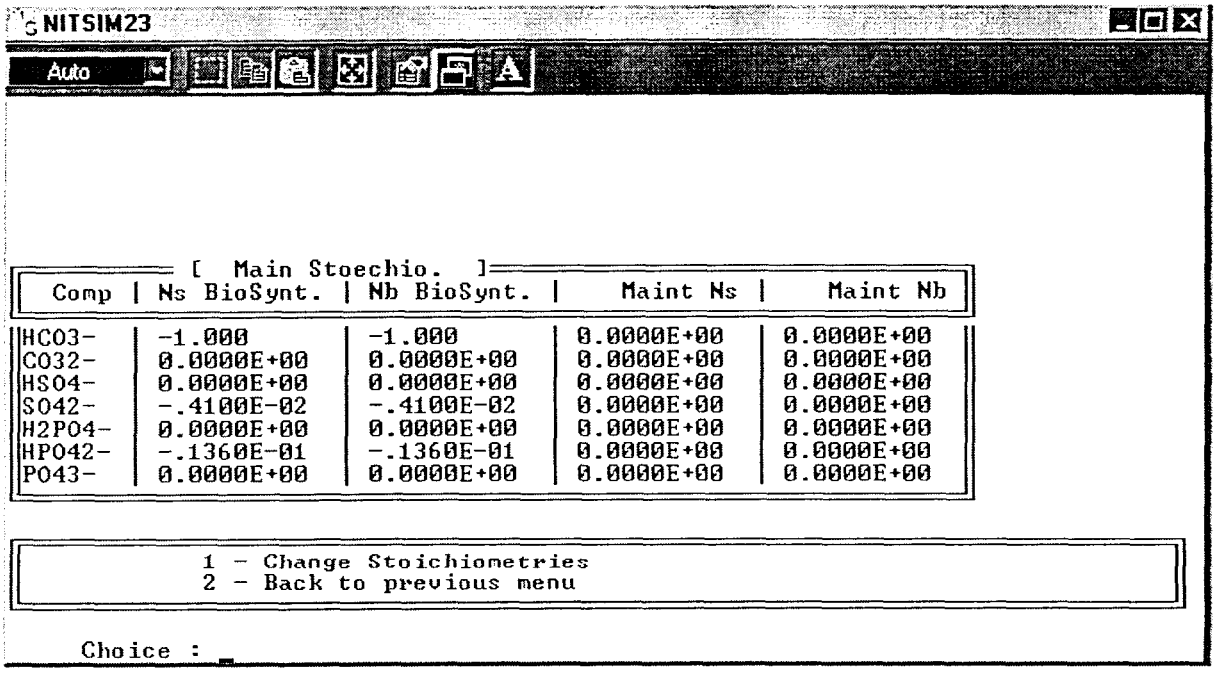
Note: If you modify the Biomass composition you must change the value of PMBIO (C-molar mass of the biomass in g) in the subroutines DERIV and BIOFILM.

In this version, only 2 stoichiometric equations for each micro-organism (Ns. Europea and Nb. Winogradskyi) are available. One for the biosynthesis and one for the maintenance.

Before any modifications, beware of the kinetics expressions of the biological model, which use the stoichiometric coefficients, especially for maintenance (TN 23.1 and 23.2). The substrate for maintenance is **fixed** in the subroutine DERIV as NH_3 for *Nitrosomonas* and NO_2^- for *Nitrobacter*. If you need to change them, you must modify the model expression for biological kinetics in subroutine DERIV.

As in all parameters listing, first are displayed the values of stoichiometric coefficients for *Nitrosomonas europea* (Ns) and *Nitrobacter winograskyi* (Nb). The stoichiometries used are reported in TN 32.1.





If you choose to modify a stoichiometric equation, it is asked which equation among the 4 you want to change.

1→ Nitrosomonas biosynthesis equation

2→ Nitrobacter biosynthesis equation

3→ Nitrosomonas maintenance equation

4→ Nitrobacter maintenance equation

The previous coefficient is given in bracket. If you want to keep this value, type a letter instead of a number.

III.3.3- Option 3: Biological kinetics

You have access to the three main classes of biological kinetic parameters:
1→ Maximum growth (μ_{max}) rate and maintenance coefficients
2→ Saturation constants
3→ Inhibitory constants

All the default parameters used are taken from the literature (TN 23.1, 23.2 and 23.3).
The parameters for each class can be displayed and modified. As in previous listing, when the change option is chosen, by typing a letter instead of a number, the previous default value (in bracket) is conserved. All changes are recorded in data files.

For saturation constants, a value of 0 mol/l indicates there is no saturation constant. For inhibitory constant, a value of 10^{-10} mol/l indicates there is no inhibition.

III.3.4- Option 4: Compounds involved and their physico-chemical associated constants

This menu allows to add new compounds and to modify the constants associated to the compounds involved in the process. All compounds involved (ionic and non ionic) are listed.

The number of compounds is limited to 30. You don't have the right to modify the 22 first compounds (the order in which the compounds are used is case sensitive for the kinetic model, this remark is the same as for the modification of the stoichiometric equations). If by adding new compounds the kinetics laws are modified, you will have to change the DERIV subroutine.

The constants associated to the compounds are:

- their acid-base equilibrium constant (calculation of pH equilibrium)

- their saturation constant (for the gases in the liquid phase) in mol/l

- their Kla value (in h^{-1}) for gas-liquid transfer. The real Kla value must be given and not only the Kl which correspond to the transfer resistance. A model for the calculation of Kla from Kl and the exchange surface between the gas and liquid phases has not yet be included in the software.

- their Kla value for liquid-biofilm transfer. Set all values to zero. The transfer resistance has not yet been included in the biofilm diffusion model (TN23.3).

- their diffusion constant in the biofilm (in $m^2.s^{-1}$). If their is no diffusion limitation, the constant is set to 10^{10}.

NITSIM23

Auto

[G-L Constants]

Comp.	Eq. pH	[C]. Satur.	Kl gaz	Kl biofilm	Diffusion
NH3	0.1762E-04	0.1173E-01	0.0000E+00	0.0000E+00	0.1000E+11
HNO3	0.1000E+06	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
HNO2	0.3980E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
CO2	0.4320E-06	1635.	51.00	0.0000E+00	0.1000E+11
O2	0.0000E+00	0.4272E+05	51.00	0.0000E+00	0.2050E-08
H2O	0.1000E-13	0.3100E-01	500.0	0.0000E+00	0.1000E+11
H2SO4	0.1000E+06	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
H3PO4	0.6918E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
BioNS	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
BioNB	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
H+	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
OH-	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
NH4+	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
NO3-	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
NO2-	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11

NEXT

NITSIM23

Auto

[G-L Constants]

Corps	Eq. pH	[C]. Satur.	Kl gaz	Kl biofilm	Diffusion
HCO3-	0.4557E-10	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
CO32-	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
HSO4-	0.1047E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
SO42-	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
H2PO4-	0.6166E-07	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
HPO42-	0.4780E-12	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11
PO43-	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E+11

1 - Change Compounds [warning]
2 - pH Equilibria Constants
3 - Partition G/L constant
4 - G/L Exchange coefficient
5 - L-Biofilm Exchange coefficient
6 - Biofilm diffusion coefficient
7 - Back to previous menu

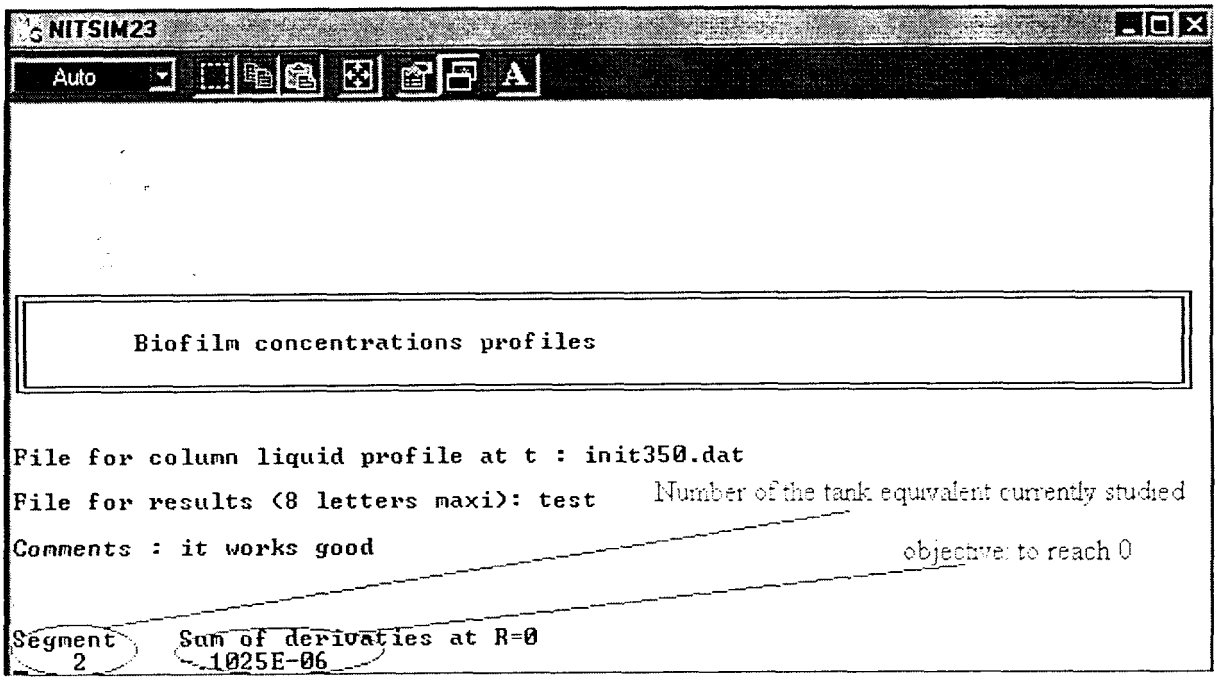
Choice :

Using the options 1 to 6 you can modify the value displayed in the table. All changes are recorded as a new default value in the data files. The previous coefficient is given in bracket. If you want to keep this value, type a letter instead of a number.

III.4 - Biofilm diffusion simulation

This is not a complete dynamic simulation. It gives only the concentration profile inside the biofilm for a defined liquid and gas concentration profile inside the column.

It is asked to you which file contains the concentration profile inside the column (this file must exist). Such a file must be created. It can be made by copying from a simulation result file (extension ".RES") the concentrations recorded at a time t for the fixed bed only (appendix 2-A) to a new file, in the same format. The first line of the file must indicate the number of tanks equivalent for the fixed bed (appendix 2-B).



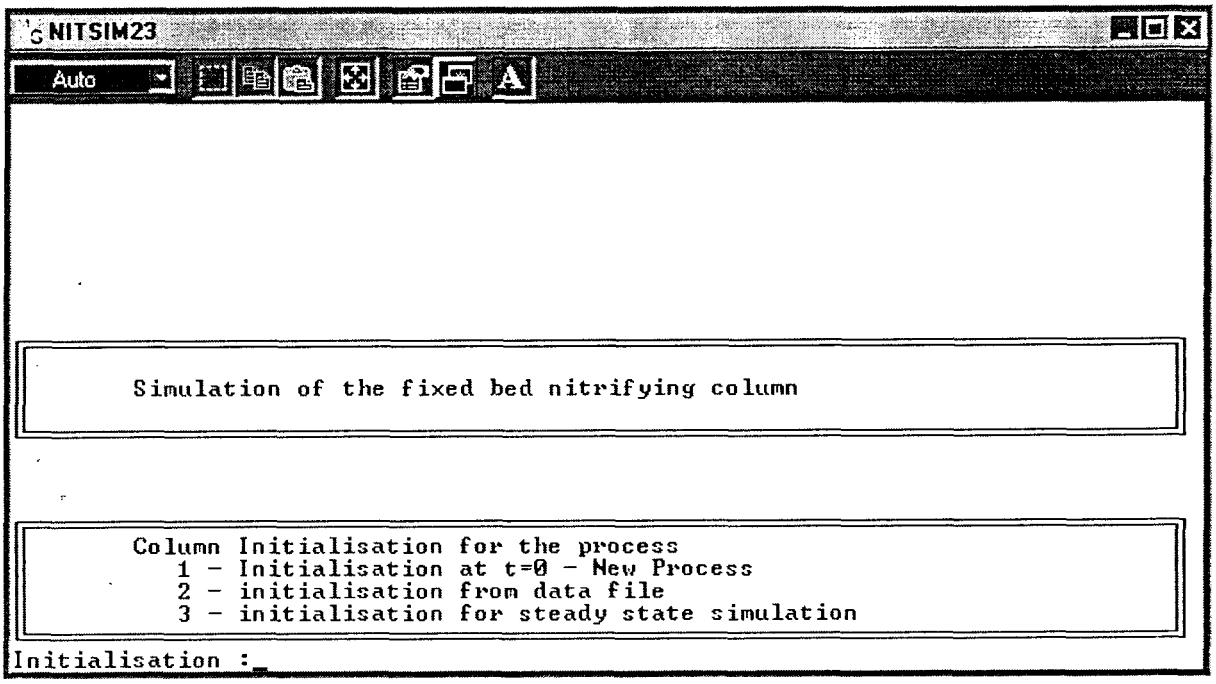
The advancement of the simulation is displayed. In the above example, "Segment" indicates that the second tank equivalent for the fixed bed (from the bottom of the column) is simulated. The algorithm is presented in TN 23.3. It is an iterative algorithm, the objective of which is to reach one of the boundaries condition of the biofilm diffusion model: all derivatives at the surface of the beads are null.

The results of the simulation are stored in the user specified file: TEST.BFM. It gives concentrations of compounds for several biofilm depth, for each section of the fixed bed.

III.5 - Nitrifying process simulation Menu

This last option of the Main Menu, drives to the dynamic simulation (without biofilm diffusion) of the fixed bed nitrifying process.

The simulation can be made to simulate a transient process, from time t_0 to time t_1, (option 1 and 2) or to simulate the column in steady-state (option 3).



III.5.1 - Complete process (option 1 and 2)

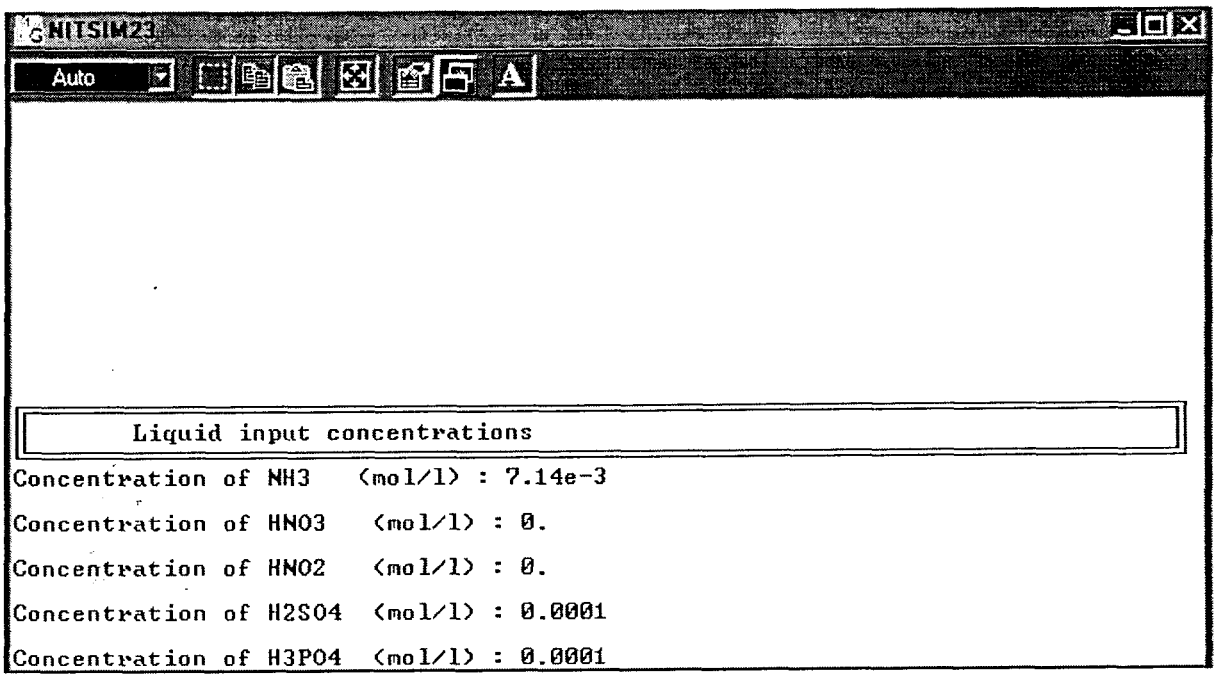
You have to choose between an initialisation with an unknown (option 1) and a known (option 2, if it exists a file describing the concentration profile inside the column) column concentration profile.

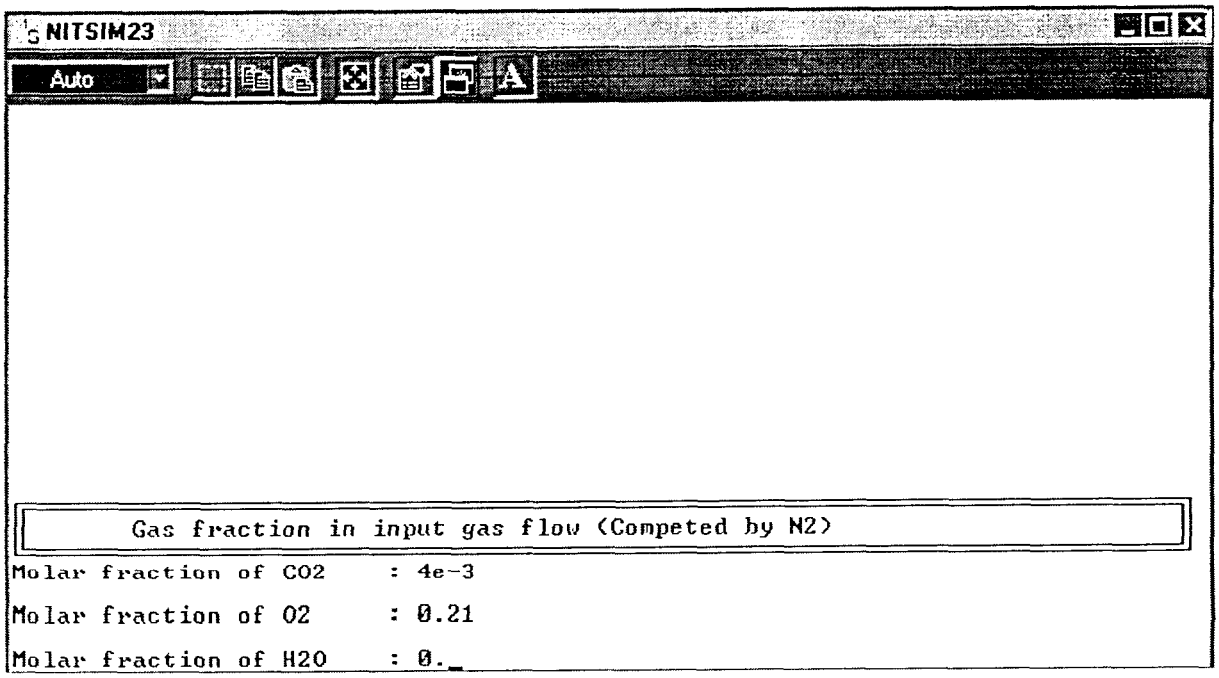
For an unknown column profile at the beginning of the simulation (e.g. New process - t=0), all concentrations inside the column are set to the input concentrations. The fixed biomass is supposed to be uniformly distributed. The biomass concentration, asked to the user (in g/l), represents the mean concentration in each tank equivalent of the column and is supposed to be composed of 70% of Nitrosomonas.

For a known column concentration profile (file initialisation), the file must give the concentrations for all compounds in every element of the column (bottom, each tank of the fixed bed, top of the column). An example of a file for the initialisation of a dynamic simulation is given in appendix 2-C. Such a file can be made by using a previous result file (extension .RES), as for the file needed for the initialisation of biofilm transfer limitation profile.

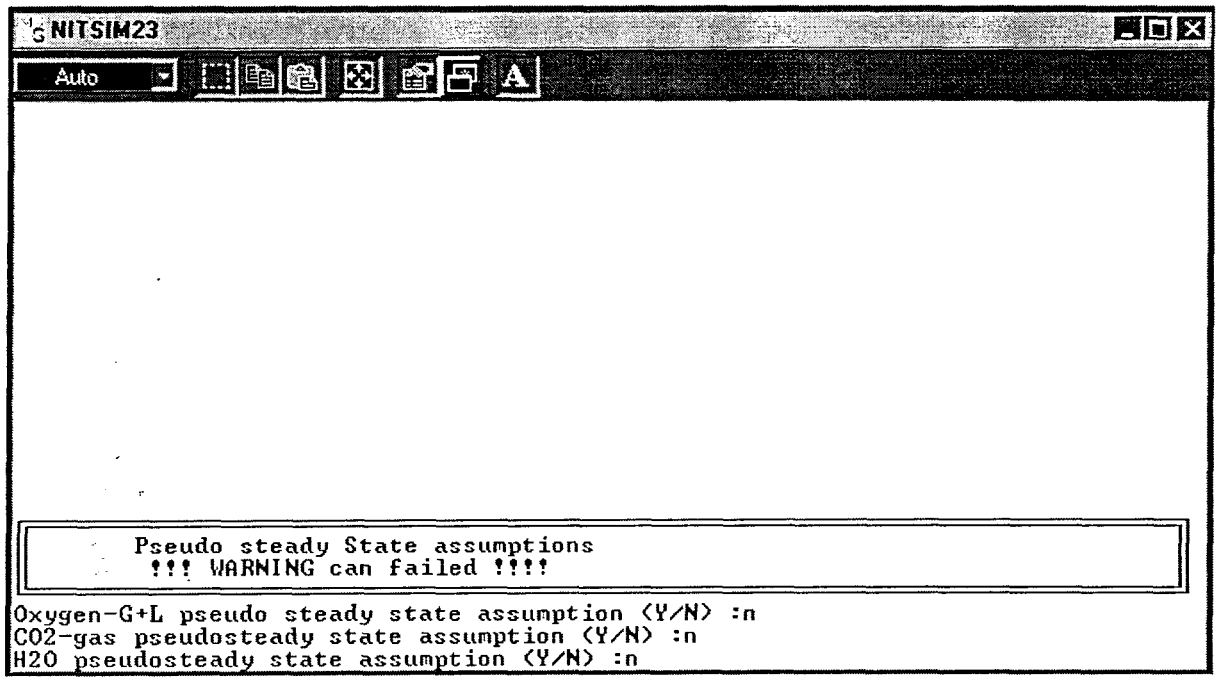
The entries on the column must be defined:

- the liquid concentrations of substrates (in mol/l) under their non-dissociated form. If NH_3, H_2SO_4 (sulphur source) or H_3PO_4 (phosphate source) are set to 0, this will result in the absence of growth (limitation).
- the gas fraction of water (vapour), CO_2 and O_2. Their sum does not exceed 1.

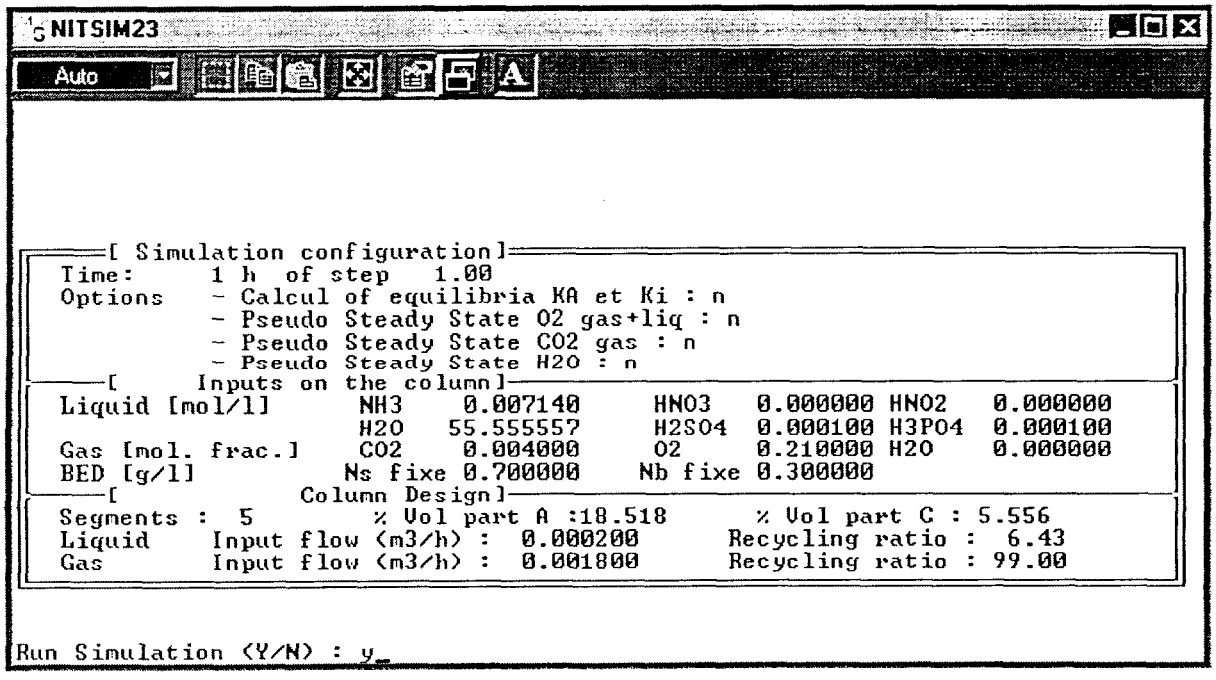




An option for the dynamic simulation is available namely the assumption of pseudo-steady-state for O_2, CO_2 and water in the gas phase. The objective was to decrease the computational time by avoiding the low integration step imposed by the dynamic of these compounds. In fact, no processes simulated at the present time have been successfully simulated with this option. When the simulation did not fail, the computational time was not reduced, and sometimes was increased. It is strongly recommended to try the pseudo-steady-state (answer "Y") only if a normal simulation fails or is "time blocked" because of a very low time integration step.



Before the simulation a window gives a summary of the main parameters used for the current simulation, including the column design, the pseudo-steady-state options and the entries definition. By accepting the configuration, you launch the simulation.

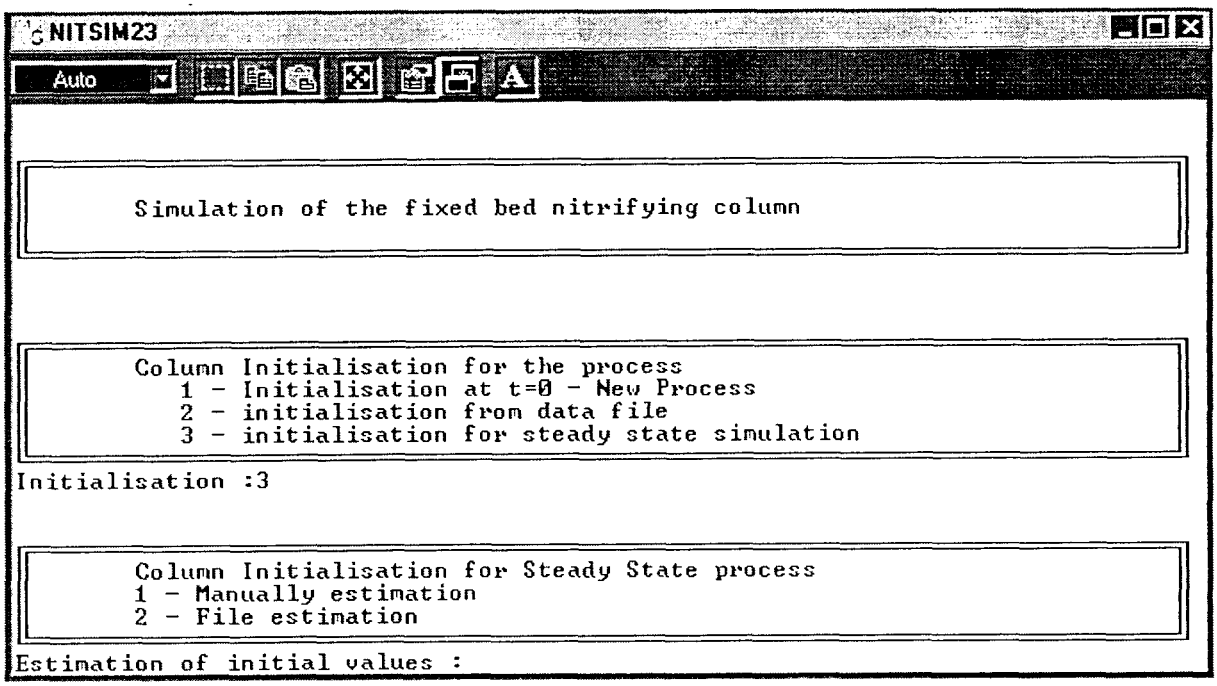


During the simulation, the advancement of the simulation is given by the percentage at the bottom of the screen. On a Pentium 100 based PC, the computation time can be estimated, for a column with 5-tanks equivalent for the fixed bed, to 40-50 minutes for a process of 100 hours.

When the current simulation is ended, you can continue the same and modify the flow rates and the input concentration and gas fraction. This possibility was used in TN23.3 for the simulation of the transient behaviours.

III.5.2 - Steady state process

As in the previous menu, 2 options are available for steady -state simulations. The definition of initial value (as close as possible of the steady-state conditions) can be given by using a file (option2) or by using a "manually" estimation included in the program. For a file estimation, the file used for the initialisation of a dynamic simulation can be used.



Then, in the same form as for dynamic simulations, the entries (liquid concentrations and gas fractions) must be given. The simulation configuration windows is displayed for confirming, then the name of the result data file (8 letters maxi) and the optional comments.

Two parameters related to the steady-state algorithm are required:

- **The Wegstein sensitivity parameter**. By default it is fixed to 10^{-3} (e.g. liquid and gas concentrations results at steady-state, when the convergence is reached, are given at 0.1%.).

- **The biomass sensitivity** (Sensi_Bio). The biomass (X) is supposed in steady state if

$$dX/dt \,/X < 100 \,.\, Sensi\_Bio$$

or $dX/dt < Sensi\_Bio$

It can be noted that by decreasing the sensibility for the biomass 10^{-6} by default), the convergence can be reached more quickly, but the results can be slightly different.

During the simulation, several informations are displayed on the screen:

A→ Number of the iterative loop for biomass reinitialisation - limited to 20-

B→ Number of the iterative loop for the Wegstein routine - limited to 25000 -

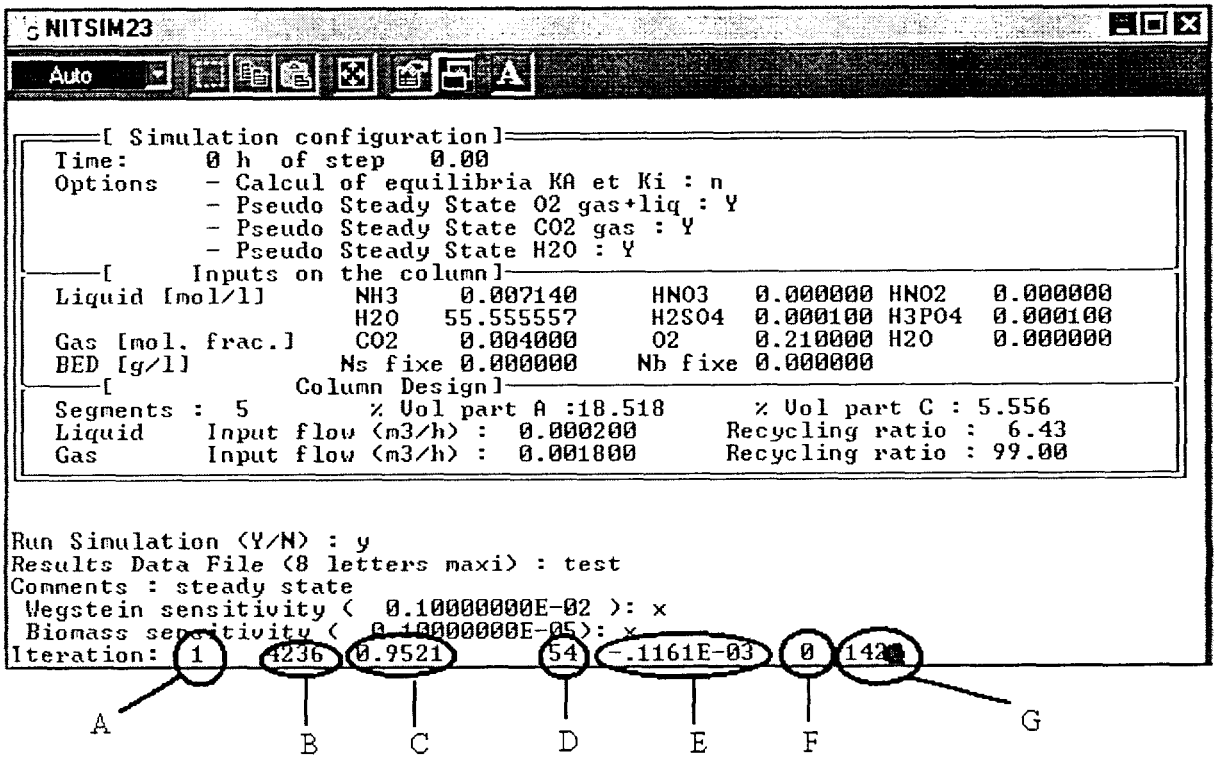
C→ Value of the greatest criteria value for a compound in the Wegstein procedure

D→ Index of the compound related to the value given at C

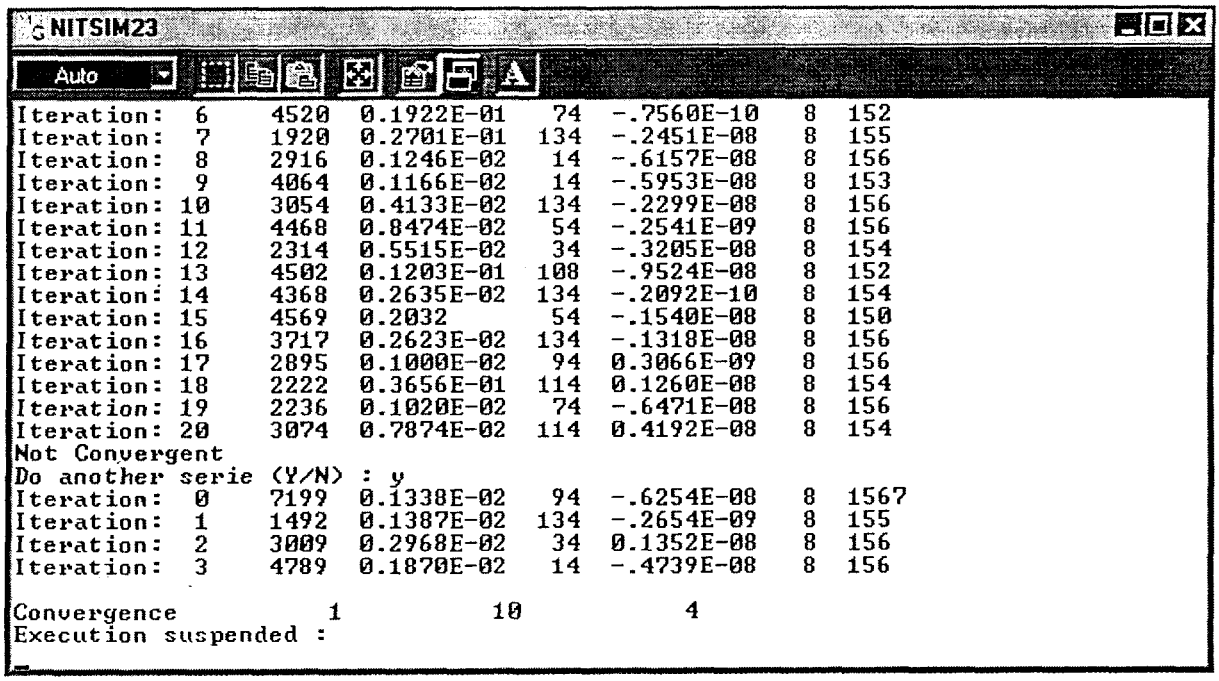
E→ Time derivative of this compound

F→ Number of biomass assumed in steady state. The convergence is reached if in each equivalent tank of the fixed, both Nitrosomonas and Nitrobacter are in steady-state (e.g., if there is N tank, the value researched is 2.N)

G→ Number of compound in the Wegstein procedure which have reach steady-state. The convergence is reached at: [N+2].20 + 3.N +2



After a maximum of 20 iterations, if the convergence, both for the Wegstein procedure and for the biomass is not reached the program stops. The user can decide to continue if he thinks that the convergence can be reached with some other iterations.



III.6 - Simulations and Results storage files

III.6.1 - Names

Files for initialisation

You can give any name to these file. When such a file is asked, you must enter the complete name (including its extension):

TOTO.TOT

Result file of Biofilm diffusion simulation

A name is asked for each new simulation. You have to enter a name of 8 letters. An extension "**.BFM**" is added to identify the file created for a biofilm simulation.

Result file for dynamic simulations

A name is asked for each new simulation. You have to enter a name of 8 letters. An extension "**.CNF**" is added to identify the file created containing the characteristics of the process simulated (column design and flow rates). An extension "**.FIN**" is added to identify the file created containing the output concentrations and gas fractions, as well as the mean biomass inside the total fixed bed. An extension "**.RES**" is added to identify the file created containing the complete results of the process simulated (profile inside the column).

III.6.2 - Examples

Files for initialisation

An example of a file required for the simulation of the biofilm transfer limitation is given in appendix 2-B. It is composed on the first line of the number of tank equivalent for the

fixed bed, and on the five last line of the concentration or gas fraction for each compound. The order in which the compounds are listed is the same as this reported in Appendix 2-A.

An example of a file required for the dynamic simulations is given in appendix 2-C. It is composed of N+2 lines (N is the number of tanks equivalent for the fixed bed). The order of the line is from the bottom of the column to the top. The order in which the compounds are listed is the same as this reported in Appendix 2-A. It is then easy to use a previous result file to initiate a new simulation.

Result file of dynamic simulation

An example of a .RES file for is given in appendix 2-A
An example of a .FIN file for is given in appendix 2-D
An example of a .CNF file for is given in appendix 2-E
These results were obtained for a simulation of 1500 hours for a process defined by the flow rate reported in section 1 (standard configuration) and 5-tanks equivalents for the fixed bed. The number of time values saved was 3.

Result file of steady state simulation

An example of a .RES file for the steady state is given in appendix 2-F. This was obtained with a "manual estimation of initial conditions. This represents the steady state for a 5-tanks equivalent representation of the fixed bed in a standard process.

APPENDIX 1

Listing of the NitriSim Programs

```
      SUBROUTINE CALCCOL()
C////////////////////////////////////////////////////////////////////////
C/                                                                      /
C/        CACULATION OF THE DYNAMIC BEHAVIOUR OF A FIXED BED            /
C/        NIRIFYING COLUMN                                              /
C/        MODEL BASED ON TN 27.1 27.2 27.3 and 32.1                     /
C/                              NITRISIM                                /
C/ V 2.3                                                 UPDATE 10/97   /
C////////////////////////////////////////////////////////////////////////

C////////////////////////////////////////////////////////////////////////
C/                            COMMENTS                                  /
C/                                                                      /
C/   TMPSFIN: Lenght of the process simulation (hours)                  /
C/   TMPS    : Number of values saved in results files                  /
C/   TMPSINI: Time of begining simulation (hours)                       /
C/   INI$    : Name of file (+extension) for initialisation             /
C/   RES$    : Name (No extension) for results                          /
C/                                                                      /
C////////////////////////////////////////////////////////////////////////

C-----------------------------------------------------------------------
C                   DECLARATION
C-----------------------------------------------------------------------

      IMPLICIT REAL*8 (A-H,O-Z)

      INTEGER NMAX,CORPMAX,TMPSMAX,RMAX,INIPS
      PARAMETER(NMAX=20)
      PARAMETER(CORPMAX=30)
      PARAMETER(TMPSMAX=300)
      PARAMETER(RMAX=4)

      CHARACTER*6 NOM$(CORPMAX)
      CHARACTER*1 CHOIX$,CHOIXO2$,CHOIXCO2$,CHOIXH2O$
      CHARACTER*12 INI$,RES1$,RES2$,RES3$
      CHARACTER*30 DESC$

      INTEGER N,CORP,TMPS,REAC
      INTEGER TMPSFIN
      INTEGER PSTO2,PSTCO2,PSTH2O,ini
      INTEGER CHOIX,MENU

      REAL*8 TEMPO,TEMPO2
      REAL*8 REHL,REHG,EPSL,EPSG,EPS,F,G,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      REAL*8 LONG,SECT,DELTAP,DCOL,DH
      REAL*8 T,P,PHINI,PHA,PHC
      REAL*8 PHB(NMAX)
      REAL*8 MUMAX,KI,KS,KA,KLGAZ,KLBIO
      REAL*8 RO
      REAL*8 ASPGAZ,ASPBIO


      DIMENSION HBIO(NMAX),RNBIO(NMAX)
      DIMENSION CSAT(CORPMAX)
      DIMENSION CINL(CORPMAX),CING(CORPMAX)
      DIMENSION Y0(650),FY0(650),XMAX(650),XMIN(650),XAT(650),
     &YAT(650)
      DIMENSION Y0INI(650)
      DIMENSION BIONS_TMP(NMAX),BIONB_TMP(NMAX)
      DIMENSION D(CORPMAX),KLBIO(CORPMAX),KLGAZ(CORPMAX)
      DIMENSION KS(RMAX,CORPMAX),KI(RMAX,CORPMAX)
      DIMENSION MUMAX(RMAX)
      DIMENSION KA(CORPMAX)
      DIMENSION STO(RMAX,CORPMAX)


      COMMON/REACBIO2/MUMAX,KS,KI,STO
      COMMON/PHYINI/T,P,PHINI
      COMMON/PHYPH/KA
      COMMON/PHYPH2/PHA,PHC,PHB
      COMMON/PHYTRANS/D,KLBIO,KLGAZ,ASPGAZ,ASPBIO
      COMMON/PHYTRANS2/CSAT
      COMMON/BILLE/RO,HBIO,RNBIO,BWO
      COMMON/COLON/EPSL,EPSG,EPS,FINP,GINP,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      COMMON/COLON2/REHL,REHG,LONG,DCOL,SECT,DELTAP,DH
```

```
      COMMON/TAB/N,CORP,TMPS,REAC
      COMMON/TAB2/NOM$
      COMMON/CINI/CINL,CING
      COMMON/STEADY/PSTO2,PSTCO2,PSTH2O

C////////////////////////////////////////////////////////////////////////
C/                          DATA INITIALISATION                         /
C////////////////////////////////////////////////////////////////////////

      WRITE(*,1100)
      WRITE(*,1001)
      WRITE(*,1006)
      WRITE(*,1002)'Simulation of the fixed bed nitrifying column'
      WRITE(*,1006)
      WRITE(*,1003)
      PRINT*
      PRINT*


C---------------------------------------------------------------------
C-- INITIALISATION OF CONCENTRATIONS TO ZERO
C-
      DO 95 I=1,CORP
         CINL(I)=0.
         CING(I)=0
95    CONTINUE

C---------------------------------------------------------------------
C-- MEnu for choice of type of simulation and initialisation
C-

      PRINT*
      WRITE(*,1001)
      WRITE(*,1002)'Column Initialisation for the process'
      WRITE(*,1002)'   1 - Initialisation at t=0 - New Process'
      WRITE(*,1002)'   2 - initialisation from data file'
      WRITE(*,1002)'   3 - initialisation for steady state simulation'
      WRITE(*,1003)
      WRITE(*,1007)' Initialisation :'
      READ*,INI
      INIPSP=0



C---------------------------------------------------------------------
C-- INITIALISATION OF DATA FOR SKYMER
C-
      IF (INI.EQ.1.OR.INI.EQ.2) THEN
      PRINT*
      PRINT*
      WRITE(*,1007)' Lenght of process simulation (Hours): '
      READ*,TMPSFIN
99    WRITE(*,1007)'Number of time iteration -Maximun 300- : '
      READ*,TMPS
      IF (TMPS.LE.1.OR.TMPS.GT.300) THEN
         WRITE(*,1009)'BETWEEN 2 AND 300'
         GOTO 99
      ENDIF
      TMPSINI=0

      ELSE
         TMPSFIN=0.
         TMPSINI=0.
         CHOIXO2$='Y'
         CHOIXCO2$='Y'
         CHOIXH2O$='Y'
      ENDIF


      IF (INI.EQ.1) THEN
C---------------------------------------------------------------------
C--  INITIALISATION AT t=0  (New PROCESS)
C-
         PRINT*
```

```
          WRITE(*,1001)
          WRITE(*,1002)'Initial Biomass in the bed '
          WRITE(*,1003)
          WRITE(*,1007)' Total Fixed biomass (70% Ns; 30% Nb) g/l : '
          READ*,CIBIO


       ELSEIF (INI.EQ.2) THEN
C-------------------------------------------------------------------------
C-  FILE INITIALISATION
C-
          PRINT*
          WRITE(*,1008)'Data file for initialisation (NAME.EXT) : '
          READ(*,'(A)')INI$


        ELSEIF (INI.EQ.3) THEN
C-------------------------------------------------------------------------
C--  INITIALISATION FOR STEADY STATE
C-
       PRINT*
       WRITE(*,1001)
       WRITE(*,1002)'Column Initialisation for Steady State process'
       WRITE(*,1002)'1 - Manually estimation'
       WRITE(*,1002)'2 - File estimation'
       WRITE(*,1003)
       WRITE(*,1007)' Estimation of initial values :'
       READ*,INIPSP

       IF (INIPSP.EQ.2) THEN
          WRITE(*,1008)'Data file for initialisation (NAME.EXT) : '
          READ(*,'(A)')INI$
       ELSEIF (INIPSP.EQ.1) THEN
C
C            Manual Initialisation of steady state
C            is made in the stedy state calulation routine
C
       ELSE
          RETURN
       ENDIF

C-------------------------------------------------------------------------
C-- BAD VALUE FOR INITIALISATION
       ELSE
         return
       ENDIF




C-------------------------------------------------------------------------
C-- INPUT LIQUID CONCENTRATION  Valid for all types of processes
C-
C
       WRITE(*,1100)
       WRITE(*,1001)
       WRITE(*,1002)'Liquid input concentrations'
       WRITE(*,1003)

80     WRITE(*,1008)'Concentration of ',NOM$(1),'(mol/l) : '
         READ*,CINL(1)
         IF (CINL(1).LT.0) THEN
            WRITE(*,1009)'NEGATIVE'
            GOTO 80
         ENDIF
81     WRITE(*,1008)'Concentration of ',NOM$(2),' (mol/l) : '
         READ*,CINL(2)
         IF (CINL(2).LT.0) THEN
            WRITE(*,1009)'NEGATIVE'
            GOTO 81
         ENDIF
82     WRITE(*,1008)'Concentration of ',NOM$(3),' (mol/l) : '
         READ*,CINL(3)
         IF (CINL(3).LT.0) THEN
            WRITE(*,1009)'NEGATIVE'
```

```
          GOTO 82
        ENDIF

      CINL(6)=55.55555555

83    WRITE(*,1008)'Concentration of ',NOM$(7),' (mol/l) : '
        READ*,CINL(7)
        IF (CINL(7).LT.0) THEN
          WRITE(*,1009)'NEGATIVE'
          GOTO 83
        ENDIF
84    WRITE(*,1008)'Concentration of ',NOM$(8),' (mol/l) : '
        READ*,CINL(8)
        IF (CINL(8).LT.0) THEN
          WRITE(*,1009)'NEGATIVE'
          GOTO 84
        ENDIF



C-------------------------------------------------------------------------
C-- INPUT GAS FRACTION Valid for all types of processes
C-

85    FRAC_INDEX=0.
      WRITE(*,1100)
      WRITE(*,1001)
      WRITE(*,1002)'Gas fraction in input gas flow (Completed by N2)'
      WRITE(*,1003)

      WRITE(*,1008)'Molar fraction of ',NOM$(4),' : '
      READ*,CING(4)
      FRAC_INDEX=FRAC_INDEX+CING(4)
      CING(4)=CING(4)/(8.314*(273.15+T)/(P*1.013E2))

      WRITE(*,1008)'Molar fraction of ',NOM$(5),' : '
      READ*,CING(5)
      FRAC_INDEX=FRAC_INDEX+CING(5)
      CING(5)=CING(5)/(8.314*(273.15+T)/(P*1.013E2))

      WRITE(*,1008)'Molar fraction of ',NOM$(6),' : '
      READ*,CING(6)
      FRAC_INDEX=FRAC_INDEX+CING(6)
      CING(6)=CING(6)/(8.314*(273.15+T)/(P*1.013E2))

      IF (FRAC_INDEX.GT.1.) THEN
          PRINT*,'Total gas fraction greater than 1'
          pause
          GOTO 85
      ENDIF

C/////////////////////////////////////////////////////////////////////////
C/            PH EQUILIBRIA FUNCTION OF TEMPERATURE                      /
C/////////////////////////////////////////////////////////////////////////
C
C    NOT IMPLEMENTED IN THIS VERSION
C    Report to TN 17.1 and 23.1


C/////////////////////////////////////////////////////////////////////////
C/         GAS-LIQUID EQUILIBRIA FUNCTION OF P AND T                     /
C/////////////////////////////////////////////////////////////////////////
C
C   NOT IMPLEMENTED IN THIS VERSION
C   Report to TN 17.1 and 23.1



C/////////////////////////////////////////////////////////////////////////
C/     Steady state hypothesis for a complete simulation process        /
C/                    (with INI=1 and INI=2)                            /
C/     Warning often failed                                             /
C/////////////////////////////////////////////////////////////////////////

      IF (INI.NE.3) THEN
```

```
97    WRITE(*,1100)
      WRITE(*,1001)
      WRITE(*,1002)'Pseudo steady State assumptions '
      WRITE(*,1002)' !!! WARNING can failed !!!!'
      WRITE(*,1003)
      WRITE(*,1007)' Oxygen-G+L pseudo steady state assumption (Y/N) :'
      READ(*,'(A)')CHOIXO2$
      IF (CHOIXO2$.EQ.'Y'.OR.CHOIXO2$.EQ.'y') THEN
          PSTO2=1
      ELSEIF (CHOIXO2$.EQ.'N'.OR.CHOIXO2$.EQ.'n') THEN
          PSTO2=0
      ELSE
          GOTO 97
      ENDIF
98    WRITE(*,1007)'CO2-gas pseudosteady state assumption (Y/N) :'
      READ(*,'(A)')CHOIXCO2$
      IF (CHOIXCO2$.EQ.'Y'.OR.CHOIXCO2$.EQ.'y') THEN
          PSTCO2=1
      ELSEIF (CHOIXCO2$.EQ.'N'.OR.CHOIXCO2$.EQ.'n') THEN
          PSTCO2=0
      ELSE
          GOTO 98
      ENDIF
94    WRITE(*,1007)'H2O pseudosteady state assumption (Y/N) :'
      READ(*,'(A)')CHOIXH2O$
      IF (CHOIXH2O$.EQ.'Y'.OR.CHOIXH2O$.EQ.'y') THEN
          PSTH2O=1
      ELSEIF (CHOIXH2O$.EQ.'N'.OR.CHOIXH2O$.EQ.'n') THEN
          PSTH2O=0
      ELSE
          GOTO 94
      ENDIF

      ENDIF



C////////////////////////////////////////////////////////////////////////
C/              CONFIRM CONFIGURATION FOR SIMULATION                    /
C////////////////////////////////////////////////////////////////////////

100   WRITE(*,1100)
      WRITE(*,1030)'Simulation configuration'
      TMPO=float(TMPSFIN)/FLOAT(TMPS-1)
      WRITE(*,1031)'Time: ',TMPSFIN,' h  of step ',TMPO
      WRITE(*,1032)'Options ','- Calcul of equilibria KA et Ki : ',
     +'n'
      WRITE(*,1032)' ','- Pseudo Steady State O2 gas+liq : ',CHOIXO2$
      WRITE(*,1032)' ','- Pseudo Steady State CO2 gas : ',CHOIXCO2$
      WRITE(*,1032)' ','- Pseudo Steady State H2O : ',CHOIXH2O$
      WRITE(*,1033)'Inputs on the column'
      WRITE(*,1034)'Liquid [mol/l]',NOM$(1),CINL(1),NOM$(2),CINL(2),
     +NOM$(3),CINL(3)
      WRITE(*,1034)' ',NOM$(6),CINL(6),NOM$(7),CINL(7),
     +NOM$(8),CINL(8)
      WRITE(*,1034)'Gas [mol. frac.]',NOM$(4),
     +CING(4)*(8.314*(273.15+T)/(P*101.3)),NOM$(5),
     +CING(5)*(8.314*(273.15+T)/(P*101.3)),
     +NOM$(6),CING(6)*(8.314*(273.15+T)/(P*101.3))
      WRITE(*,1037)'BED [g/l]','Ns fixed',CIBIO*0.7,'Nb fixed',CIBIO*0.3
      WRITE(*,1033)'Column Design'
      WRITE(*,1035)'Segments :',N,'% Vol part A :',VA/(VA+VB+VC)*100,
     +'% Vol part C :',VC/(VA+VB+VC)*100
      WRITE(*,1036)'Liquid ','Input flow (m3/h) :',fINP,
     +'Recycling ratio :',RL
      WRITE(*,1036)'Gas ','Input flow (m3/h) :',GINP,
     +'Recycling ratio :',RG
      WRITE(*,1003)


      PRINT*
      PRINT*
102   WRITE(*,1007)' Run Simulation (Y/N) : '
      READ(*,'(A)')CHOIX$
```

```
C-----------------------------------------------------------------------
C--   ASK THE NAME OF THE FILE WHERE RESULTS ARE STORED
C-
      IF (CHOIX$.EQ.'Y'.OR.CHOIX$.EQ.'y') THEN
103     WRITE(*,1007)'Results Data File (8 letters maxi) : '
        READ(*,'(A)')RES1$
        IDBLAN=INDEX(RES1$,' ')-1
        IF (IDBLAN.EQ.0.OR.IDBLAN.GT.8) GOTO 103
        RES3$=RES1$(1:IDBLAN)//'.CNF'
        RES2$=RES1$(1:IDBLAN)//'.FIN'
        RES1$=RES1$(1:IDBLAN)//'.RES'
        WRITE(*,1007)'Comments : '
        READ(*,'(A)')DESC$
          GOTO 200

      ELSEIF (CHOIX$.EQ.'N'.OR.CHOIX$.EQ.'n') THEN
          RETURN

      ELSE
          GOTO 102
      ENDIF


C//////////////////////////////////////////////////////////////////////
C/                                                                     /
C/            SIMULATION  OF A COMPLETE PROCESS                        /
C/                                                                     /
C//////////////////////////////////////////////////////////////////////


C-- INITIALISATION FOR SKYMER ROUTINE - VECTORS OF COMPOUNDS
C-- WARNING MUST BE MODIFIED FOR NEW COMPOUNDS
C         1-NH3
C         2-HNO3
C         3-HNO2
C         4-CO2
C         5-O2
C         6-H2O
C         7-H2SO4
C         8-H3PO4
C         9-BIOMASSE LIBRE NS
C         10-BIOMASSE LIBRE NB
C
C-- ORGANISATION OF THE DIFFERENTIAL VECTOR Y
C   1-10                                  LIQUID A
C   11-20                                 GAS A
C   10(2N)+1 - 10(2N)+10                  LIQUID SECTION N OF B
C   10(2N+1)+1 - 10(2N+1)+10              GAS SECTION N OF B
C   10(2(N+1))+1 - 10(2(N+1))+10          LIQUID C
C   10(2(N+1))+11 - 10(2(N+1))+20         GAS C
C   10(2(N+2))+1 - 10(2(N+2)+N            BIOMASS NS FIXED IN SECTION N
C   10(2(N+2))+N+1 - 10(2(N+2))+2N        BIOMASS NB FIXED IN SECTION N
C   10(2(N+2))+2N+1 - 10(2(N+2))+2N+N+2 THEORETICAL EVOLUTION OF PH



C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C--  INITIALISATION OF THE PARAMETERS FOR SIMULATION FOR NEW PROCESS
C--                        DEFAULT INITIALISATION
C-- Liquid = Input concentration
C-- Gas = Input fraction
C-- Dissolved gas = Saturation
C-- Biomass (version since 03/97) = on all column Ns=70% of entry value
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------

200   IF (INI.EQ.1.OR.INIPSP.EQ.1) THEN
      NEQ=N*10*2+10*2*(1+1)+2*N+N+2
      SATUCO2=CINL(6)*(CING(4)*(8.314*(273.15+T)/(P*101.3)))/CSAT(4)
      SATUO2=CINL(6)*(CING(5)*(8.314*(273.15+T)/(P*101.3)))/CSAT(5)
      SATUCO2=SATUCO2*(1+(KA(4)/(10**(-PHINI)))*
     +(1+KA(16)/10**(-PHINI)))
      EAUGAZ=csat(6)/(8.314*(273.15+T)/(P*1.013E2))
```

```
C-----------------------------------------------------------------------
C-- STOCKING THE 10 COMPOUNDS NON IONICS FOR UNIT A
C-
      DO 150 I=1,10
            Y0(I)=CINL(I)

        IF (I.EQ.4.OR.I.EQ.5) then
           Y0(5)=SATUO2
           Y0(4)=SATUCO2
        ENDIF
        Y0(10+I)=CING(I)
        IF (I.EQ.6) THEN
        Y0(10+I)=EAUGAZ
        ENDIF

150   CONTINUE


C-----------------------------------------------------------------------
C-- STOCKING THE  10 NON IONIC COMPOUNDS FOR SECTION N OF B
C-
      DO 151 J=1,N
         DO 152 I=1,10
            Y0(10*(2*J)+I)=CINL(I)

        IF (I.EQ.4.OR.I.EQ.5) then
           Y0(10*(2*J)+5)=SATUO2
           Y0(10*(2*J)+4)=SATUCO2
        ENDIF
        Y0(10*(2*J+1)+I)=CING(I)
        IF (I.EQ.6) THEN
        Y0(10*(2*J+1)+I)=EAUGAZ
        ENDIF
152   CONTINUE
151   CONTINUE


C-----------------------------------------------------------------------
C-- STOCKING OF THE 10 NON IONIC COMPOUNDS OF UNIT C
C-
      DO 154 I=1,10
         Y0(10*(2*(N+1))+I)=CINL(I)

        IF (I.EQ.4.OR.I.EQ.5) then
           Y0(10*(2*(N+1))+5)=SATUO2
           Y0(10*(2*(N+1))+4)=SATUCO2
        ENDIF
        Y0(10*(2*(N+1)+1)+I)=CING(I)
        IF (I.EQ.6) THEN
        Y0(10*(2*(N+1)+1)+I)=EAUGAZ
        ENDIF
154   CONTINUE


C-----------------------------------------------------------------------
C-- STOCKING FIXED BIOMASS
C-
      DO 155 I=1,N
        Y0(10*(2*(N+2))+I)=CIBIO*0.7
        Y0(10*(2*(N+2))+N+I)=CIBIO*0.3
155   CONTINUE


C-----------------------------------------------------------------------
C-- STOCKING H+ FOR pH
C-
      DO 156 I=1,N+2
        Y0(10*(2*(N+2))+2*N+I)=10**(-PHINI)
156   CONTINUE
```

```
      ELSEIF (INI.EQ.2.OR.INIPSP.EQ.2) THEN
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C             FILE INIATILISATION OF CONCENTRATION
C      REQUIRE A FILE CORRECTLY DIMENSIONNED FOR THE PROCESS
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      OPEN(1,FILE=INI$,FORM='FORMATTED')

      NEQ=N*10*2+10*2*(1+1)+2*N+N+2

C-----------------------------------------------------------------------
C-- UNITE A
C-
         rappg=1/(8.314*(273.15+T)/(P*101.3))
         read(1,*)Y0(1),y0(2),y0(3),y0(4),y0(5),y0(6),y0(7),y0(8),
     +y0(9),Y0(10),Y0(11),y0(12),y0(13),
     +y0(14),Y0(15),y0(16),
     +y0(17),y0(18),y0(19),Y0(20),
     +cibio,cibio,Y0(10*(2*(N+2))+2*N+1)

C-----------------------------------------------------------------------
C-- SECTION N OF B
C-
      DO 51 J=1,N
         read(1,*)Y0(10*(2*J)+1),y0(10*(2*J)+2),y0(10*(2*J)+3),
     +y0(10*(2*J)+4),y0(10*(2*J)+5),y0(10*(2*J)+6),y0(10*(2*J)+7),
     +y0(10*(2*J)+8),y0(10*(2*J)+9),Y0(10*(2*J)+10),
     +Y0(10*(2*J)+11),y0(10*(2*J)+12),
     +y0(10*(2*J)+13),y0(10*(2*J)+14),
     +y0(10*(2*J)+15),y0(10*(2*J)+16),
     +y0(10*(2*J)+17),y0(10*(2*J)+18),
     +y0(10*(2*J)+19),Y0(10*(2*J)+20),
     +Y0(10*(2*(N+2))+J),Y0(10*(2*(N+2))+N+J),
     +Y0(10*(2*(N+2))+2*N+1+J)
51    CONTINUE

C-----------------------------------------------------------------------
C--UNIT C
C-

         read(1,*)Y0(10*(2*(N+1))+1),y0(10*(2*(N+1))+2),
     +y0(10*(2*(N+1))+3),y0(10*(2*(N+1))+4),y0(10*(2*(N+1))+5),
     +y0(10*(2*(N+1))+6),y0(10*(2*(N+1))+7),y0(10*(2*(N+1))+8),
     +y0(10*(2*(N+1))+9),Y0(10*(2*(N+1))+10),Y0(10*(2*(N+1))+11),
     +y0(10*(2*(N+1))+12),y0(10*(2*(N+1))+13),y0(10*(2*(N+1))+14),
     +y0(10*(2*(N+1))+15),y0(10*(2*(N+1))+16),y0(10*(2*(N+1))+17),
     +y0(10*(2*(N+1))+18),y0(10*(2*(N+1))+19),Y0(10*(2*(N+1))+20),
     +cibio,cibio,Y0(10*(2*(N+2))+2*N+N+2)


      CLOSE(1)

      do 53 j=11,20
       y0(10*(2*(N+1))+j)=y0(10*(2*(N+1))+j)*rappg
       do 52 i=1,N
          y0(10*(2*I)+j)=y0(10*(2*I)+j)*rappg
52     continue
       y0(j)=y0(j)*rappg
53    continue


C-----------------------------------------------------------------------
C---  END OF VECTOR INITIALISATION FOR SIMULATIONS

      ENDIF



C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C--                DEFINITION OF SAVING FILES
C--
C--    *******.RES CONTAINS THE COMPLETE COLUMN CONCENTRATIONS PROFILE
```

```
C--     *******.FIN CONTAINS THE COLUMN OUTPUT CONCENTRATIONS
C--     *******.CNF CONTAINS THE COLUMN CONFIGURATION FOR THE PROCESS
C-------------------------------------------------------------------------
C-------------------------------------------------------------------------

       OPEN(1,FILE=RES1$)
       OPEN(2,FILE=RES2$)
       OPEN(3,FILE=RES3$)

       IF (INI.EQ.2) THEN
           WRITE(1,'(A,A)')'#  File initialisation : ',INI$
           WRITE(2,'(A,A)')'#  File initialisation : ',INI$
           WRITE(3,'(A,A)')'#  File initialisation : ',INI$
       ENDIF
       WRITE(1,'(A,A)')'# ',DESC$
       WRITE(1,1020)'#  Time  '
       WRITE(1,1020)'  Unit  '
       WRITE(2,'(A,A)')'# ',DESC$
       WRITE(2,1020)'#  Time  '

       DO 190 I=1,10
          WRITE(1,1021)'Liq',NOM$(I)
          WRITE(2,1021)'Liq',NOM$(I)
190    CONTINUE
       DO 191 I=1,10
          WRITE(1,1021)'Gas',NOM$(I)
          WRITE(2,1021)'Gas',NOM$(I)
191    CONTINUE
       WRITE(1,1020)'Ns fix'
       WRITE(1,1020)'Nb fix'
       WRITE(2,1020)'Bio mean'
       WRITE(2,1020)'Pop mean'
       WRITE(1,1020)'H+     '
       WRITE(2,1020)'H+     '
       WRITE(1,1020)'Height     '
       WRITE(1,1020)'Bio Thick '
       WRITE(1,1020)'Thick Lim '
       WRITE(1,*)
       WRITE(2,*)


C-------------------------------------------------------------------------
C-------------------------------------------------------------------------
C--  SIMULATION PROCEDES COMPLET (INI=1 et INI=2)
C-------------------------------------------------------------------------
C-------------------------------------------------------------------------
       IF (INI.NE.3) THEN


C-------------------------------------------------------------------------
C-- CALLING RESOLUTION ROUTINE 'Runge Kutta Merson'
C-
       PRINT*
       PRINT*,' Processing [Hit Ctrl+C to abort]....'
       PRINT*
       CALL SAVEPARA ()
       TEMPO=float(TMPSFIN)
       CALL DRKMER(TMPSINI,TEMPO,Y0,NEQ,TMPS)
       TMPSINI=TEMPO



C-------------------------------------------------------------------------
C-- CONTINUING SIMULATION
C-
300    PRINT*
       WRITE(*,1007)' Continue Simulation (Y/N) : '
       READ(*,'(A)')CHOIX$
       IF (CHOIX$.EQ.'Y'.OR.CHOIX$.EQ.'y') THEN

C-- REINITIALISE FLOW RATES
            WRITE(*,1100)
            WRITE(*,1001)
            WRITE(*,1002)'New flow rates:'
```

```
          WRITE(*,1003)
          WRITE(*,1040)'Liquid Flow Rate ( ',FINP,'m3/h ): '
          READ(*,*,err=301)REP
          FINP=REP

301       WRITE(*,1040)'Liquid Recycling ( ',RL,'m3/h ): '
          READ(*,*,err=302)REP
          RL=REP

302       WRITE(*,1040)'Gas Flow Rate ( ',GINP,'m3/h ): '
          READ(*,*,err=303)REP
          GINP=REP

303       WRITE(*,1040)'Gas Recycling ( ',RG,'m3/h ): '
          READ(*,*,err=304)REP
          RG=REP


C--- REINITIALISE LIQUID INPUTS CONCENTRATION
304    PRINT*
       PRINT*
       WRITE(*,1001)
       WRITE(*,1002)'Liquid Inputs concentrations'
       WRITE(*,1003)

380     WRITE(*,1008)'Concentration of ',NOM$(1),'(',CINL(1),'mol/l): '
        READ(*,*,err=381)REP
        CINL(1)=REP
        IF (CINL(1).LT.0) THEN
           WRITE(*,1009)'NEGATIVE'
           GOTO 380
        ENDIF
381    WRITE(*,1008)'Concentration of ',NOM$(2),'(',CINL(2),'mol/l): '
       READ(*,*,err=382)REP
       CINL(2)=REP
        IF (CINL(2).LT.0) THEN
           WRITE(*,1009)'NEGATIVE'
           GOTO 381
        ENDIF
382    WRITE(*,1008)'Concentration of ',NOM$(3),'(',CINL(3),'mol/l): '
       READ(*,*,err=383)REP
       CINL(3)=REP
        IF (CINL(3).LT.0) THEN
           WRITE(*,1009)'NEGATIVE'
           GOTO 382
        ENDIF

383    CINL(6)=55.55555555

       WRITE(*,1008)'Concentration of ',NOM$(7),'(',CINL(7),'mol/l): '
       READ(*,*,err=384)REP
       CINL(7)=REP
         IF (CINL(7).LT.0) THEN
            WRITE(*,1009)'NEGATIVE'
            GOTO 383
         ENDIF

384     WRITE(*,1008)'Concentration of ',NOM$(8),'(',CINL(8),'mol/l): '
        READ(*,*,err=385)REP
        CINL(8)=REP
         IF (CINL(8).LT.0) THEN
            WRITE(*,1009)'NEGATIVE'
            GOTO 384
         ENDIF


C--- REINITIALISE GAS INPUTS CONCENTRATION
385    PRINT*
       PRINT*
       WRITE(*,1001)
       WRITE(*,1002)'Molar Gas Input Fraction (Completed by N2)'
       WRITE(*,1003)
       WRITE(*,1008)'Molar Fraction of ',NOM$(4),'(',CING(4),') : '
       READ(*,*,err=386)REP
       CING(4)=REP
```

```
      CING(4)=CING(4)/(8.314*(273.15+T)/(P*1.013E2))

386   WRITE(*,1008)'Molar Fraction of ',NOM$(5),'(',CING(5),') : '
      READ(*,*,err=387)REP
      CING(5)=REP
      CING(5)=CING(5)/(8.314*(273.15+T)/(P*1.013E2))

387   WRITE(*,1008)'Molar Fraction of ',NOM$(6),'(',CING(6),') : '
      READ(*,*,err=388)REP
      CING(6)=REP
      CING(6)=CING(6)/(8.314*(273.15+T)/(P*1.013E2))
388   PRINT*

C--- New Time Initialisation

      WRITE(*,1007)' Lenght of Simulation (Hours): '
      READ*,TMPSFIN
      TMPSFIN=TMPSFIN+INT(TMPSINI)
      PRINT*,'Total Lenght of Simulation : ',TMPSFIN, 'h'
399    WRITE(*,1007)' Number of iterative times -Maximun 300- : '
      READ*,TMPS
      IF (TMPS.LE.1.OR.TMPS.GT.300) THEN
        WRITE(*,1009)'BETWEEN 2 AND 300'
        GOTO 399
      ENDIF


C////////////////////////////////////////////////////////////////////////
C/     CONFIRM CONFIGURATION FOR CONTINUING THE SIMULATION             /
C////////////////////////////////////////////////////////////////////////

400   WRITE(*,1100)
      WRITE(*,1030)'Simulation Configuration'
      TMPO=(float(TMPSFIN)-TMPSINI)/FLOAT(TMPS-1)
      WRITE(*,1031)'Lenght: ',TMPSFIN-INT(TMPSINI),' h  of step ',TMPO
      WRITE(*,1041)'Total: ',TMPSFIN,' h'
      WRITE(*,1032)'Options ','- Calcul of equilibria KA et Ki : ',
     +'n'
      WRITE(*,1032)' ','- Pseudo Steady State O2 gas : ',CHOIXO2$
      WRITE(*,1032)' ','- Pseudo Steady State CO2 gas : ',CHOIXCO2$
      WRITE(*,1032)' ','- Pseudo Steady State H2O : ',CHOIXH2O$
      WRITE(*,1033)'Inputs on the column'
      WRITE(*,1034)'Liquid [mol/h]',NOM$(1),CINL(1),NOM$(2),CINL(2),
     +NOM$(3),CINL(3)
      WRITE(*,1034)' ',NOM$(6),CINL(6),NOM$(7),CINL(7),
     +NOM$(8),CINL(8)
      WRITE(*,1034)'Gas [mol. frac.]',NOM$(4),
     +CING(4)*(8.314*(273.15+T)/(P*101.3)),NOM$(5),
     +CING(5)*(8.314*(273.15+T)/(P*101.3)),
     +NOM$(6),CING(6)*(8.314*(273.15+T)/(P*101.3))
      WRITE(*,1037)'BED [g/l]','Ns fixed',CIBIO,'Nb fixed',CIBIO
      WRITE(*,1033)'Column Design'
      WRITE(*,1035)'Segments :',N,'% Vol part A :',VA/(VA+VB+VC)*100,
     +'% Vol part C :',VC/(VA+VB+VC)*100
      WRITE(*,1036)'Liquid ','Input flow (m3/h) :',fINP,
     +'Recycling ratio :',RL
      WRITE(*,1036)'Gas ','Input flow (m3/h) :',GINP,
     +'Recycling ratio :',RG
      WRITE(*,1003)


      PRINT*
      PRINT*
402   WRITE(*,1007)' Running Simulation (Y/N) : '
      READ(*,'(A)')CHOIX$
      IF (CHOIX$.EQ.'Y'.OR.CHOIX$.EQ.'y') THEN
          GOTO 500
      ELSEIF (CHOIX$.EQ.'N'.OR.CHOIX$.EQ.'n') THEN
          GOTO 300
      ELSE
          GOTO 402
      ENDIF
```

```
C-----------------------------------------------------------------------
C--- Calcul of the new column design Before simulation
C-
500    MENU=2
       CALL CONFIGSIM(MENU)


C-----------------------------------------------------------------------
C--- SIMULATION - Call of RKMER
C-
       PRINT*
       PRINT*,'Processing [Hit Ctrl+C to abort]....'
       print*
       CALL SAVEPARA()
       TEMPO=float(TMPSFIN)
       CALL DRKMER(TMPSINI,TEMPO,Y0,NEQ,TMPS)
       TMPSINI=TEMPO

C-----------------------------------------------------------------------
C--- Ask for continuing simulation
       GOTO 300


C-----------------------------------------------------------------------
C--- END OF CURRENT SIMLULATION
C-
       ELSEIF (CHOIX$.EQ.'N'.OR.CHOIX$.EQ.'n') THEN

C-----------------------------------------------------------------------
C-- CLOSE OF SAVING FILES
C-
       CLOSE(1)
       CLOSE(2)
       CLOSE(3)
           RETURN
       ELSE
           GOTO 300
       ENDIF


C/////////////////////////////////////////////////////////////////////////
C/          COLUMN CONCENTRATION PROFILE IN STEADY STATE (INI=3)          /
C/              CALCULUS OF d/dt=0 BY WEGSTEIN ALGORYTHME                 /
C/                CALCULUS OF BIOMASS PROFILE "By Hand"                   /
C/                                                                        /
C/    In Case of convergence problems:                                    /
C/        increase the number of Wegstein iterations (ITER_W) for compounds /
C/        increase the number of biomass reinitialisation (ITER_REINI)    /
C/        modify the test for wegstein convergence (WEG.FOR: CRIT)        /
C/        modify the biomass criterium                                    /
C/////////////////////////////////////////////////////////////////////////
       ELSEIF (INI.EQ.3) THEN


C---- Ask for the sensibility of the tests
C---- Sensibility for WEGSTEIN
C---- Sensibility for Biomass calculation derivaties based
       SENSI_W=1e-3
       SENSI_BIO=1.0e-6
           WRITE(*,1040)'Wegstein sensitivity ( ',SENSI_W,' ): '
           READ(*,*,err=625)REP
           SENSI_W=REP
625        WRITE(*,1040)'Biomass sensitivity ( ',SENSI_BIO,'): '
           READ(*,*,err=626)REP
           SENSI_BIO=REP



626    IF (INIPSP.EQ.1) THEN
C-----------------------------------------------------------------------
C-- Estimation des parametres pour le regime permanent a partir des valeurs
C-- d'entree sur la colonne
C-- Modification de l'initialisation par defaut:
C--     efficacite du procede= 95% NO3
```

```
C--                              3%  NH3
C--                              2%  NO2
C--     Gas: CO2=[entree]-95%*0.06*[N-entree]
C--          O2=[entree]-95%*1.9*[N-entree]
C--          H2O=Saturation
C--     Dissolved Gas = Saturation
C--     Biomass = Dilution*maintenance*0.95*[N-entree] on first unit
C--                and 0 on other

C-----------------------------------------------------------------------
C--    Calcul for gases
C-
       Y0_O2=(CING(5)*GINP-(CINL(1)*0.95*1.9*FINP))/GINP
       IF (Y0_O2.LT.1e-15) Y0_O2=1e-15
       Y0_CO2=(CING(4)*GINP-(CINL(1)*0.95*0.06*FINP))/GINP
       IF (Y0_CO2.LT.1e-15) Y0_CO2=1e-15

      SATUCO2=CINL(6)*(Y0_CO2*(8.314*(273.15+T)/(P*101.3)))/CSAT(4)
      SATUO2=CINL(6)*(Y0_O2*(8.314*(273.15+T)/(P*101.3)))/CSAT(5)
      SATUCO2=SATUCO2*(1+(KA(4)/(10**(-PHINI)))*
     +(1+KA(16)/10**(-PHINI)))


C-----------------------------------------------------------------------
C-- STOCKING THE 10 COMPOUNDS NON IONICS FOR UNIT A
C-
      DO 650 I=1,10
         IF (I.eq.1) Y0(I)=CINL(1)*0.03
         IF (I.eq.2) Y0(I)=CINL(1)*0.95
         IF (I.eq.3) Y0(I)=CINL(1)*0.02
         IF (I.eq.4) THEN
               Y0(I)=SATUCO2
               Y0(10+I)=Y0_CO2
         ENDIF
         IF (I.eq.5) THEN
               Y0(I)=SATUO2
               Y0(10+I)=Y0_O2
         ENDIF
650   CONTINUE

C-----------------------------------------------------------------------
C-- STOCKING THE  10 NON IONIC COMPOUNDS FOR SECTION N OF B
C-
      DO 651 J=1,N
         DO 652 I=1,10
            Y0(10*(2*J)+I)=CINL(I)
            IF (I.eq.1) Y0(10*(2*J)+I)=CINL(1)*0.03
            IF (I.eq.2) Y0(10*(2*J)+I)=CINL(1)*0.95
            IF (I.eq.3) Y0(10*(2*J)+I)=CINL(1)*0.02
            IF (I.eq.4) THEN
               Y0(10*(2*J)+I)=SATUCO2
               Y0(10*(2*J+1)+I)=Y0_CO2
           ENDIF
           IF (I.eq.5) THEN
               Y0(10*(2*J)+I)=SATUO2
               Y0(10*(2*J+1)+I)=Y0_O2
           ENDIF

652   CONTINUE
651   CONTINUE

C-----------------------------------------------------------------------
C-- STOCKING OF THE 10 NON IONIC COMPOUNDS OF UNIT C
C-
      DO 654 I=1,10
         Y0(10*(2*(N+1))+I)=CINL(I)

         IF (I.eq.1) Y0(10*(2*(N+1))+I)=CINL(1)*0.03
         IF (I.eq.2) Y0(10*(2*(N+1))+I)=CINL(1)*0.95
         IF (I.eq.3) Y0(10*(2*(N+1))+I)=CINL(1)*0.02
         IF (I.eq.4) THEN
               Y0(10*(2*(N+1))+I)=SATUCO2
               Y0(10*(2*(N+1)+1)+I)=Y0_CO2
         ENDIF
         IF (I.eq.5) THEN
               Y0(10*(2*(N+1))+I)=SATUO2
```

```
                Y0(10*(2*(N+1)+1)+I)=Y0_O2
            ENDIF
654     CONTINUE

C-----------------------------------------------------------------------
C-- STOCKING FIXED BIOMASS
C-
        DO 655 I=1,N
          IF (I.eq.1) then
           IF (N.GT.2) THEN
           Y0(10*(2*(N+2))+I)=(N-2)*(CINL(1)*0.95/(1+RL)/MUMAX(3)-1e-2)
           Y0(10*(2*(N+2))+N+I)=(N-2)*(CINL(1)*0.95/(1+RL)/MUMAX(4)-1e-2)
           ELSE
           Y0(10*(2*(N+2))+I)=(CINL(1)*0.95/(1+RL)/MUMAX(3))
           Y0(10*(2*(N+2))+N+I)=(CINL(1)*0.95/(1+RL)/MUMAX(4))
           ENDIF

          ELSE
              Y0(10*(2*(N+2))+I)=1e-2
              Y0(10*(2*(N+2))+N+I)=1e-2
          ENDIF
655     CONTINUE

C-----------------------------------------------------------------------
C-- STOCKING H+ FOR pH
C-

        DO 656 I=1,N+2
          Y0(10*(2*(N+2))+2*N+I)=10**(-2)
656     CONTINUE


        ELSE
C
C-- No Modifications of data from a file
C
        ENDIF


C-----------------------------------------------------------------------
C-- Iterative convergence methode of Wegstein (WEG.FOR)
C-- Solve the function F(X)=X
C-----------------------------------------------------------------------
        ITER_W=0
        CRIT=0.
        NCONV_W=0
        ITER_REINI=1

C-- Save Initialisation
        WRITE(1,*)'#Initialisation Values'
        WRITE(2,*)'#Initialisation Values'
        CALL SAVESIM(TMPSINI,Y0,K)

C-- Definition des bornes
C-- Concentration>=0
C-- Biomasse>=debit N entree/2*maintenance
        DO 605 I=1,NEQ
           XMIN(I)=1e-15
           XMAX(I)=55.555555555
605     continue

C-- Sauvegarde valeurs initiales
         Do 611 I=1,NEQ
            Y0INI(I)=Y0(I)
611      CONTINUE

         DO 606 i=1,N
C          XMIN(10*(2*(N+2))+I)=CINL(1)*FINP/(2*MUMAX(3))
C          XMIN(10*(2*(N+2))+N+I)=CINL(1)*FINP/(3*MUMAX(4))
           XMIN(10*(2*(N+2))+I)=1e-10
           XMIN(10*(2*(N+2))+N+I)=1e-10
606      CONTINUE

C--- Calcul de f(x)=dx/dt pour x
600     CALL DERIV(Y0,TMPSINI,FY0)
```

```
C--- Stockage valeur biomasse
      DO 607 I=1,N
         BIONS_TMP(I)=Y0(10*(2*(N+2))+I)
         BIONB_TMP(I)=Y0(10*(2*(N+2))+N+I)

         FY0(10*(2*(N+2))+I)=0.
         FY0(10*(2*(N+2))+N+I)=0.

607   CONTINUE

      ITER_W=ITER_W+1
      WRITE(*,1110)'Iteration: ',ITER_REINI,ITER_W,CRIT,NCRIT,
     &FY0(NCRIT),NCONV_BIO,KCRIT

C---- Definition of the function F(x) for WEGSTEIN procedure
C---- F(x)=x - 10*dx/dt
      DO 601 I=1,NEQ
          FY0(I)=-FY0(I)*10+Y0(I)
601   CONTINUE


C---- Call of Wegstein procedure
      CALL WEG(Y0,FY0,XMIN,XMAX,XAT,YAT,NEQ,NCONV_W,SENSI_W,
     &CRIT,NCRIT,KCRIT)


C--- Storage of biomass values
      DO 608 I=1,N
         Y0(10*(2*(N+2))+I)=BIONS_TMP(I)
         Y0(10*(2*(N+2))+N+I)=BIONB_TMP(I)
608   CONTINUE


C---- End of simulation tests

      IF(NCONV_W.NE.1.AND.ITER_W.LT.25000) THEN
        GOTO 600
      ELSEIF (ITER_REINI.LT.20) THEN
        PRINT*
640     ITER_REINI=ITER_REINI+1


            WRITE(1,*)'#Intermediate Result Values',KCRIT,NCONV_BIO
            WRITE(2,*)'#Intermediate Result Values',KCRIT,NCONV_BIO
            CALL SAVESIM(TMPSINI,Y0,K)

C-------------------------------------------------------------------------
C--- Test Stability of the FIXED BIOMASS "By Hand methode"
C--- Test 1 : F'(X)/X  Great than 1e-3
C---                  ======> Different calculations of the biomass
C--- Test 2 : F'(X)/X  Great than 1e-5
C---                  ======>  Convergence fo X if F'(X) less than 1e-6
C---                  ======>  Or calcultion of a new value of X
C--- Test 1 and 2 passed then convergence for X
C---
C--- The stability of the bed is reached when all biomass values have
C--- converged to a stable value
C-------------------------------------------------------------------------
      CALL DERIV(Y0,TMPSINI,FY0)

      NCONV_BIO=0

      DO 610 I=1,N

      IF (DABS(FY0(10*(2*(N+2))+I)/Y0(10*(2*(N+2))+I)).GT.
     &SENSI_BIO*100) THEN

      IF (DABS(1e2*FY0(10*(2*(N+2))+I)).LT.Y0(10*(2*(N+2))+I)) THEN
       Y0(10*(2*(N+2))+I)=Y0(10*(2*(N+2))+I)+1e2*FY0(10*(2*(N+2))+I)
      ELSEIF (DABS(1e1*FY0(10*(2*(N+2))+I)).LT.Y0(10*(2*(N+2))+I)) THEN
       Y0(10*(2*(N+2))+I)=Y0(10*(2*(N+2))+I)+1e1*FY0(10*(2*(N+2))+I)
      ELSEIF (DABS(1e0*FY0(10*(2*(N+2))+I)).LT.Y0(10*(2*(N+2))+I)) THEN
       Y0(10*(2*(N+2))+I)=Y0(10*(2*(N+2))+I)+1e0*FY0(10*(2*(N+2))+I)
      ELSEIF (DABS(1e1*FY0(10*(2*(N+2))+I)).LT.Y0(10*(2*(N+2))+I)) THEN
       Y0(10*(2*(N+2))+I)=Y0(10*(2*(N+2))+I)+1e1*FY0(10*(2*(N+2))+I)
      ELSEIF (DABS(1e0*FY0(10*(2*(N+2))+I)).LT.Y0(10*(2*(N+2))+I)) THEN
```

```
          Y0(10*(2*(N+2))+I)=Y0(10*(2*(N+2))+I)+1e0*FY0(10*(2*(N+2))+I)

        ELSE
            IF (FY0(10*(2*(N+2))+I).LE.0.) THEN
              Y0(10*(2*(N+2))+I)=Y0(10*(2*(N+2))+I)/2
            ELSE
               Y0(10*(2*(N+2))+I)=Y0(10*(2*(N+2))+I)*2
            ENDIF
        ENDIF


        ELSEIF(DABS(FY0(10*(2*(N+2))+I)/Y0(10*(2*(N+2))+I)).GT.
       &SENSI_BIO*10) THEN
           IF (DABS(FY0(10*(2*(N+2))+I)).LT.SENSI_BIO) THEN
              Y0(10*(2*(N+2))+I)=BIONS_TMP(I)
              NCONV_BIO=NCONV_BIO+1
           ELSE
           Y0(10*(2*(N+2))+I)=Y0(10*(2*(N+2))+I)+100*FY0(10*(2*(N+2))+I)
           ENDIF


        ELSE
C----      Conserve the last saved biomass values instead of the
C----      Wegstein calculated ones
           Y0(10*(2*(N+2))+I)=BIONS_TMP(I)
           NCONV_BIO=NCONV_BIO+1
        ENDIF


        IF (DABS(FY0(10*(2*(N+2))+N+I)/Y0(10*(2*(N+2))+N+I))
       &.GT.SENSI_BIO*100) THEN

            IF (DABS(1e2*FY0(10*(2*(N+2))+N+I)).LT.
       &Y0(10*(2*(N+2))+N+I)) THEN
             Y0(10*(2*(N+2))+N+I)=Y0(10*(2*(N+2))+N+I)
       &+1e2*FY0(10*(2*(N+2))+N+I)
            ELSEIF (DABS(1e1*FY0(10*(2*(N+2))+N+I)).LT.
       &Y0(10*(2*(N+2))+N+I)) THEN
             Y0(10*(2*(N+2))+N+I)=Y0(10*(2*(N+2))+N+I)
       &+1e1*FY0(10*(2*(N+2))+N+I)
            ELSEIF (DABS(1e0*FY0(10*(2*(N+2))+N+I)).LT.
       &Y0(10*(2*(N+2))+N+I)) THEN
             Y0(10*(2*(N+2))+N+I)=Y0(10*(2*(N+2))+N+I)
       &+1e0*FY0(10*(2*(N+2))+N+I)
            ELSEIF (DABS(1e1*FY0(10*(2*(N+2))+N+I)).LT.
       &Y0(10*(2*(N+2))+N+I)) THEN
             Y0(10*(2*(N+2))+N+I)=Y0(10*(2*(N+2))+N+I)
       &+1e1*FY0(10*(2*(N+2))+N+I)
            ELSEIF (DABS(1e0*FY0(10*(2*(N+2))+N+I)).LT.
       &Y0(10*(2*(N+2))+N+I)) THEN
             Y0(10*(2*(N+2))+N+I)=Y0(10*(2*(N+2))+N+I)
       &+1e0*FY0(10*(2*(N+2))+N+I)

            ELSE
              IF (FY0(10*(2*(N+2))+N+I).LE.0.) THEN
                Y0(10*(2*(N+2))+N+I)=Y0(10*(2*(N+2))+N+I)/2
               ELSE
                Y0(10*(2*(N+2))+N+I)=Y0(10*(2*(N+2))+N+I)*2
               ENDIF
            ENDIF

        ELSEIF(DABS(FY0(10*(2*(N+2))+N+I)/Y0(10*(2*(N+2))+N+I))
       &.GT.SENSI_BIO*10) THEN
           IF (DABS(FY0(10*(2*(N+2))+N+I)).LT.SENSI_BIO) THEN
              Y0(10*(2*(N+2))+N+I)=BIONB_TMP(I)
              NCONV_BIO=NCONV_BIO+1
           ELSE
          Y0(10*(2*(N+2))+N+I)=Y0(10*(2*(N+2))+N+I)
       &+100*FY0(10*(2*(N+2))+N+I)
           ENDIF


        ELSE
C----      Conserve the last saved biomass values instead of the
C----      Wegstein calculated ones
```

```
          Y0(10*(2*(N+2))+N+I)=BIONB_TMP(I)
          NCONV_BIO=NCONV_BIO+1
        ENDIF

610    CONTINUE

C---- Test of Stability of the bed
       IF (NCONV_BIO.EQ.(2*N).AND.NCONV_W.EQ.1) GOTO 630

C---- Reinitailise
       IF (NCONV_W.NE.1) THEN
          IF (KCRIT.GT.(NEQ-3)) THEN
             WRITE(1,*)'#Intermediate Result Values',KCRIT,NCONV_BIO
             WRITE(2,*)'#Intermediate Result Values',KCRIT,NCONV_BIO
             CALL SAVESIM(TMPSINI,Y0,K)
          ENDIF
        DO 618 I=1,N
          BIONS_TMP(I)=Y0(10*(2*(N+2))+I)
          BIONB_TMP(I)=Y0(10*(2*(N+2))+N+I)
618     CONTINUE
        DO 619 I=1,NEQ
          Y0(I)=Y0INI(I)
619     CONTINUE
        DO 620 I=1,N
          Y0(10*(2*(N+2))+I)=BIONS_TMP(I)
          Y0(10*(2*(N+2))+N+I)=BIONB_TMP(I)
620     CONTINUE

        ENDIF

       ITER_W=0.
       GOTO 600

       ELSEIF (ITER_REINI.GE.20) THEN
         PRINT*,'Not Convergent '
         IF (NCONV_BIO.EQ.(2*N)) PRINT*,'Convergent Biomasse'

C----    Do another serie of tests if not convergent
641    WRITE(*,1007)' Do another serie (Y/N) : '
       READ(*,'(A1)')CHOIX$
       IF (CHOIX$.EQ.'Y'.OR.CHOIX$.EQ.'y') THEN
           ITER_REINI=0
           GOTO 600
       ELSEIF (CHOIX$.EQ.'N'.OR.CHOIX$.EQ.'n') THEN
           GOTO 642
       ENDIF
       GOTO 641

642    pause
         GOTO 603
       ENDIF




C---- Convergence
630    CALL SAVEPARA()
       PRINT*,'Convergent',NCONV_W,NCONV_BIO,ITER_REINI
       pause

C-------------------------------------------------------------------------
C-- Save results
603     WRITE(1,*)'#Result Values','Nber equations',KCRIT,'sur',NEQ,
      &'Biomass',NCONV_BIO,'on',2*N
        WRITE(2,*)'#Result Values''Nber equations',KCRIT,'sur',NEQ,
      &'Biomass',NCONV_BIO,'on',2*N
        CALL SAVESIM(TMPSINI,Y0,K)


C-------------------------------------------------------------------------
C-- CLOSE OF SAVING FILES
C-
       CLOSE(1)
       CLOSE(2)
       CLOSE(3)
```

```
      RETURN


C////////////////////////////////////////////////////////////////////////
C/            End of CALCOL subroutine
C////////////////////////////////////////////////////////////////////////
      ENDIF

C////////////////////////////////////////////////////////////////////////
C                          FORMATS
C////////////////////////////////////////////////////////////////////////
1001  FORMAT(1x,'É',77('Í'),'»')
1002  FORMAT(1x,'°',T10,A,T80,'°')
1003  FORMAT(1x,'È',77('Í'),'¼')
1004  FORMAT(1x)
1005  FORMAT(1x,5X,'Choix : ',\)
1006  FORMAT(1x,'°',T80,'°')
1007  FORMAT(A,\)
1008  FORMAT(1x,A,A6,A,F10.6,A,\)
1009  FORMAT(1x,'   WARNING !!!     THESE VALUE CAN ONLY BE ',A)
1010  FORMAT(1x,'°',T3,A6,T10,F9.6,T23,A6,T30,F9.6,T43,A6,T50,F9.6,T63,
     &A6,T70,F9.6,T80,'°')
1011  FORMAT(1x,'°', T5,A,A,T80,'°')
1012  FORMAT(1x,'°', T3,A,F10.6,A,A,F10.6,T80,'°')
1013  FORMAT(1x,'°',T3,'   yCO2 : ',F10.6,'   yO2 : ',F10.6,
     &'   yH2O : ',F10.6,T80,'°')
1020  FORMAT(A10,'   ',\)
1021  FORMAT(A4,A6,'   ',\)
1030  FORMAT(1X,'É',5('Í'),'[',A25,']',45('Í'),'»')
1031  FORMAT(1X,'°',T5,A,I5,A,F6.2,T80,'°')
1032  FORMAT(1X,'°',t5,a,t15,a,a,t80,'°')
1033  format(1x,'Ì',5('Ä'),'[',A25,']',45('Ä'),'¹')
1034  FORMAT(1x,'°',t5,a,t24,a7,f9.6,t44,a7,f9.6,t60,a7,f9.6,t80,'°')
1037  FORMAT(1x,'°',t5,a,t24,a7,f9.6,t44,a7,f9.6,t80,'°')
1035  FORMAT(1x,'°',t5,a,i3,t26,a,f6.3,t52,a,f6.3,t80,'°')
1036  FORMAT(1x,'°',t5,a,t15,a,f10.6,t50,a,f6.2,t80,'°')
1040  FORMAT(1X,A,G15.8,A,\)
1041  FORMAT(1X,'°',T5,A,I5,A,T80,'°')
1100  FORMAT(24(/))
1110  FORMAT('+',A11,t13,i2,t17,I6,t25,G10.4,t37,I3,t42,G10.4,T54,I2,
     &t58,I3)

      RETURN
      END


C////////////////////////////////////////////////////////////////////////
C////////////////////////////////////////////////////////////////////////
C/                        Modeles                                      /
C/        Reduction of the number of differential equation by          /
C/        Using only non ionic compounds - calculation for ionic       /
C/        Included in the calculations - H+ neutralised                /
C////////////////////////////////////////////////////////////////////////
C////////////////////////////////////////////////////////////////////////


C////////////////////////////////////////////////////////////////////////
C             MODELE 3 : NO BIOFILM LIMITATION SUPPOSED
C////////////////////////////////////////////////////////////////////////
      SUBROUTINE DERIV(Y,X,FCT)
C-------------------------------------------------------------------------
C--   DECLARATIONS
C-------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)

      INTEGER NMAX,CORPMAX,TMPSMAX,RMAX
      PARAMETER(NMAX=20)
      PARAMETER(CORPMAX=30)
      PARAMETER(TMPSMAX=300)
      PARAMETER(RMAX=4)

      INTEGER N,CORP,TMPS,REAC
      INTEGER PSTO2,PSTCO2,PSTH2O

      REAL*8 REHL,REHG,EPSL,EPSG,EPS,F,G,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
```

```
      REAL*8 LONG,SECT,DELTAP,DCOL,DH
      REAL*8 T,P,PHINI,PHA,PHC
      REAL*8 PHB(NMAX)
      REAL*8 MUMAX,KI,KS,KA,KLGAZ,KLBIO
      REAL*8 RO


      REAL*8 ASPGAZ,ASPBIO

      DIMENSION C(CORPMAX),CSU(CORPMAX),CPR(CORPMAX),COUT(CORPMAX)
      DIMENSION FCTL(CORPMAX),FCTG(CORPMAX)
      DIMENSION CINL(corpmax),CING(CORPMAX)
      DIMENSION Y(650),FCT(650)
      DIMENSION RSNS(CORPMAX)
      DIMENSION RSNB(CORPMAX)

      DIMENSION HBIO(NMAX),RNBIO(NMAX)

      DIMENSION CSAT(CORPMAX)

      DIMENSION D(CORPMAX),KLBIO(CORPMAX),KLGAZ(CORPMAX)

      DIMENSION KS(RMAX,CORPMAX),KI(RMAX,CORPMAX)
      DIMENSION MUMAX(RMAX)


      DIMENSION KA(CORPMAX)

      DIMENSION STO(RMAX,CORPMAX)

      DIMENSION YIM1(650)

      COMMON/REACBIO2/MUMAX,KS,KI,STO
      COMMON/PHYINI/T,P,PHINI
      COMMON/PHYPH/KA
      COMMON/PHYPH2/PHA,PHC,PHB
      COMMON/PHYTRANS/D,KLBIO,KLGAZ,ASPGAZ,ASPBIO
      COMMON/PHYTRANS2/CSAT
      COMMON/BILLE/RO,HBIO,RNBIO,BWO
      COMMON/COLON/EPSL,EPSG,EPS,FINP,GINP,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      COMMON/COLON2/REHL,REHG,LONG,DCOL,SECT,DELTAP,DH
      COMMON/TAB/N,CORP,TMPS,REAC
      COMMON/CINI/CINL,CING
      COMMON/RKMERY/XIM1,YIM1,HITER
      COMMON/STEADY/PSTO2,PSTCO2,PSTH2O


      PMBIO=23.1443
      FREC=FINP*RL
      F=FINP+FREC
      GREC=GINP*RG
      G=GINP+GREC
      FOUT=FINP
      GOUT=GINP

C////////////////////////////////////////////////////////////////////////
C////////////////////////////////////////////////////////////////////////
C                       UNIT A OF THE FIXED BED
C////////////////////////////////////////////////////////////////////////
C////////////////////////////////////////////////////////////////////////


C---INPUT CONCNETRATIONS ON THE COLUMN ================================

C-----------------------------------------------------------------------
C--   Initialisation
C-
      CTNH3=CINL(1)
      CTNO3H=CINL(2)
      CTNO2H=CINL(3)
      CTCO2=CINL(4)
      CTO2=CINL(5)
      CTH2O=CINL(6)
      CTH2SO=CINL(7)
      CTH3PO=CINL(8)
      CTNS=CINL(9)
```

```
      CTNB=CINL(10)


C-------------------------------------------------------------------------
C--   PH  EQUILIBRIA CALCULATION FOR THE FIXED PHINI
C-

      CPR(11)=10**(-PHINI)
      CPR(12)=10**(PHINI-14)

      CPR(1)=CTNH3*CPR(12)/(KA(1)+CPR(12))
      CPR(13)=CTNH3-CPR(1)

      CPR(2)=CTNO3H*CPR(11)/(KA(2)+CPR(11))
      CPR(14)=CTNO3H-CPR(2)

      CPR(3)=CTNO2H*CPR(11)/(KA(3)+cpr(11))
      CPR(15)=CTNO2H-CPR(3)

      CPR(16)=CTCO2/(10**(-PHINI)/KA(4)+1+KA(16)/(10**(-PHINI)))
      CPR(17)=CPR(16)*KA(16)/10**(-PHINI)
      CPR(4)=CTCO2-CPR(16)-CPR(17)

      CPR(18)=CTH2SO/(10**(-PHINI)/KA(7)+1+KA(18)/(10**(-PHINI)))
      CPR(19)=CPR(18)*KA(18)/(10**(-PHINI))
      CPR(7)=CTH2SO-CPR(18)-CPR(19)


      CPR(21)=CTH3PO/((10**(-2*PHINI)/(KA(8)*KA(20)))+
     +10**(-PHINI)/KA(20)+1+KA(21)/10**(-PHINI))
      CPR(22)=KA(21)*CPR(21)/10**(-PHINI)
      CPR(20)=CPR(21)*10**(-PHINI)/KA(20)
      CPR(8)=CTH3PO-CPR(21)-CPR(20)-CPR(22)

      CPR(5)=CTO2
      CPR(6)=CTH2O
      CPR(9)=CTNS
      CPR(10)=CTNB


C=== CONCENTRATIONS IN UNIT A OF THE COLUMN ==============================

C-------------------------------------------------------------------------
C--   INITIALISATION FROM SKYKMER VARIABLES
C-
      IND=20
      CTNH3=Y(IND+1)
      CTNO3H=Y(IND+2)
      CTNO2H=Y(IND+3)
      CTCO2=Y(IND+4)
      CTO2=Y(IND+5)
      CTH2O=Y(IND+6)
      CTH2SO=Y(IND+7)
      CTH3PO=Y(IND+8)
      CTNS=Y(IND+9)
      CTNB=Y(IND+10)


C-------------------------------------------------------------------------
C--   PH  EQUILIBRIA CALCULATION FOR THE FIXED PHINI
C-
      CSU(11)=10**(-PHINI)
      CSU(12)=10**(PHINI-14)

      CSU(1)=CTNH3*CSU(12)/(KA(1)+csu(12))
      CSU(13)=CTNH3-CSU(1)

      CSU(2)=CTNO3H*CSU(11)/(KA(2)+csu(11))
      CSU(14)=CTNO3H-CSU(2)

      CSU(3)=CTNO2H*CSU(11)/(KA(3)+csu(11))
      CSU(15)=CTNO2H-CSU(3)

      CSU(16)=CTCO2/(10**(-PHINI)/KA(4)+1+KA(16)/10**(-PHINI))
      CSU(17)=CSU(16)*KA(16)/10**(-PHINI)
```

```
      CSU(4)=CTCO2-CSU(16)-CSU(17)

      CSU(18)=CTH2SO/(10**(-PHINI)/KA(7)+1+KA(18)/10**(-PHINI))
      CSU(19)=CSU(18)*KA(18)/10**(-PHINI)
      CSU(7)=CTH2SO-CSU(18)-CSU(19)


      CSU(21)=CTH3PO/((10**(-2*PHINI)/(KA(8)*KA(20)))+
     +10**(-PHINI)/KA(20)+1+KA(21)/10**(-PHINI))
      CSU(22)=KA(21)*CSU(21)/10**(-PHINI)
      CSU(20)=CSU(21)*10**(-PHINI)/KA(20)
      CSU(8)=CTH3PO-CSU(21)-CSU(20)-CSU(22)

      CSU(5)=CTO2
      CSU(6)=CTH2O
      CSU(9)=CTNS
      CSU(10)=CTNB



C=== OUTPUT CONCENTRATIONS ON THE COLUMN  ================================

C-----------------------------------------------------------------------
C--   INITIALISATION FROM SKYKMER VARIABLES

      IND=10*2*(N+1)
      CTNH3=Y(IND+1)
      CTNO3H=Y(IND+2)
      CTNO2H=Y(IND+3)
      CTCO2=Y(IND+4)
      CTO2=Y(IND+5)
      CTH2O=Y(IND+6)
      CTH2SO=Y(IND+7)
      CTH3PO=Y(IND+8)
      CTNS=Y(IND+9)
      CTNB=Y(IND+10)


C-----------------------------------------------------------------------
C--   PH  EQUILIBRIA CALCULATION FOR THE FIXED PHINI
C-
      COUT(11)=10**(-PHINI)
      COUT(12)=10**(PHINI-14)

      COUT(1)=CTNH3*COUT(12)/(KA(1)+cout(12))
      COUT(13)=CTNH3-COUT(1)

      COUT(2)=CTNO3H*COUT(11)/(KA(2)+cout(11))
      COUT(14)=CTNO3H-COUT(2)

      COUT(3)=CTNO2H*COUT(11)/(KA(3)+cout(11))
      COUT(15)=CTNO2H-COUT(3)

      COUT(16)=CTCO2/(10**(-PHINI)/KA(4)+1+KA(16)/10**(-PHINI))
      COUT(17)=COUT(16)*KA(16)/10**(-PHINI)
      COUT(4)=CTCO2-COUT(16)-C(17)

      COUT(18)=CTH2SO/(10**(-PHINI)/KA(7)+1+KA(18)/10**(-PHINI))
      COUT(19)=COUT(18)*KA(18)/10**(-PHINI)
      COUT(7)=CTH2SO-COUT(18)-COUT(19)


      COUT(21)=CTH3PO/((10**(-2*PHINI)/(KA(8)*KA(20)))+
     +10**(-PHINI)/KA(20)+1+KA(21)/10**(-PHINI))
      COUT(22)=KA(21)*COUT(21)/10**(-PHINI)
      COUT(20)=COUT(21)*10**(-PHINI)/KA(20)
      COUT(8)=CTH3PO-COUT(21)-COUT(20)-COUT(22)

      COUT(5)=CTO2
      COUT(6)=CTH2O
      COUT(9)=CTNS
      COUT(10)=CTNB
```

```
C=== CONCENTRATIONS IN UNIT A OF THE COLUMN==============================
C-------------------------------------------------------------------------
C--    INITIALISATION FROM SKYKMER VARIABLES
C-
       IND=0
       CTNH3=Y(IND+1)
       CTNO3H=Y(IND+2)
       CTNO2H=Y(IND+3)
       CTCO2=Y(IND+4)
       CTO2=Y(IND+5)
       CTH2O=Y(IND+6)
       CTH2SO=Y(IND+7)
       CTH3PO=Y(IND+8)
       CTNS=Y(IND+9)
       CTNB=Y(IND+10)


C-------------------------------------------------------------------------
C--    PH  EQUILIBRIA CALCULATION FOR THE FIXED PHINI
C-
       C(11)=10**(-PHINI)
       C(12)=10**(PHINI-14)

       C(1)=CTNH3*C(12)/(KA(1)+c(12))
       C(13)=CTNH3-C(1)

       C(2)=CTNO3H*C(11)/(KA(2)+c(11))
       C(14)=CTNO3H-C(2)

       C(3)=CTNO2H*C(11)/(KA(3)+c(11))
       C(15)=CTNO2H-C(3)

       C(16)=CTCO2/(10**(-PHINI)/KA(4)+1+KA(16)/10**(-PHINI))
       C(17)=C(16)*KA(16)/10**(-PHINI)
       C(4)=CTCO2-C(16)-C(17)

       C(18)=CTH2SO/(10**(-PHINI)/KA(7)+1+KA(18)/10**(-PHINI))
       C(19)=C(18)*KA(18)/10**(-PHINI)
       C(7)=CTH2SO-C(18)-C(19)


       C(21)=CTH3PO/((10**(-2*PHINI)/(KA(8)*KA(20)))+
      +10**(-PHINI)/KA(20)+1+KA(21)/10**(-PHINI))
       C(22)=KA(21)*C(21)/10**(-PHINI)
       C(20)=C(21)*10**(-PHINI)/KA(20)
       C(8)=CTH3PO-C(21)-C(20)-C(22)

       C(5)=CTO2
       C(6)=CTH2O
       C(9)=CTNS
       C(10)=CTNB



C=========================================================================
C    Differential equations for Gas and liquid phases on UNIT A
C=========================================================================


       DO 50 I=1,CORP
C-------------------------------------------------------------------------
C--  Gas-Liquid transfer reaction
C-
C- WARNING !: CSAT(I) IS NOT the concentration at saturation but the partition
C- coefficient (KI) stored in files
C- SATU IS the STAURATION concnetration calculated from KI(=CSAT)
C- For gases, the perfect gas law is used for the calculation

       IF (CSAT(I).NE.0.) THEN
          SATU=C(6)*(Y(10+I)*(8.314*(273.15+T)/(P*101.3)))/CSAT(I)
          FLUXGAZ=KLGAZ(I)*ASPGAZ*(SATU-C(I))

C--For water (index I=6) the flux liquid-->gas is calculated
          IF (I.EQ.6) THEN
```

```
            EAUGAZ=csat(6)/(8.314*(273.15+T)/(P*1.013E2))
            FLUXEAU=KLGAZ(I)*ASPGAZ*(EAUGAZ-Y(10+I))
          FLUXGAZ=-FLUXEAU*EPSG/EPSL
        ENDIF

C-- Reduction of low values
        IF (DABS(FLUXGAZ).LT.1e-16) FLUXGAZ=0.
      ELSE
        FLUXGAZ=0.
      ENDIF

C-------------------------------------------------------------------------
C--   DYNAMIC EQUATION FOR LIQUID PHASE
C-
C-    For value lower than 1e-15 We assumed a Concentration nul
C--   Modification Include in version 2.2
      IF (C(I).LE.1e-15) C(I)=0.
      IF (COUT(I).LE.1e-15) COUT(I)=0.
      IF (CSU(I).LE.1e-15) CSU(I)=0.

C--   For the water the problem is different because concentration
C--   cannot exceed 55.5555 mol/l
C--   If production of water ---> modification of volume.
C--   The effect on volume is neglected
C--   Concentration of water is limited to 55.5555 mol/l
C--   Modification included in version 2.2
C      IF (C(6).GT.55.55555555) C(6)=55.55555555
C--   Modification supressed in version 2.3

C-- Liquid relation
      FCTL(I)=FINP*CPR(I)+FREC*COUT(I)-(1+FBAK)*F*C(I)+FBAK*F*CSU(I)+
     +EPSL/EPS*VA*FLUXGAZ
      FCTL(I)=FCTL(I)/(EPSL*VA/EPS)

C--  liquid reaction for pseudo steady state
C-- PSEUDO-STEADY STATE FOR POUR OXYGENE d/dt=0
      IF (I.EQ.5.AND.PSTO2.EQ.1) THEN
       COXYL=FINP*CPR(I)+FREC*COUT(I)+FBAK*F*CSU(I)+EPSL/EPS*VA*FLUXGAZ
       COXYL=COXYL/((1+FBAK)*F)

C-- PSEUDO-STEADY STATE FOR CO2 d/dt=0
      ELSEIF (I.EQ.4.AND.PSTCO2.EQ.1) THEN
       CDIOXYL1=FINP*CPR(I)+FREC*COUT(I)+FBAK*F*CSU(I)+EPSL/EPS*VA*FLUXGAZ
       CDIOXYL1=CDIOXYL1/((1+FBAK)*F)

      ELSEIF (I.EQ.16.AND.PSTCO2.EQ.1) THEN
       CDIOXYL2=FINP*CPR(I)+FREC*COUT(I)+FBAK*F*CSU(I)+EPSL/EPS*VA*FLUXGAZ
       CDIOXYL2=CDIOXYL2/((1+FBAK)*F)

      ELSEIF (I.EQ.17.AND.PSTCO2.EQ.1) THEN
       CDIOXYL3=FINP*CPR(I)+FREC*COUT(I)+FBAK*F*CSU(I)+EPSL/EPS*VA*FLUXGAZ
       CDIOXYL3=CDIOXYL3/((1+FBAK)*F)

C-- PSEUDO-STEATDY STATE FOR H2O d/dt=0
      ELSEIF (I.EQ.6.AND.PSTH2O.EQ.1) THEN
       CEAUL=FINP*CPR(I)+FREC*COUT(I)+FBAK*F*CSU(I)+EPSL/EPS*VA*FLUXGAZ
       CEAUL=CEAUL/((1+FBAK)*F)

      ENDIF


C-------------------------------------------------------------------------
C--  DYNAMIC GAS EQUATION
C-
      IF (I.LE.10) THEN
        CSUG=Y(20+10+I)
        CPRG=CING(I)
        CG=Y(10+I)
        COUTG=Y(10*2*(N+1)+10+I)
      ELSE
        FCTG(I)=0.
        GOTO 50
      ENDIF

C--  NORMAL RELATION
      FCTG(I)=GINP*CPRG+GREC*COUTG-(1+FBAKPRIM)*G*CG+FBAKPRIM*G*CSUG-
```

```
      +EPSL/EPS*VA*FLUXGAZ
       FCTG(I)=FCTG(I)/(EPSG*VA/EPS)


C-- PSEUDO-STEADY STATE FOR OXYGENE d/dt=0
       IF (I.EQ.5.AND.PSTO2.EQ.1) THEN
          COXY=GINP*CPRG+GREC*COUTG+FBAKPRIM*G*CSUG-
      +EPSL/EPS*VA*FLUXGAZ
          COXY=COXY/((1+FBAKPRIM)*G)

C-- PSEUDO-STEADY STATE FOR CO2 d/dt=0
       ELSEIF (I.EQ.4.AND.PSTCO2.EQ.1) THEN
          CDIOXY=GINP*CPRG+GREC*COUTG+FBAKPRIM*G*CSUG-
      +EPSL/EPS*VA*FLUXGAZ
          CDIOXY=CDIOXY/((1+FBAKPRIM)*G)

C-- PSEUDO-STAEDY STATE FOR H2O d/dt=0
       ELSEIF (I.EQ.6.AND.PSTH2O.EQ.1) THEN
          Ceau=GINP*CPRG+GREC*COUTG+FBAKPRIM*G*CSUG-
      +EPSL/EPS*VA*FLUXGAZ
          CEAU=CEAU/((1+FBAKPRIM)*G)

       ENDIF

50     CONTINUE

C-------------------------------------------------------------------------
C--  SUM OF DISSOCIATED AND NON DISSOCIATED FORMS
C--  FOR INPUTS ON SKYMER SUB ROUTINE
C-
       IND=0
       INDG=10
       FCT(IND+1)=FCTL(1)+FCTL(13)
       FCT(INDG+1)=FCTG(1)+FCTG(13)
       FCT(IND+2)=FCTL(2)+FCTL(14)
       FCT(INDG+2)=FCTG(2)+FCTG(14)
       FCT(IND+3)=FCTL(3)+FCTL(15)
       FCT(INDG+3)=FCTG(3)+FCTG(15)
       FCT(IND+4)=FCTL(4)+FCTL(16)+FCTL(17)
       FCT(INDG+4)=FCTG(4)+FCTG(16)+FCTG(17)

C---Verification for psp
       IF (PSTCO2.EQ.1) THEN
          IF (DABS(FCT(INDG+4)/YIM1(INDG+4)).LE.1e-2) THEN
              YIM1(INDG+4)=CDIOXY
              FCT(INDG+4)=0.
          ENDIF
          IF (DABS(FCT(IND+4)/YIM1(IND+4)).LE.1e-2) THEN
              YIM1(IND+4)=CDIOXYL1+CDIOXYL2+CDIOXYL3
              FCT(IND+4)=0.
          ENDIF

       ENDIF

       FCT(IND+5)=FCTL(5)
       FCT(INDG+5)=FCTG(5)

C---Verification for psp
       IF (PSTO2.EQ.1) THEN
          IF (DABS(FCT(INDG+5)/YIM1(INDG+5)).LE.1e-2) THEN
              YIM1(INDG+5)=COXY
              FCT(INDG+5)=0.
          ENDIF
          IF (DABS(FCT(IND+5)/YIM1(IND+5)).LE.1e-2) THEN
              YIM1(IND+5)=COXYL
              FCT(IND+5)=0.
          ENDIF

       ENDIF

       FCT(IND+6)=FCTL(6)
       FCT(INDG+6)=FCTG(6)

C---Verification for psp
       IF (PSTH2O.EQ.1) THEN
          IF (DABS(FCT(INDG+6)/YIM1(INDG+6)).LE.1e-2) THEN
```

```
            YIM1(INDG+6)=CEAU
            FCT(INDG+6)=0.
         ENDIF
         IF (DABS(FCT(IND+6)/YIM1(IND+6)).LE.1e-2) THEN
            YIM1(IND+6)=CEAUL
            FCT(IND+6)=0.
         ENDIF

      ENDIF

      FCT(IND+7)=FCTL(7)+FCTL(18)+FCTL(19)
      FCT(INDG+7)=FCTG(7)+FCTG(18)+FCTG(19)
      FCT(IND+8)=FCTL(8)+FCTL(20)+FCTL(21)+FCTL(22)
      FCT(INDG+8)=FCTG(8)+FCTG(20)+FCTG(21)+FCTG(22)
      FCT(IND+9)=FCTL(9)
      FCT(INDG+9)=FCTG(9)
      FCT(IND+10)=FCTL(10)
      FCT(INDG+10)=FCTG(10)


C-------------------------------------------------------------------------
C--   Theoretical evolution of H+ for absence of neutralisation
C-
      CPRH=10**(-PHINI)
      COUTH=Y(10*(2*(N+2))+2*N+N+2)
      CSUH=Y(10*(2*(N+2))+2*N+2)
      CH=Y(10*(2*(N+2))+2*N+1)
      FCTLH=FINP*CPRH+FREC*COUTH-(1+FBAK)*F*CH+FBAK*F*CSUH
      FCTLH=FCTLH/(EPSL*VA/EPS)
      FCT((10*(2*(N+2))+2*N+1))=FCTLH


C/////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////
C                          SEGMENT N OF THE BED (UNIT B)
C/////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////

C-------------------------------------------------------------------------
C--   NSEC=INDEX FOR THE SEGMENT NUMBER OF THE BED
C-
      DO 70 NSEG=1,N


C=== CONCENTRATIONS IN SEGMENT N-1 ======================================

C-------------------------------------------------------------------------
C--   INITIALISATION FROM SKYKMER VARIABLES
C-
      IND=10*2*(NSEG-1)
      CTNH3=Y(IND+1)
      CTNO3H=Y(IND+2)
      CTNO2H=Y(IND+3)
      CTCO2=Y(IND+4)
      CTO2=Y(IND+5)
      CTH2O=Y(IND+6)
      CTH2SO=Y(IND+7)
      CTH3PO=Y(IND+8)
      CTNS=Y(IND+9)
      CTNB=Y(IND+10)


C-------------------------------------------------------------------------
C--   PH  EQUILIBRIA CALCULATION FOR THE FIXED PHINI
C-

      CPR(11)=10**(-PHINI)
      CPR(12)=10**(PHINI-14)

      CPR(1)=CTNH3*CPR(12)/(KA(1)+CPR(12))
      CPR(13)=CTNH3-CPR(1)

      CPR(2)=CTNO3H*CPR(11)/(KA(2)+CPR(11))
      CPR(14)=CTNO3H-CPR(2)

      CPR(3)=CTNO2H*CPR(11)/(KA(3)+cpr(11))
```

```
      CPR(15)=CTNO2H-CPR(3)

      CPR(16)=CTCO2/(10**(-PHINI)/KA(4)+1+KA(16)/(10**(-PHINI)))
      CPR(17)=CPR(16)*KA(16)/10**(-PHINI)
      CPR(4)=CTCO2-CPR(16)-CPR(17)

      CPR(18)=CTH2SO/(10**(-PHINI)/KA(7)+1+KA(18)/(10**(-PHINI)))
      CPR(19)=CPR(18)*KA(18)/(10**(-PHINI))
      CPR(7)=CTH2SO-CPR(18)-CPR(19)


      CPR(21)=CTH3PO/((10**(-2*PHINI)/(KA(8)*KA(20)))+
     +10**(-PHINI)/KA(20)+1+KA(21)/10**(-PHINI))
      CPR(22)=KA(21)*CPR(21)/10**(-PHINI)
      CPR(20)=CPR(21)*10**(-PHINI)/KA(20)
      CPR(8)=CTH3PO-CPR(21)-CPR(20)-CPR(22)

      CPR(5)=CTO2
      CPR(6)=CTH2O
      CPR(9)=CTNS
      CPR(10)=CTNB



C=== CONCENTRATIONS IN SEGMENT N+1 ========================================

C-------------------------------------------------------------------------
C--   INITIALISATION FROM SKYKMER VARIABLES
C-
      IND=10*2*(NSEG+1)
      CTNH3=Y(IND+1)
      CTNO3H=Y(IND+2)
      CTNO2H=Y(IND+3)
      CTCO2=Y(IND+4)
      CTO2=Y(IND+5)
      CTH2O=Y(IND+6)
      CTH2SO=Y(IND+7)
      CTH3PO=Y(IND+8)
      CTNS=Y(IND+9)
      CTNB=Y(IND+10)


C-------------------------------------------------------------------------
C--   PH  EQUILIBRIA CALCULATION FOR THE FIXED PHINI
C-
      CSU(11)=10**(-PHINI)
      CSU(12)=10**(PHINI-14)

      CSU(1)=CTNH3*CSU(12)/(KA(1)+csu(12))
      CSU(13)=CTNH3-CSU(1)

      CSU(2)=CTNO3H*CSU(11)/(KA(2)+csu(11))
      CSU(14)=CTNO3H-CSU(2)

      CSU(3)=CTNO2H*CSU(11)/(KA(3)+csu(11))
      CSU(15)=CTNO2H-CSU(3)

      CSU(16)=CTCO2/(10**(-PHINI)/KA(4)+1+KA(16)/10**(-PHINI))
      CSU(17)=CSU(16)*KA(16)/10**(-PHINI)
      CSU(4)=CTCO2-CSU(16)-CSU(17)

      CSU(18)=CTH2SO/(10**(-PHINI)/KA(7)+1+KA(18)/10**(-PHINI))
      CSU(19)=CSU(18)*KA(18)/10**(-PHINI)
      CSU(7)=CTH2SO-CSU(18)-CSU(19)


      CSU(21)=CTH3PO/((10**(-2*PHINI)/(KA(8)*KA(20)))+
     +10**(-PHINI)/KA(20)+1+KA(21)/10**(-PHINI))
      CSU(22)=KA(21)*CSU(21)/10**(-PHINI)
      CSU(20)=CSU(21)*10**(-PHINI)/KA(20)
      CSU(8)=CTH3PO-CSU(21)-CSU(20)-CSU(22)

      CSU(5)=CTO2
      CSU(6)=CTH2O
      CSU(9)=CTNS
```

```
      CSU(10)=CTNB



C== CONCENTRATIONS IN SEGMENT N ============================================

C-------------------------------------------------------------------------------
C--   INITIALISATION FROM SKYKMER VARIABLES
C-
      IND=10*2*(NSEG)
      CTNH3=Y(IND+1)
      CTNO3H=Y(IND+2)
      CTNO2H=Y(IND+3)
      CTCO2=Y(IND+4)
      CTO2=Y(IND+5)
      CTH2O=Y(IND+6)
      CTH2SO=Y(IND+7)
      CTH3PO=Y(IND+8)
      CTNS=Y(IND+9)
      CTNB=Y(IND+10)

C-------------------------------------------------------------------------------
C--   PH  EQUILIBRIA CALCULATION FOR THE FIXED PHINI
C-
      C(11)=10**(-PHINI)
      C(12)=10**(PHINI-14)

      C(1)=CTNH3*C(12)/(KA(1)+c(12))
      C(13)=CTNH3-C(1)

      C(2)=CTNO3H*C(11)/(KA(2)+c(11))
      C(14)=CTNO3H-C(2)

      C(3)=CTNO2H*C(11)/(KA(3)+c(11))
      C(15)=CTNO2H-C(3)

      C(16)=CTCO2/(10**(-PHINI)/KA(4)+1+KA(16)/10**(-PHINI))
      C(17)=C(16)*KA(16)/10**(-PHINI)
      C(4)=CTCO2-C(16)-C(17)

      C(18)=CTH2SO/(10**(-PHINI)/KA(7)+1+KA(18)/10**(-PHINI))
      C(19)=C(18)*KA(18)/10**(-PHINI)
      C(7)=CTH2SO-C(18)-C(19)


      C(21)=CTH3PO/((10**(-2*PHINI)/(KA(8)*KA(20)))+
     +10**(-PHINI)/KA(20)+1+KA(21)/10**(-PHINI))
      C(22)=KA(21)*C(21)/10**(-PHINI)
      C(20)=C(21)*10**(-PHINI)/KA(20)
      C(8)=CTH3PO-C(21)-C(20)-C(22)

      C(5)=CTO2
      C(6)=CTH2O
      C(9)=CTNS
      C(10)=CTNB


C===============================================================================
C===   CALCULATION OF THE BIOFILM PROFILE
C===   (Not Implemented in this model - No biofilm transfer limitation
C===============================================================================
      CNS=Y(10*(2*(N+2))+NSEG)
      CNB=Y(10*(2*(N+2))+N+NSEG)



C===============================================================================
C==   CALCUL DES CINETIQUES BIOLOGIQUES
C===============================================================================

C-------------------------------------------------------------------------------
C-- LIMITATIONS
```

```
C-- Limiting and inhibitory substrates and products
C-
      SUBSLIMNS=1
      SUBSLIMNB=1

      DO 100 I=1,CORP
         IF (STO(1,I).LT.0) THEN
         IF (KS(1,I).NE.0..OR.KI(1,I).LE.1000) THEN
            if (c(i).lt.1e-15) then
               c(i)=0.
            endif
          SUBSLIMNS=SUBSLIMNS*C(I)/((KS(1,I)+C(I))*(1+C(I)/KI(1,I)))
         ENDIF
         ELSEIF (STO(1,I).NE.0) THEN
         IF (KS(1,I).NE.0..OR.KI(1,I).LE.1000) THEN
            if (c(i).lt.1e-15) then
               c(i)=0.
            endif
          SUBSLIMNS=SUBSLIMNS/(1+C(I)/KI(1,I))
         ENDIF
         ENDIF

         IF (STO(2,I).LT.0.) THEN
         IF (KS(2,I).NE.0..OR.KI(2,I).LE.1000) THEN
            if (c(i).lt.1e-15) then
                c(i)=0.
             endif
          SUBSLIMNB=SUBSLIMNB*C(I)/((KS(2,I)+C(I))*(1+C(I)/KI(2,I)))
         ENDIF
         ELSEIF (STO(2,I).NE.0.) THEN
         IF (KS(2,I).NE.0..OR.KI(2,I).LE.1000) THEN
            if (c(i).lt.1e-15) then
                c(i)=0.
             endif
          SUBSLIMNB=SUBSLIMNB/(1+C(I)/KI(2,I))
         ENDIF
         ENDIF
100   CONTINUE

C-----------------------------------------------------------------------
C--Calculation of the growth -- Warning! BE CARE OF THE REFERENCE SUBSTRATE
C-
C          1-NH3    ---> refernce substrate for Ns
C          2-HNO3
C          3-HNO2
C          4-CO2
C          5-O2
C          6-H2O
C          7-H2SO4
C          8-H3PO4
C          9-BIOMASSE LIBRE NS
C          10-BIOMASSE LIBRE NB
C          11-H+
C          12-OH-
C          13-NH4+
C          14-NO3-
C          15-NO2-   ---> Reference substrate for Nb
C          16-HCO3-
C          17-CO32-
C          18-HSO4-
C          19-SO42-
C          20-H2PO4-
C          21-HPO4-
C          22-PO42-

       RXNS=MUMAX(1)*SUBSLIMNS*CNS+
      +(SUBSLIMNS-1)*(STO(1,9)*PMBIO/(-1.*STO(1,1)))*MUMAX(3)*CNS

       RXNB=MUMAX(2)*SUBSLIMNB*CNB+
      +(SUBSLIMNB-1)*(STO(2,10)*PMBIO/(-1.*STO(2,15)))*MUMAX(4)*CNB



C-----------------------------------------------------------------------
C-- Substrate and products kinetics --
C-- Warning! Be CARE OF THE REFERENCE SUBSTRATE
```

```
C-
      DO 101 I=1,8
         RSNS(I)=(STO(1,I)/(PMBIO*STO(1,9)))*RXNS+
     +STO(3,I)/(-1.*STO(3,1))*MUMAX(3)*CNS

         RSNB(I)=(STO(2,I)/(PMBIO*STO(2,10)))*RXNB+
     +STO(4,I)/(-1.*STO(4,15))*MUMAX(4)*CNB

101   CONTINUE

      DO 103 I=11,CORP
         RSNS(I)=(STO(1,I)/(PMBIO*STO(1,9)))*RXNS+
     +STO(3,I)/(-1.*STO(3,1))*MUMAX(3)*CNS

         RSNB(I)=(STO(2,I)/(PMBIO*STO(2,10)))*RXNB+
     +STO(4,I)/(-1.*STO(4,15))*MUMAX(4)*CNB

103   CONTINUE


C=======================================================================
C===  CALCULCULATION OF THE BIOMASS RELEASED FROM BEADS
C=======================================================================
      RSNS(9)=BWO*RXNS
      RSNB(10)=BWO*RXNB



C=======================================================================
C    GAS AND LIQUID DYNAMIC EQUATION ON SEGMENT N OF THE BED
C=======================================================================


      DO 200 I=1,CORP

C-----------------------------------------------------------------------
C--  CALCULATION OF THE TRANSFER BIOFILM-LIQUIDE
C--  Not implemented in this model : FluXBIO=RS
C-
      FLUXBIO=RSNS(I)+RSNB(I)

C-----------------------------------------------------------------------
C-- CALCULATION OF THE RESPIRATORY RATE IN BIOFILM IN MOL/ LITER Biomasse h
C-
      IF (I.EQ.5) THEN
          RNBIO(NSEG)=RSNS(I)/CNS+RSNB(I)/CNB
          RNBIO(NSEG)=RNBIO(NSEG)*270.27
      ENDIF

C-----------------------------------------------------------------------
C--  Gas-Liquid transfer reaction
C-
C- WARNING !: CSAT(I) IS NOT the concentration at saturation but the partition
C- coefficient (KI) stored in files
C- SATU IS the STAURATION concnetration calculated from KI(=CSAT)
C- For gases, the perfect gas law is used for the calculation

      IF (CSAT(I).NE.0.) THEN
        SATU=(Y(10*2*(NSEG)+10+I)*(8.314*(273.15+T)/(P*101.3)))/CSAT(I)
        SATU=SATU*C(6)
        FLUXGAZ=KLGAZ(I)*ASPGAZ*(SATU-C(I))

C--For water (index I=6) the flux liquid-->gas is calculated
         IF (I.EQ.6) THEN
            EAUGAZ=csat(6)/(8.314*(273.15+T)/(P*1.013E2))
            FLUXEAU=KLGAZ(I)*ASPGAZ*(EAUGAZ-Y(10*2*(NSEG)+10+I))
            FLUXGAZ=-FLUXEAU*EPSG/EPSL
         ENDIF

C-- Reduction of low values
         IF (DABS(FLUXGAZ).LT.1e-16) FLUXGAZ=0.
      ELSE
          FLUXGAZ=0.
      ENDIF
```

```
C-----------------------------------------------------------------------
C--   DYNAMIC EQUATION FOR LIQUID PHASE
C-
C-    For value lower than 1e-15 We assumed a Concentration nul
C--   Modification Include in version 2.2
      IF (C(I).LE.1e-15) C(I)=0.
      IF (CPR(I).LE.1e-15) CPR(I)=0.
      IF (CSU(I).LE.1e-15) CSU(I)=0.

C--   For the water the problem is different because concentration
C--   cannot exceed 55.5555 mol/l
C--   If production of water ---> modification of volume.
C--   The effect on volume is neglected
C--   Concentration of water is limited to 55.5555 mol/l
C--   Modification included in version 2.2
C      IF (C(6).GT.55.55555555) C(6)=55.55555555
C--   Modification supressed in version 2.3


      FCTL(I)=(1+FBAK)*F*CPR(I)+FBAK*F*CSU(I)-(1+FBAK)*F*C(I)-
     +FBAK*F*C(I)+EPSL*VB/N*FLUXGAZ+EPSL*VB/N*FLUXBIO
      FCTL(I)=FCTL(I)/(EPSL*VB/N)

C-- Liquid relation for pseudo steady state
C-- PSEUDO-STEADY STATE FOR OXYGENE d/dt=0
      IF (I.EQ.5.AND.PSTO2.EQ.1) THEN
       COXYL=(1+FBAK)*F*CPR(I)+FBAK*F*CSU(I)-
     +FBAK*F*C(I)+EPSL*VB/N*FLUXGAZ+EPSL*VB/N*FLUXBIO
       COXYL=COXYL/((1+FBAK)*F)

C-- PSEUDO-STEADY STATE FOR CO2 d/dt=0
      ELSEIF (I.EQ.4.AND.PSTCO2.EQ.1) THEN
       CDIOXYL1=(1+FBAK)*F*CPR(I)+FBAK*F*CSU(I)-
     +FBAK*F*C(I)+EPSL*VB/N*FLUXGAZ+EPSL*VB/N*FLUXBIO
       CDIOXYL1=CDIOXYL1/((1+FBAK)*F)

      ELSEIF (I.EQ.16.AND.PSTCO2.EQ.1) THEN
       CDIOXYL2=(1+FBAK)*F*CPR(I)+FBAK*F*CSU(I)-
     +FBAK*F*C(I)+EPSL*VB/N*FLUXGAZ+EPSL*VB/N*FLUXBIO
       CDIOXYL2=CDIOXYL2/((1+FBAK)*F)

      ELSEIF (I.EQ.17.AND.PSTCO2.EQ.1) THEN
       CDIOXYL3=(1+FBAK)*F*CPR(I)+FBAK*F*CSU(I)-
     +FBAK*F*C(I)+EPSL*VB/N*FLUXGAZ+EPSL*VB/N*FLUXBIO
       CDIOXYL3=CDIOXYL3/((1+FBAK)*F)

C-- PSEUDO-STEADY STATE FOR d/dt=0
      ELSEIF (I.EQ.6.AND.PSTH2O.EQ.1) THEN
       CEAUL=(1+FBAK)*F*CPR(I)+FBAK*F*CSU(I)-
     +FBAK*F*C(I)+EPSL*VB/N*FLUXGAZ+EPSL*VB/N*FLUXBIO
       CEAUL=CEAUL/((1+FBAK)*F)

      ENDIF

C-----------------------------------------------------------------------
C--  DYNAMIC GAZ  EQUATION
C-
      IF (I.LE.10) THEN
        CSUG=Y(10*2*(NSEG+1)+10+I)
        CPRG=Y(10*2*(NSEG-1)+10+I)
        CG=Y(10*2*(NSEG)+10+I)
      ELSE
        FCTG(I)=0.
        GOTO 200
      ENDIF

C-- NORMAL RELATION

      FCTG(I)=(1+FBAKPRIM)*G*CPRG+FBAKPRIM*G*CSUG-
     +(1+2*FBAKPRIM)*G*CG-EPSL*VB/N*FLUXGAZ
      FCTG(I)=FCTG(I)/(EPSG*VB/N)


C-- PSEUDO-STEADY STATE FOR OXYGENE d/dt=0
      IF (I.EQ.5.AND.PSTO2.EQ.1) THEN
         COXY=(1+FBAKPRIM)*G*CPRG+FBAKPRIM*G*CSUG-
```

```
     +EPSL*VB/N*FLUXGAZ
         COXY=COXY/((1+2*FBAKPRIM)*G)

C-- PSEUDO-STEADY STATE FOR CO2 d/dt=0
      ELSEIF (I.EQ.4.AND.PSTCO2.EQ.1) THEN
         CDIOXY=(1+FBAKPRIM)*G*CPRG+FBAKPRIM*G*CSUG-
     +EPSL*VB/N*FLUXGAZ
         CDIOXY=CDIOXY/((1+2*FBAKPRIM)*G)

C-- PSEUDO-STEADY STATE FOR H2O d/dt=0
      ELSEIF (I.EQ.6.AND.PSTH2O.EQ.1) THEN
         CEAU=(1+FBAKPRIM)*G*CPRG+FBAKPRIM*G*CSUG-
     +EPSL*VB/N*FLUXGAZ
         CEAU=CEAU/((1+2*FBAKPRIM)*G)

      ENDIF

200   CONTINUE

C-------------------------------------------------------------------------
C--  BIOMASS-BIOFILM GROWTH EQUATION
C-
      FCT(10*(2*(N+2))+NSEG)=RXNS*(1-BWO)
      FCT(10*(2*(N+2))+N+NSEG)=RXNB*(1-BWO)

C-------------------------------------------------------------------------
C--  SUM OF DISSOCIATED AND NON DISSOCIATED FORMS
C--  FOR INPUTS ON SKYMER SUB ROUTINE
C-

      INDG=10*2*(NSEG)+10
      FCT(IND+1)=FCTL(1)+FCTL(13)
      FCT(INDG+1)=FCTG(1)+FCTG(13)
      FCT(IND+2)=FCTL(2)+FCTL(14)
      FCT(INDG+2)=FCTG(2)+FCTG(14)
      FCT(IND+3)=FCTL(3)+FCTL(15)
      FCT(INDG+3)=FCTG(3)+FCTG(15)
      FCT(IND+4)=FCTL(4)+FCTL(16)+FCTL(17)
      FCT(INDG+4)=FCTG(4)+FCTG(16)+FCTG(17)

C---Verification of psp
      IF (PSTCO2.EQ.1) THEN
         IF (DABS(FCT(INDG+4)/YIM1(INDG+4)).LE.1e-2) THEN
             YIM1(INDG+4)=CDIOXY
             FCT(INDG+4)=0.
         ENDIF
         IF (DABS(FCT(IND+4)/YIM1(IND+4)).LE.1e-2) THEN
             YIM1(IND+4)=CDIOXYL1+CDIOXYL2+CDIOXYL3
             FCT(IND+4)=0.
         ENDIF

      ENDIF

      FCT(IND+5)=FCTL(5)
      FCT(INDG+5)=FCTG(5)

C---Verification  of psp
      IF (PSTO2.EQ.1) THEN
         IF (DABS(FCT(INDG+5)/YIM1(INDG+5)).LE.1e-2) THEN
             YIM1(INDG+5)=COXY
             FCT(INDG+5)=0.
         ENDIF
         IF (DABS(FCT(IND+5)/YIM1(IND+5)).LE.1e-2) THEN
             YIM1(IND+5)=COXYL
             FCT(IND+5)=0.
         ENDIF

      ENDIF

      FCT(IND+6)=FCTL(6)
      FCT(INDG+6)=FCTG(6)

C---Verification  of psp
      IF (PSTH2O.EQ.1) THEN
         IF (DABS(FCT(INDG+6)/YIM1(INDG+6)).LE.1e-2) THEN
             YIM1(INDG+6)=CEAU
```

```
            FCT(INDG+6)=0.
         ENDIF
         IF (DABS(FCT(IND+6)/YIM1(IND+6)).LE.1e-2) THEN
            YIM1(IND+6)=CEAUL
            FCT(IND+6)=0.
         ENDIF


      ENDIF

      FCT(IND+7)=FCTL(7)+FCTL(18)+FCTL(19)
      FCT(INDG+7)=FCTG(7)+FCTG(18)+FCTG(19)
      FCT(IND+8)=FCTL(8)+FCTL(20)+FCTL(21)+FCTL(22)
      FCT(INDG+8)=FCTG(8)+FCTG(20)+FCTG(21)+FCTG(22)
      FCT(IND+9)=FCTL(9)
      FCT(INDG+9)=FCTG(9)
      FCT(IND+10)=FCTL(10)
      FCT(INDG+10)=FCTG(10)

C-------------------------------------------------------------------------
C--   Theoretical evolution of H+ with neutralisation
C-
      CPRH=Y(10*(2*(N+2))+2*N+NSEG-1+1)
      CSUH=Y(10*(2*(N+2))+2*N+NSEG+1+1)
      CH=Y(10*(2*(N+2))+2*N+NSEG+1)
      FCTLH=(1+FBAK)*F*CPRH+FBAK*F*CSUH-(1+FBAK)*F*CH-
     +FBAK*F*CH+EPSL*VB/N*(RSNS(11)+RSNB(11)-RSNS(12)-RSNB(12))
      FCTLH=FCTLH/(EPSL*VB/N)
      FCT((10*(2*(N+2))+2*N+NSEG+1))=FCTLH


70    CONTINUE


C/////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////
C                        UNITE C OF THE COLUMN
C/////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////


C=========================================================================
C== CONCENTRATIONS IN THE TERMINAL SEGMENT N OF UNIT B
C=========================================================================

C-------------------------------------------------------------------------
C--   INITIALISATION FROM SKYKMER VARIABLES
C-
      IND=10*2*(N)
      CTNH3=Y(IND+1)
      CTNO3H=Y(IND+2)
      CTNO2H=Y(IND+3)
      CTCO2=Y(IND+4)
      CTO2=Y(IND+5)
      CTH2O=Y(IND+6)
      CTH2SO=Y(IND+7)
      CTH3PO=Y(IND+8)
      CTNS=Y(IND+9)
      CTNB=Y(IND+10)


C-------------------------------------------------------------------------
C--   PH  EQUILIBRIA CALCULATION FOR THE FIXED PHINI
C-
      CPR(11)=10**(-PHINI)
      CPR(12)=10**(PHINI-14)

      CPR(1)=CTNH3*CPR(12)/(KA(1)+CPR(12))
      CPR(13)=CTNH3-CPR(1)

      CPR(2)=CTNO3H*CPR(11)/(KA(2)+CPR(11))
      CPR(14)=CTNO3H-CPR(2)

      CPR(3)=CTNO2H*CPR(11)/(KA(3)+cpr(11))
      CPR(15)=CTNO2H-CPR(3)
```

```
      CPR(16)=CTCO2/(10**(-PHINI)/KA(4)+1+KA(16)/(10**(-PHINI)))
      CPR(17)=CPR(16)*KA(16)/10**(-PHINI)
      CPR(4)=CTCO2-CPR(16)-CPR(17)

      CPR(18)=CTH2SO/(10**(-PHINI)/KA(7)+1+KA(18)/(10**(-PHINI)))
      CPR(19)=CPR(18)*KA(18)/(10**(-PHINI))
      CPR(7)=CTH2SO-CPR(18)-CPR(19)


      CPR(21)=CTH3PO/((10**(-2*PHINI)/(KA(8)*KA(20)))+
     +10**(-PHINI)/KA(20)+1+KA(21)/10**(-PHINI))
      CPR(22)=KA(21)*CPR(21)/10**(-PHINI)
      CPR(20)=CPR(21)*10**(-PHINI)/KA(20)
      CPR(8)=CTH3PO-CPR(21)-CPR(20)-CPR(22)

      CPR(5)=CTO2
      CPR(6)=CTH2O
      CPR(9)=CTNS
      CPR(10)=CTNB


C=======================================================================
C== CONCENTRATIONS IN UNIT C
C=======================================================================

C-----------------------------------------------------------------------
C--   INITIALISATION FROM SKYKMER VARIABLES
C-
      IND=10*(2*(N+1))
      CTNH3=Y(IND+1)
      CTNO3H=Y(IND+2)
      CTNO2H=Y(IND+3)
      CTCO2=Y(IND+4)
      CTO2=Y(IND+5)
      CTH2O=Y(IND+6)
      CTH2SO=Y(IND+7)
      CTH3PO=Y(IND+8)
      CTNS=Y(IND+9)
      CTNB=Y(IND+10)


C-----------------------------------------------------------------------
C--   PH  EQUILIBRIA CALCULATION FOR THE FIXED PHINI
C-
      C(11)=10**(-PHINI)
      C(12)=10**(PHINI-14)

      C(1)=CTNH3*C(12)/(KA(1)+c(12))
      C(13)=CTNH3-C(1)

      C(2)=CTNO3H*C(11)/(KA(2)+c(11))
      C(14)=CTNO3H-C(2)

      C(3)=CTNO2H*C(11)/(KA(3)+c(11))
      C(15)=CTNO2H-C(3)

      C(16)=CTCO2/(10**(-PHINI)/KA(4)+1+KA(16)/10**(-PHINI))
      C(17)=C(16)*KA(16)/10**(-PHINI)
      C(4)=CTCO2-C(16)-C(17)

      C(18)=CTH2SO/(10**(-PHINI)/KA(7)+1+KA(18)/10**(-PHINI))
      C(19)=C(18)*KA(18)/10**(-PHINI)
      C(7)=CTH2SO-C(18)-C(19)


      C(21)=CTH3PO/((10**(-2*PHINI)/(KA(8)*KA(20)))+
     +10**(-PHINI)/KA(20)+1+KA(21)/10**(-PHINI))
      C(22)=KA(21)*C(21)/10**(-PHINI)
      C(20)=C(21)*10**(-PHINI)/KA(20)
      C(8)=CTH3PO-C(21)-C(20)-C(22)

      C(5)=CTO2
      C(6)=CTH2O
      C(9)=CTNS
      C(10)=CTNB
```

```
C=======================================================================
C    DIFERENTIALS EQUATIONS FOR GASES AND LIQUIDS IN UNITE C
C=======================================================================


      DO 300 I=1,CORP

C-----------------------------------------------------------------------
C--  Gas-Liquid transfer reaction
C-
C- WARNING !: CSAT(I) IS NOT the concentration at saturation but the partition
C- coefficient (KI) stored in files
C- SATU IS the STAURATION concnetration calculated from KI(=CSAT)
C- For gases, the perfect gas law is used for the calculation

      IF (CSAT(I).NE.0.) THEN
      SATU=(Y(10*2*(N+1)+10+I)*(8.314*(273.15+T)/(P*101.3)))/CSAT(I)
      SATU=SATU*C(6)
      FLUXGAZ=KLGAZ(I)*ASPGAZ*(SATU-C(I))

C--For water (index I=6) the flux liquid-->gas is calculated
      IF (I.EQ.6) THEN
         EAUGAZ=csat(6)/(8.314*(273.15+T)/(P*1.013E2))
         FLUXEAU=KLGAZ(I)*ASPGAZ*(EAUGAZ-Y(10*2*(N+1)+10+I))
         FLUXGAZ=-FLUXEAU*EPSG/EPSL
      ENDIF

C-- Reduction of low values
         IF (DABS(FLUXGAZ).LT.1e-16) FLUXGAZ=0.

      ELSE
         FLUXGAZ=0.
      ENDIF

C-----------------------------------------------------------------------
C--   DYNAMIC EQUATION FOR LIQUID PHASE
C-
C-    For value lower than 1e-15 We assumed a Concentration nul
C--   Modification Include in version 2.2
      IF (C(I).LE.1e-15) C(I)=0.
      IF (CPR(I).LE.1e-15) CPR(I)=0.

C--   For the water the problem is different because concentration
C--   cannot exceed 55.5555 mol/l
C--   If production of water ---> modification of volume.
C--   The effect on volume is neglected
C--   Concentration of water is limited to 55.5555 mol/l
C--   Modification included in version 2.2
C      IF (C(6).GT.55.55555555) C(6)=55.55555555
C--   Modification supressed in version 2.3


      FCTL(I)=(1+FBAK)*F*CPR(I)-FBAK*F*C(I)-FREC*C(I)-FOUT*C(I)+
     +EPSL/EPS*VC*FLUXGAZ
      FCTL(I)=FCTL(I)/(EPSL*VC/EPS)

C-- liquid  Relation in pseudo steady state
C-- PSEUDO-STEADY STATE FOR OXYGENE d/dt=0
      IF (I.EQ.5.AND.PSTO2.EQ.1) THEN
       COXYL=(1+FBAK)*F*CPR(I)-FOUT*C(I)+EPSL/EPS*VC*FLUXGAZ
       COXYL=COXYL/(FBAK*F+FREC+FOUT)

C-- PSEUDO-STEADY STATE FOR CO2 d/dt=0
      ELSEIF (I.EQ.4.AND.PSTCO2.EQ.1) THEN
       CDIOXYL1=(1+FBAK)*F*CPR(I)-FOUT*C(I)+EPSL/EPS*VC*FLUXGAZ
       CDIOXYL1=CDIOXYL1/(FBAK*F+FREC+FOUT)

      ELSEIF (I.EQ.16.AND.PSTCO2.EQ.1) THEN
       CDIOXYL2=(1+FBAK)*F*CPR(I)-FOUT*C(I)+EPSL/EPS*VC*FLUXGAZ
       CDIOXYL2=CDIOXYL2/(FBAK*F+FREC+FOUT)

      ELSEIF (I.EQ.17.AND.PSTCO2.EQ.1) THEN
       CDIOXYL3=(1+FBAK)*F*CPR(I)-FOUT*C(I)+EPSL/EPS*VC*FLUXGAZ
```

```
       CDIOXYL3=CDIOXYL3/(FBAK*F+FREC+FOUT)

C-- PSEUDO-STEADY STATE FOR H2O d/dt=0
      ELSEIF (I.EQ.6.AND.PSTH2O.EQ.1) THEN
       CEAUL=(1+FBAK)*F*CPR(I)-FOUT*C(I)+EPSL/EPS*VC*FLUXGAZ
       CEAUL=CEAUL/(FBAK*F+FREC+FOUT)

      ENDIF


C--------------------------------------------------------------------------
C--  Dynamic equation for GAS
C-
      IF (I.LE.10) THEN
        CPRG=Y(10*2*(N)+10+I)
        CG=Y(10*2*(N+1)+10+I)
      ELSE
        FCTG(I)=0.
        GOTO 300
      ENDIF

C-- Normal relation

       FCTG(I)=(1+FBAKPRIM)*G*CPRG-FBAKPRIM*G*CG-GREC*CG-GOUT*CG-
     +EPSL/EPS*VC*FLUXGAZ
       FCTG(I)=FCTG(I)/(EPSG*VC/EPS)


C-- PSEUDO-STEADY STATE FOR OXYGENE d/dt=0
      IF (I.EQ.5.AND.PSTO2.EQ.1) THEN
         COXY=(1+FBAKPRIM)*G*CPRG-EPSL/EPS*VC*FLUXGAZ
         COXY=COXY/(FBAKPRIM*G+GREC+GOUT)

C-- PSEUDO-STEADY STATE FOR CO2 d/dt=0
      ELSEIF (I.EQ.4.AND.PSTCO2.EQ.1) THEN
         CDIOXY=(1+FBAKPRIM)*G*CPRG-EPSL/EPS*VC*FLUXGAZ
         CDIOXY=CDIOXY/(FBAKPRIM*G+GREC+GOUT)

C-- PSEUDO-STEADY STATE FOR H2O d/dt=0
      ELSEIF (I.EQ.6.AND.PSTH2O.EQ.1) THEN
         CEAU=(1+FBAKPRIM)*G*CPRG-EPSL/EPS*VC*FLUXGAZ
         CEAU=CEAU/(FBAKPRIM*G+GREC+GOUT)

      ENDIF

300   CONTINUE

C--------------------------------------------------------------------------
C--  SUM OF DISSOCIATED AND NON DISSOCIATED FORMS
C--  FOR INPUTS ON SKYMER SUB ROUTINE
C-

       INDG=10*2*(N+1)+10
       FCT(IND+1)=FCTL(1)+FCTL(13)
       FCT(INDG+1)=FCTG(1)+FCTG(13)
       FCT(IND+2)=FCTL(2)+FCTL(14)
       FCT(INDG+2)=FCTG(2)+FCTG(14)
       FCT(IND+3)=FCTL(3)+FCTL(15)
       FCT(INDG+3)=FCTG(3)+FCTG(15)
       FCT(IND+4)=FCTL(4)+FCTL(16)+FCTL(17)
       FCT(INDG+4)=FCTG(4)+FCTG(16)+FCTG(17)


C---Verification for psp
      IF (PSTCO2.EQ.1) THEN
         IF (DABS(FCT(INDG+4)/YIM1(INDG+4)).LE.1e-2) THEN
             YIM1(INDG+4)=CDIOXY
             FCT(INDG+4)=0.
         ENDIF
         IF (DABS(FCT(IND+4)/YIM1(IND+4)).LE.1e-2) THEN
             YIM1(IND+4)=CDIOXYL1+CDIOXYL2+CDIOXYL3
             FCT(IND+4)=0.
         ENDIF

       ENDIF
```

```
      FCT(IND+5)=FCTL(5)
      FCT(INDG+5)=FCTG(5)

C---Verification  for psp
      IF (PSTO2.EQ.1) THEN
         IF (DABS(FCT(INDG+5)/YIM1(INDG+5)).LE.1e-2) THEN
             YIM1(INDG+5)=COXY
             FCT(INDG+5)=0.
         ENDIF
         IF (DABS(FCT(IND+5)/YIM1(IND+5)).LE.1e-2) THEN
             YIM1(IND+5)=COXYL
             FCT(IND+5)=0.
         ENDIF

      ENDIF

      FCT(IND+6)=FCTL(6)
      FCT(INDG+6)=FCTG(6)

C---Verification  for psp
      IF (PSTH2O.EQ.1) THEN
         IF (DABS(FCT(INDG+6)/YIM1(INDG+6)).LE.1e-2) THEN
             YIM1(INDG+6)=CEAU
             FCT(INDG+6)=0.
         ENDIF
         IF (DABS(FCT(IND+6)/YIM1(IND+6)).LE.1e-2) THEN
             YIM1(IND+6)=CEAUL
             FCT(IND+6)=0.
         ENDIF


      ENDIF

C-----------------------------------------------------------------------
C--  SUM OF DISSOCIATED AND NON DISSOCIATED FORMS
C--  FOR INPUTS ON SKYMER SUB ROUTINE
C-
      FCT(IND+7)=FCTL(7)+FCTL(18)+FCTL(19)
      FCT(INDG+7)=FCTG(7)+FCTG(18)+FCTG(19)
      FCT(IND+8)=FCTL(8)+FCTL(20)+FCTL(21)+FCTL(22)
      FCT(INDG+8)=FCTG(8)+FCTG(20)+FCTG(21)+FCTG(22)
      FCT(IND+9)=FCTL(9)
      FCT(INDG+9)=FCTG(9)
      FCT(IND+10)=FCTL(10)
      FCT(INDG+10)=FCTG(10)

C-----------------------------------------------------------------------
C--   Theoretical evolution of H+ without neutralisation
C-
      CPRH=Y(10*(2*(N+2))+2*N+N+1)
      CH=Y(10*(2*(N+2))+2*N+N+2)
      FCTLH=(1+FBAK)*F*CPRH-FBAK*F*CH-FREC*CH-FOUT*CH
      FCTLH=FCTLH/(EPSL*VC/EPS)
      FCT((10*(2*(N+2))+2*N+N+2))=FCTLH



C-----------------------------------------------------------------------

      RETURN
      END
```

```
      SUBROUTINE CONFIGSIM(MENU)
C////////////////////////////////////////////////////////////////////////
C///////  LOADING OF DATA DEFAULT FILES AND CALCULATIONS          ////////
C///////  OF COLUMN PARAMETERS FOR SIMULATION                     ////////
C///////                    NITRISIM                              ////////
C//////// V 2.3                                LAST UPDATE 03/97 ////////
C////////////////////////////////////////////////////////////////////////
C-------------------------------------------------------------------------
C            DECLARATIONS
C-------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER MENU
      INTEGER NMAX,CORPMAX,TMPSMAX,RMAX
      PARAMETER(NMAX=20)
      PARAMETER(CORPMAX=30)
      PARAMETER(TMPSMAX=300)
      PARAMETER(RMAX=4)
      INTEGER N,CORP,TMPS,REAC
      REAL*8 F1
      REAL*8 MUMAX,KI,KS,KA,KLGAZ,KLBIO
      DIMENSION D(CORPMAX),KLBIO(CORPMAX),KLGAZ(CORPMAX)
      REAL*8 LONG
      DIMENSION KS(RMAX,CORPMAX),KI(RMAX,CORPMAX)
      DIMENSION MUMAX(RMAX)
      DIMENSION KA(CORPMAX)
      DIMENSION CSAT(CORPMAX)
      DIMENSION STO(RMAX,CORPMAX)

      CHARACTER*6 NOM$(CORPMAX)

      COMMON/REACBIO2/MUMAX,KS,KI,STO
      COMMON/PHYPH/KA
      COMMON/PHYTRANS/D,KLBIO,KLGAZ,ASPGAZ,ASPBIO
      COMMON/PHYTRANS2/CSAT
      COMMON/BILLE/RO,HBIO,RNBIO,BWO
      COMMON/COLON/EPSL,EPSG,EPS,FINP,GINP,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      COMMON/COLON2/REHL,REHG,LONG,DCOL,SECT,DELTAP,DH
      COMMON/TAB/N,CORP,TMPS,REAC
      COMMON/TAB2/NOM$
      COMMON/PHYINI/T,P,PHINI

C-------------------------------------------------------------------------
C               ORIGIN OF THE SUBROUTINE CALL
C-------------------------------------------------------------------------
      IF (MENU.EQ.2) THEN
          GOTO 100
      ENDIF

C-------------------------------------------------------------------------
C           LOADING ACTUAL DEFAULT DATA FILES
C-------------------------------------------------------------------------
      OPEN (1,FILE='CARCOL.DAT',FORM='FORMATTED',STATUS='OLD')
        READ(1,*)LONG
        READ(1,*)DCOL
        READ(1,*)VA
        READ(1,*)VB
        READ(1,*)VC
        READ(1,*)EPS
        READ(1,*)RO
        READ(1,*)T
        READ(1,*)P
        READ(1,*)PHINI
        READ(1,*)BWO
      CLOSE(1)

      OPEN (1,FILE='FLOWCOL.DAT',FORM='FORMATTED',STATUS='OLD')
        READ(1,*)FINP
        READ(1,*)GINP
        READ(1,*)RL
        READ(1,*)RG
        READ(1,*)FBAK
        READ(1,*)FBAKPRIM
        READ(1,*)N

      CLOSE(1)
      OPEN (1,FILE='CORPS.DAT',FORM='FORMATTED',STATUS='OLD')
```

```
        READ(1,*)CORP
        DO 99 I=1,CORP
          READ(1,'(A6)')NOM$(I)
99      CONTINUE
      CLOSE(1)

      OPEN (1,FILE='PHYPH.DAT',FORM='FORMATTED',STATUS='OLD')
        DO 98 I=1,CORP
           READ(1,*)KA(I)
98      CONTINUE
      CLOSE(1)

      OPEN (1,FILE='STOIC.DAT',FORM='FORMATTED',STATUS='OLD')
        READ(1,*)REAC
        DO 97 I=1,REAC
          DO 96 J=1,CORP
           READ(1,*)STO(I,J)
96        CONTINUE
97      CONTINUE
      CLOSE(1)

      OPEN (1,FILE='PHYTRANS.DAT',FORM='FORMATTED',STATUS='OLD')
        DO 95 I=1,CORP
           READ(1,*)D(I)
           READ(1,*)KLBIO(I)
           READ(1,*)KLGAZ(I)
           READ(1,*)CSAT(I)
95    CONTINUE
      CLOSE(1)

      OPEN (1,FILE='CINET.DAT',FORM='FORMATTED',STATUS='OLD')
      DO 93 J=1,REAC
        READ(1,*)MUMAX(J)
        DO 94 I=1,CORP
           READ(1,*)KS(J,I)
           READ(1,*)KI(J,I)
94      CONTINUE
93    CONTINUE
      CLOSE(1)


C-------------------------------------------------------------------------
C             CALCULATION OF COLUMN PARAMETERS
C-------------------------------------------------------------------------

C-- COLUMN SECTION
100   SECT=2*3.1416*(DCOL/2)**2

C-- EMPTY DEGREES
C-- FROM ERGUN RELATION [FORLER (1992)]
C-- F(X)=0 RESOLVED BY NEWTOWN
      A=0.00001
      B=EPS*0.99
      C=1E-6
      NITERMAX=400
      FA=F1(A)
      FB=F1(B)
      FAFB=FA*FB
      XOLD=(A+B)/2
      XNEW=XOLD-2*C*F1(XOLD)/(F1(XOLD+C)-F1(XOLD-C))
      NITER=1
92    IF (ABS(XNEW-XOLD).LT.1E-6) THEN
         EPSG=XNEW
      ELSEIF (NITER.GT.NITERMAX) THEN
         PRINT*,'empty degrees not founds'
         PRINT*,'empty degree of column:',EPS
         PRINT*,'liquid empty degree: ',XNEW
         PRINT*,'function results: ',F1(XNEW)
         STOP
      ELSE
         NITER=NITER+1
         XOLD=XNEW
         XNEW=XOLD-2*C*F1(XOLD)/(F1(XOLD+C)-F1(XOLD-C))
         GOTO 92
      ENDIF
      EPSL=EPS-EPSG
```

```
C--
C-- Corrections from UAB TN 25.330
C-- Voidage fixed
      epsl=eps*(3800./(3800.+400.))
      epsg=eps-epsl

C-- Specific Gas/liquid exchange area
      ASPGAZ=1
      ASPBIO=1

C-- Flows velocity (m/s)
      VGAZ=GINP*(1+RG)/(epsg*SECT)/3600
      VLIQ=FINP*(1+RL)/(epsl*SECT)/3600

C-- VISCOsity A 20C (KG/M.S.)
      VISCLIQ=1E-3
      VISCGAZ=17E-6

C--  Hydrolic diameter (m)
      DH=RO

C-- Hydrolic reynolds
      REHL=2./3.*(dh*vliq*998)/((1-epsl)*viscliq)

      REHG=2./3.*(dh*vgaz*1.2)/((1-epsg)*viscgaz)

      RETURN
      END

C-------------------------------------------------------------------------
C                          FUNCTIONS
C-------------------------------------------------------------------------
      FUNCTION F1(X)
      IMPLICIT REAL*8 (A-H,O-Z)

      REAL*8 X
      REAL*8 LONG
      COMMON/COLON/EPSL,EPSG,EPS,FINP,GINP,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      COMMON/COLON2/REHL,REHG,LONG,DCOL,SECT,DELTAP,DH
      COMMON/BILLE/RO,HBIO,RNBIO,BWO

C-- Flow velocity
      IF (X.EQ.EPS.OR.X.EQ.0) THEN
         PRINT*,'EPS',EPS
         PRINT*,X
         STOP
      ENDIF

      IF (SECT.EQ.0) THEN
         SECT=1
      ENDIF

      VGAZ=GINP*(1+RG)/(X*SECT)
      VLIQ=FINP*(1+RL)/((EPS-X)*SECT)

C-- VISCOSITY 20C  (KG/M.S.)
      VISCLIQ=1E-3
      VISCGAZ=17E-6

C-- Gas ratio (FORLER 1992)
      F1=(VGAZ*VISCGAZ)/(VLIQ*VISCLIQ)*(1-X)*(1-X)*(EPS-X)*(EPS-X)
      F1=F1*(EPS-X)-X**3*(1-EPS+X)**2

C--  Pressure drop
      DELTAP=(170*(1-X)*17E-6/(RO*VGAZ/3600*1.2)+1.75)
      DELTAP=DELTAP*LONG*(1-X)/X**3
      DELTAP=DELTAP*1.2*(VGAZ/3600)**2/RO

C-------------------------------------------------------------------------
      RETURN
      END
```

```
        SUBROUTINE EDITCONF()
C////////////////////////////////////////////////////////////////////////
C///////          ROUTINE FOR LISTING AND MANAGMENT                ////////
C///////                  OF COLUMN PARAMETERS                     ////////
C///////                       NITRISIM                            ////////
C/////// V 2.3                                  LAST UPDATE 03/97 ////////
C////////////////////////////////////////////////////////////////////////
C-------------------------------------------------------------------------
C            DECLARATIONS
C-------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)

      INTEGER CHOIX,MENU,ITEMPO
      REAL*8 TRN,TEMPO

      INTEGER NMAX,CORPMAX,TMPSMAX,RMAX
      PARAMETER(NMAX=20)
      PARAMETER(CORPMAX=30)
      PARAMETER(TMPSMAX=300)
      PARAMETER(RMAX=4)
      INTEGER N,CORP,TMPS,REAC

      REAL*8 MUMAX,KI,KS,KA,KLGAZ,KLBIO

      DIMENSION D(CORPMAX),KLBIO(CORPMAX),KLGAZ(CORPMAX)

      REAL*8 LONG

      DIMENSION KS(RMAX,CORPMAX),KI(RMAX,CORPMAX)
      DIMENSION MUMAX(RMAX)

      DIMENSION KA(CORPMAX)
      DIMENSION CSAT(CORPMAX)

      DIMENSION STO(RMAX,CORPMAX)

      CHARACTER*6 NOM$(CORPMAX)


      COMMON/REACBIO2/MUMAX,KS,KI,STO
      COMMON/PHYPH/KA
      COMMON/PHYTRANS/D,KLBIO,KLGAZ,ASPGAZ,ASPBIO
      COMMON/PHYTRANS2/CSAT
      COMMON/BILLE/RO,HBIO,RNBIO,BWO
      COMMON/COLON/EPSL,EPSG,EPS,FINP,GINP,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      COMMON/COLON2/REHL,REHG,LONG,DCOL,SECT,DELTAP
      COMMON/TAB/N,CORP,TMPS,REAC
      COMMON/TAB2/NOM$
      COMMON/PHYINI/T,P,PHINI


C-------------------------------------------------------------------------
C            MENU
C-------------------------------------------------------------------------
99    WRITE(*,1100)
      WRITE(*,1001)
      WRITE(*,1004)
      WRITE(*,1002)'Parmeters of the simulation'
      WRITE(*,1002)'      1 - Design of the column'
      WRITE(*,1002)'      2 - Stoichiometries'
      WRITE(*,1002)'      3 - Biological kinetics'
      WRITE(*,1002)'      4 - Compound involved and their constants'
      WRITE(*,1002)'      5 - Back to previous menu'
      WRITE(*,1004)
      WRITE(*,1003)
      PRINT*
      PRINT*
      WRITE(*,1005)
      READ(*,*,err=99)CHOIX
      GOTO (100,200,300,400,500),CHOIX
      GOTO 99


C////////////////////////////////////////////////////////////////////////
C///////                      DESIGN                             ////////
C////////////////////////////////////////////////////////////////////////
```

```
C-- EDITION
100   WRITE(*,1100)
      WRITE(*,1006)'[ Column ]'
      TRN=N
      WRITE(*,1007)'Number of tanks:',TRN,' ','Volume part B:',
     +VB,'m3'
      WRITE(*,1007)'Volume part A:',VA,'m3','Volume part C:',VC,'m3'
      WRITE(*,1007)'Height:',LONG,'m','Diam. column:',DCOL,'m'
      WRITE(*,1007)'Empty degree of bed:',EPS,' ','Diam. Beads:',RO,'m'
      WRITE(*,1007)'Pressure:',P,'atm.','Temperature:',T,'C'
      WRITE(*,1007)'PH fixed:',PHINI,' ','Bio. Wash. :',BWO
      WRITE(*,1006)'[  flows  ]'
      WRITE(*,1007)'Liquid Flow:',FINP,'m3/h','Gas Flow:',GINP,'m3/h'
      WRITE(*,1007)'Liquid Recirc.:',RL,' ','Gas Recirc.:',RG,' '
      WRITE(*,1007)'Liquid back-mix:',FBAK,' ','Gas back-mix:',
     'FBAKPRIM,' '
      WRITE(*,1006)'[  infos  ]'
      WRITE(*,1007)'Reyn. hyd. Liquid:',REHL,' ','Reyn. hyd. Gas:',REHG
     +,' '
      WRITE(*,1007)'Liquid degree:',EPSL,' ','Gas degree:',EPSG,' '
      WRITE(*,1007)'Press. drop:',DELTAP,' ','Hydro. Diam.:',DH,'m'
      WRITE(*,1007)'Gas Area Exch.:',ASPGAZ,'m2','Bio. Area Exch.:',
     +ASPBIO,'m2'
      WRITE(*,1003)
      PRINT*

      WRITE(*,1001)
      WRITE(*,1002)'      1 - Change parameters'
      WRITE(*,1002)'      2 - Back to previous menu'
      WRITE(*,1003)
      PRINT*
      WRITE(*,1005)
      READ(*,*,err=100)CHOIX
      GOTO (110,99),CHOIX
      GOTO 100




C-- MODIFICATION
110   WRITE(*,1100)

      TRN=N
      WRITE(*,1008)'Number of tank equivalent for the bed (',TRN,') : '
      READ(*,*,err=124)REP
      N=INT(REP)


124   WRITE(*,1008)'Volume part A (bottom) (',VA,' m3) : '
      READ(*,*,err=125)REP
      VA=REP



125   WRITE(*,1008)'Volume part C (top) (',VC,' m3) : '
      READ(*,*,err=112)REP
      VC=REP



112   WRITE(*,1008)'Column height (',LONG,' m) : '
      READ(*,*,err=113)REP
      IF (REP.LE.0) THEN
         WRITE(*,1009)'NEGATIVE OR NUL'
         GOTO 112
      ENDIF
      LONG=REP


113   WRITE(*,1008)'Column Diameter (',DCOL,' m) : '
      READ(*,*,err=114)REP
      IF (REP.LE.0) THEN
         WRITE(*,1009)'NEGATIVE OR NUL'
```

```
          GOTO 113
        ENDIF
        DCOL=REP


114     VB=LONG*3.1416*(DCOL/2)**2-VA-VC
        IF (VB.LE.0) THEN
          PRINT*,'Error in scaling column'
          GOTO 112
        ENDIF


        WRITE(*,1008)'Empty degree of fixed bed (',EPS,') : '
        READ(*,*,err=111)REP
        IF (REP.LE.0.OR.REP.GT.1) THEN
          WRITE(*,1009)'between 0 and 1'
          GOTO 114
        ENDIF
        EPS=REP


111     WRITE(*,1008)'Beads mean diameter (',RO,') : '
        READ(*,*,err=115)REP
        IF (REP.LE.0.OR.REP.GE.DCOL) THEN
          WRITE(*,1009)'between 0 and column diameter'
          GOTO 111
        ENDIF
        RO=REP


115     WRITE(*,1008)'Input Gas flow rate (',GINP,'m3/h) : '
        READ(*,*,err=116)REP
        IF (REP.LE.0) THEN
          WRITE(*,1009)'NEGATIVE OR NUL'
          GOTO 115
        ENDIF
        GINP=REP


116     WRITE(*,1008)'Input Liquid flow rate (',FINP,
       +'m3/h) : '
        READ(*,*,err=117)REP
        IF (REP.LE.0) THEN
          WRITE(*,1009)'NEGATIVE OR NUL'
          GOTO 116
        ENDIF
        FINP=REP


117     WRITE(*,1008)'Liquid recycling ratio (',RL,') : '
        READ(*,*,err=118)REP
        IF (REP.LT.0) THEN
          WRITE(*,1009)'NEGATIVE'
          GOTO 117
        ENDIF
        RL=REP


118     WRITE(*,1008)'Gas recycling ratio (',RG,') : '
        READ(*,*,err=119)REP
        IF (REP.LT.0) THEN
          WRITE(*,1009)'NEGATIVE'
          GOTO 118
        ENDIF
        RG=REP


119     WRITE(*,1008)'Liquid Back-mixing ratio  (',FBAK,') : '
        READ(*,*,err=120)REP
        IF (REP.LT.0.OR.REP.GT.1) THEN
```

```
         WRITE(*,1009)'BETWEEN 0 AND 1'
         GOTO 119
      ENDIF
      FABK=REP


120   WRITE(*,1008)'Gas Back-mixing ratio (',FBAKPRIM,') : '
      READ(*,*,err=121)REP
      IF (REP.LT.0.OR.REP.GT.1) THEN
         WRITE(*,1009)'BETWEEN 0 AND 1'
         GOTO 120
      ENDIF
      FBAKPRIM=REP


121   WRITE(*,1008)'Pressure (',P,' atm) : '
      READ(*,*,err=126)REP
      IF (REP.LT.0) THEN
         WRITE(*,1009)'NEGATIVE'
         GOTO 121
      ENDIF
      P=REP


126   WRITE(*,1008)'Temperature (',T,' C) : '
      READ(*,*,err=122)REP
      T=REP


122   WRITE(*,1008)'pH fixed (',PHINI,') : '
      READ(*,*,err=123)REP
      IF (REP.LE.0.OR.REP.GT.14) THEN
         WRITE(*,1009)'BETWEEN 0 AND 14'
         GOTO 122
      ENDIF
      PHINI=REP

123   WRITE(*,1008)'Biomass ratio Wash out from beads (',BWO,') : '
      READ(*,*,err=130)REP
      IF (REP.LT.0.OR.REP.GT.1) THEN
         WRITE(*,1009)'BETWEEN 0 AND 1'
         GOTO 123
      ENDIF
      BWO=REP


130   MENU=2
      CALL CONFIGSIM(MENU)
      MENU=1
      CALL SAVECONF(MENU)
      GOTO 100



C////////////////////////////////////////////////////////////////////////
C///////      STOICHIOMETRIES    (MAX 4  REACTION )              ///////
C////////////////////////////////////////////////////////////////////////

C--  LISTING 4 MAIN REACTIONS NS, NB, MAINT NS, MAINT NB
200   WRITE(*,1100)
      WRITE(*,1010)'[  Main Stoechio.  ]'
      WRITE(*,1011)'Comp','Ns BioSynt.','Nb BioSynt.','Maint Ns',
     +'Maint Nb'
      WRITE(*,1017)
      IF (CORP.LE.15) THEN
          ITEMPO=CORP
      ELSE
          ITEMPO=15
      ENDIF
      DO  201 I=1,ITEMPO
          WRITE(*,1012)NOM$(I)
          DO 202 J=1,4
```

```
            WRITE(*,1013)STO(J,I)
202         CONTINUE
            WRITE(*,1015)'°'
201     CONTINUE
        WRITE(*,1017)

        PRINT*
        IF (CORP.GT.15) THEN
           WRITE(*,1023)'NEXT'
           READ*
           WRITE(*,1100)
        WRITE(*,1010)'[  Main Stoechio.  ]'
        WRITE(*,1011)'Comp','Ns BioSynt.','Nb BioSynt.','Maint Ns',
       +'Maint Nb'
        WRITE(*,1017)

           DO  204 I=ITEMPO+1,CORP
              WRITE(*,1012)NOM$(I)
              DO 205 J=1,4
                 WRITE(*,1013)STO(J,I)
205           CONTINUE
              WRITE(*,1015)'°'
204        CONTINUE
        WRITE(*,1017)
        ENDIF



C-------------------------------------------------------------------
C--     MORE THAN 4  REACTIONS
        IF (REAC.GT.4) THEN
           PRINT*,'NOT AVAILABLE ACTUALLY'
        ENDIF
C-------------------------------------------------------------------




C-- Menu For Stoichiometries managment
        PRINT*
        WRITE(*,1001)
        WRITE(*,1002)'      1 - Change Stoichiometries'
        WRITE(*,1002)'      2 - Back to previous menu'
        WRITE(*,1003)
        PRINT*
        WRITE(*,1005)
        READ(*,*,err=200)CHOIX
        GOTO (220,99),CHOIX
        GOTO 200




C-- MODIFICATION
220     WRITE(*,1100)
221     TRN=REAC

        WRITE(*,1008)'Nomber of Reactions -Maximum 4- (',TRN,') : '
        READ(*,*,err=222)REP
        IF (REP.LT.4.OR.REP.GT.4) THEN
           WRITE(*,1009)'BE LOWER AND GREATER THAN 4 '
           GOTO 221
        ENDIF
        REAC=INT(REP)
        PRINT*,' Warning !!! YOU MUST VALIDATE ALL NEW PARAMETERS BEFORE'
        PRINT*,' RETURN TO THE MAIN MENU OR IT WILL FAILED'


222     TRN=REAC
        WRITE(*,1008)'Reaction to modify (1-',TRN,') : '
        READ(*,*,err=222)REP
        IF (REP.LT.0.OR.REP.GT.REAC) THEN
           GOTO 222
```

```
      ENDIF
      IREAC=INT(REP)

      PRINT*
      PRINT*,'NOTE : coeeficients of Substrates are negative'
      PRINT*,'       coefficients of Produits are positive'
      PRINT*

         DO 224 J=1,CORP
            WRITE(*,1016)'Stochiometric coef. of ',NOM$(J),
     +' of Reaction ',IREAC,' (',STO(IREAC,J),') : '
            READ(*,*,err=224)REP
            STO(IREAC,J)=REP
224      CONTINUE


      MENU=2
      CALL SAVECONF(MENU)
      GOTO 200


C////////////////////////////////////////////////////////////////////////
C//////                    KINETIC PARAMETERS                        ///////
C////////////////////////////////////////////////////////////////////////

C-- EDITION
300   WRITE(*,1100)
      WRITE(*,1001)
      WRITE(*,1002)'     1 - Mu max. and Maintenance coeff.'
      WRITE(*,1002)'     2 - Saturation Constants'
      WRITE(*,1002)'     3 - Inhibitory constants'
      WRITE(*,1002)'     4 - Back to previous menu'
      WRITE(*,1003)
      PRINT*
      WRITE(*,1005)
      READ(*,*,err=300)CHOIX
      GOTO (310,320,360,99),CHOIX
      GOTO 300

C-- seulement 4 reactions fixees pour l'instant


C---------------------------------------------------------------------
C--   MU MAX AND MAINTENANCE COEFF OF THE 4 MAIN REACTIONS
C-
310   WRITE(*,1100)
      WRITE(*,1010)'[ Mu max. and Maint ]'
      WRITE(*,1011)' ','Ns BioSynt.','Nb BioSynt.','Maint. Ns',
     +'Maint. Nb'
      WRITE(*,1017)

      WRITE(*,1020)'Mu max.'
      WRITE(*,1012)' or  '
          DO 311 J=1,4
             WRITE(*,1013)MUMAX(J)
311       CONTINUE
          WRITE(*,1015)'°'
      WRITE(*,1020)'m coeff.'
      WRITE(*,1017)
      PRINT*
      PRINT*

C--  MU MAX FOR OTHER REACTIONS (MORE THAN 4 REACTIONS)
      IF (REAC.GT.4) THEN
      WRITE(*,1023)'SUITE'
      READ*
      PRINT*
      WRITE(*,1010)'[      Mu max       ]'
      WRITE(*,1011)' ','Reac 5','Reac 6','Reac 7'
      WRITE(*,1017)

      WRITE(*,1012)'Mu max.'
          DO 312 J=5,7
             WRITE(*,1013)MUMAX(J)
312       CONTINUE
```

```
          WRITE(*,1015)'°'
        WRITE(*,1017)
        PRINT*
        PRINT*
        ENDIF

        IF (REAC.GT.7) THEN
        WRITE(*,1023)'NEXT'
        READ*
        PRINT*
        WRITE(*,1010)'[      Mu max       ]'
        WRITE(*,1011)' ','Reac 8','Reac 9','Reac 10'
        WRITE(*,1017)

        WRITE(*,1012)'Mu max.'
            DO 313 J=8,10
               WRITE(*,1013)MUMAX(J)
313         CONTINUE
            WRITE(*,1015)'°'
        WRITE(*,1017)
        PRINT*
        PRINT*
        ENDIF


C-- MODIFICATION OF MU MAX
        WRITE(*,1001)
        WRITE(*,1002)'       1 - Change Mu max'
        WRITE(*,1002)'       2 - Back to previous menu'
        WRITE(*,1003)
        PRINT*
        WRITE(*,1005)
        READ(*,*,err=310)CHOIX
        GOTO (315,300),CHOIX
        GOTO 310

315     PRINT*
        PRINT*
        WRITE(*,1008)'Mu max. Biosynthesis Ns (',MUMAX(1),' h-1) :'
        READ(*,*,err=314)REP
        MUMAX(1)=REP

314     WRITE(*,1008)'Mu max. Biosynthesis Nb (',MUMAX(2),' h-1) :'
        READ(*,*,err=317)REP
        MUMAX(2)=REP

317     WRITE(*,1008)'Maintenance coeff Ns (',MUMAX(3),' NH3/h) :'
        READ(*,*,err=318)REP
        MUMAX(3)=REP

318     WRITE(*,1008)'Maintenance coeff Nb (',MUMAX(4),' NO2-/h) :'
        READ(*,*,err=319)MUMAX(4)

319     CONTINUE

C-----MORE THAN 4 REACTIONS-----------------------------------------
C        DO 316 I=5,REAC
C           WRITE(*,1018)'Mu max. reaction ',I,' (',MUMAX(I),' h-1) :'
C        READ(*,*,err=316)REP
C        MUMAX(I)=REP
C316     CONTINUE


        MENU=3
        CALL SAVECONF(MENU)
        PRINT*
        PRINT*
        GOTO 310


C-------------------------------------------------------------------------
C-- SATURATIONS CONSTANTS
C-
320     WRITE(*,1100)
        WRITE(*,1010)'[ Saturation Const. ]'
```

```
      WRITE(*,1011)' Comp','Ns BioSynt.','Nb BioSynt.','Maint. Ns',
     +'Maint. Nb'
      WRITE(*,1017)
      IF (CORP.LE.15) THEN
          ITEMPO=CORP
      ELSE
          ITEMPO=15
      ENDIF
      DO  321 I=1,ITEMPO
          WRITE(*,1012)NOM$(I)
          DO 322 J=1,4
              WRITE(*,1013)KS(J,I)
322       CONTINUE
          WRITE(*,1015)'°'
321   CONTINUE
      WRITE(*,1017)

      PRINT*
      IF (CORP.GT.15) THEN
         WRITE(*,1023)'SUITE'
         READ*
         WRITE(*,1100)
         WRITE(*,1010)'[ Const. Saturation  ]'
         WRITE(*,1011)' Corps','Ns BioSynt.','Nb BioSynt.','Maint. Ns',
     +'Maint. Nb'
         WRITE(*,1017)
         DO  323 I=ITEMPO+1,CORP
          WRITE(*,1012)NOM$(I)
          DO 324 J=1,4
              WRITE(*,1013)KS(J,I)
324       CONTINUE
          WRITE(*,1015)'°'
323       CONTINUE
          WRITE(*,1017)
      ENDIF


C--  Saturation constants for more than 4 reactions
      IF (REAC.GT.4) THEN
         PRINT*,'NOT IMPLEMENTED'
      ENDIF


C-- MODIFICATION OF SATURATION CONSTANTS
      PRINT*
      WRITE(*,1001)
      WRITE(*,1002)'      1 - Change Saturation Constants'
      WRITE(*,1002)'      2 - Back to previous menu'
      WRITE(*,1003)
      PRINT*
      WRITE(*,1005)
      READ(*,*,err=320)CHOIX
      GOTO (330,300),CHOIX
      GOTO 320

330   WRITE(*,1100)
      DO 331 I=1,CORP
         DO 332 J=1,REAC
            WRITE(*,1016)'Saturation Constant of ',NOM$(I),
     +' for reaction ',J,' (',KS(J,I),') : '
      READ(*,*,err=332)REP
      KS(J,I)=REP
332   CONTINUE
331   CONTINUE




C-- CHANGES IN KS FOR MORE THAN 4 REACTIONS
      IF (REAC.GT.4) THEN
         PRINT*,'NOT IMPLEMENTED'
      ENDIF


      MENU=3
```

```
      CALL SAVECONF(MENU)
      PRINT*
      PRINT*
      GOTO 320


C-------------------------------------------------------------------------
C-- INHIBITORY CONSTANTS
C-
360   WRITE(*,1100)
      WRITE(*,1010)'[  Inhibitory Const. ]'
      WRITE(*,1011)' Comp','Ns BioSynt.','Nb BioSynt.','Maint. Ns',
     +'Maint. Nb'
      WRITE(*,1017)
      IF (CORP.LE.15) THEN
          ITEMPO=CORP
      ELSE
          ITEMPO=15
      ENDIF
      DO  361 I=1,ITEMPO
          WRITE(*,1012)NOM$(I)
          DO 362 J=1,4
              WRITE(*,1013)KI(J,I)
362       CONTINUE
          WRITE(*,1015)'°'
361   CONTINUE
      WRITE(*,1017)

      PRINT*
      IF (CORP.GT.15) THEN
         WRITE(*,1023)'NEXT'
         READ*
         WRITE(*,1100)
         WRITE(*,1010)'[  Inhibitory Const. ]'
         WRITE(*,1011)' Comp','Ns BioSynt.','Nb BioSynt.','Maint. Ns',
     +'Maint. Nb'
         WRITE(*,1017)
         DO  363 I=ITEMPO+1,CORP
          WRITE(*,1012)NOM$(I)
          DO 364 J=1,4
              WRITE(*,1013)KI(J,I)
364       CONTINUE
          WRITE(*,1015)'°'
363       CONTINUE
          WRITE(*,1017)
      ENDIF



C--  INHIBITORY FOR MORE THAN 4 REACTIONS
      IF (REAC.GT.4) THEN
         PRINT*,'NOT IMPLEMENTED'
      ENDIF



C-- MODIFICATION OF INHIBITION CONSTANTS
      WRITE(*,1001)
      WRITE(*,1002)'      1 - Change Inhibitory Constants'
      WRITE(*,1002)'      2 - Back to previous menu'
      WRITE(*,1003)
      PRINT*
      WRITE(*,1005)
      READ(*,*,err=360)CHOIX
      GOTO (370,300),CHOIX
      GOTO 360

370   WRITE(*,1100)
      PRINT*
      DO 371 I=1,CORP
         DO 372 J=1,REAC
            WRITE(*,1016)'Inhibitory Constants of ',NOM$(I),
     +' for reaction ',J,' (',KI(J,I),') : '
         READ(*,*,err=372)REP
         KI(J,I)=REP
372   CONTINUE
```

```
371   CONTINUE


C-- MODIFICATION KI FOR MORE THAN 4 REACTIONS
      IF (REAC.GT.4) THEN
         PRINT*,'NOT IMPLEMENTED'
      ENDIF


      MENU=3
      CALL SAVECONF(MENU)
      PRINT*
      PRINT*
      GOTO 360



C//////////////////////////////////////////////////////////////////////
C///////                    COMPOUNDS                           ///////
C//////////////////////////////////////////////////////////////////////
400   WRITE(*,1100)
      WRITE(*,1025)'[  G-L  Constants  ]'
      WRITE(*,1026)'Comp.','Eq. pH','[C]. Satur.','Kl gaz',
     +'Kl biofilm','Diffusion'
      WRITE(*,1003)
      IF (CORP.LE.15) THEN
          ITEMPO=CORP
      ELSE
          ITEMPO=15
      ENDIF
      DO  401 I=1,ITEMPO
          WRITE(*,1012)NOM$(I)
          WRITE(*,1013)KA(I)
          WRITE(*,1013)CSAT(I)
          WRITE(*,1013)KLGAZ(I)
          WRITE(*,1013)KLBIO(I)
          WRITE(*,1013)D(I)
          WRITE(*,1015)'°'
401   CONTINUE
      WRITE(*,1003)

      PRINT*
      IF (CORP.GT.15) THEN
         WRITE(*,1023)'NEXT'
         READ*
         WRITE(*,1100)
         WRITE(*,1025)'[  G-L  Constants  ]'
         WRITE(*,1026)'Corps','Eq. pH','[C]. Satur.','Kl gaz',
     +'Kl biofilm','Diffusion'
         WRITE(*,1003)

          DO  404 I=ITEMPO+1,CORP
              WRITE(*,1012)NOM$(I)
              WRITE(*,1013)KA(I)
              WRITE(*,1013)CSAT(I)
              WRITE(*,1013)KLGAZ(I)
              WRITE(*,1013)KLBIO(I)
              WRITE(*,1013)D(I)
              WRITE(*,1015)'°'
404       CONTINUE
      WRITE(*,1003)
      ENDIF


      PRINT*
      WRITE(*,1001)
      WRITE(*,1002)'     1 - Change Compounds [warning]'
      WRITE(*,1002)'     2 - pH Equilibria Constants'
      WRITE(*,1002)'     3 - Partition G/L constant '
      WRITE(*,1002)'     4 - G/L Exchange coefficient'
      WRITE(*,1002)'     5 - L-Biofilm Exchange coefficient'
      WRITE(*,1002)'     6 - Biofilm diffusion coefficient'
      WRITE(*,1002)'     7 - Back to previous menu'
      WRITE(*,1003)
```

```
      PRINT*
      WRITE(*,1005)
      READ(*,*,err=400)CHOIX
      GOTO (410,420,430,440,450,460,99),CHOIX
      GOTO 400


C-----------------------------------------------------------------------
C-- CHANGE NUMBER OF COMPOUNDS
C-- THE 22 FIRST STAY UNCHANGED - DEFAULT MODEL

410   WRITE(*,1100)
      TRN=CORP
      WRITE(*,1008)'Number of compounds -Maximun 30- (',TRN,') : '
      READ(*,*,err=412)REP
      CORP=REP
      IF (CORP.LT.22.OR.CORP.GT.30) THEN
         WRITE(*,1009)'BETWEEN 22 AND 30'
         CORP=TRN
         GOTO 410
      ENDIF

412   IF (CORP.GT.22) THEN
      DO 411 I=23,CORP
         WRITE(*,1019)'Compound ',I,' (',NOM$(I),') : '
         READ(*,'(A)',err=411)NOM$(I)
411   CONTINUE
      ENDIF

      MENU=4
      CALL SAVECONF(MENU)
      PRINT*
      PRINT*
      GOTO 400

C-----------------------------------------------------------------------
C-- PH EQUILIBRIA CONSTANTS
C--
420   WRITE(*,1100)
      TRN=CORP
      DO 421 I=1,CORP
         WRITE(*,1027)'Ka of Compound ',NOM$(I),' at 25 C (',KA(I),') :'
      READ(*,*,err=421)REP
      KA(I)=REP
421   CONTINUE

      MENU=4
      CALL SAVECONF(MENU)
      GOTO 400



C-----------------------------------------------------------------------
C-- Partition coefficeint for compund in G/L equilibria
C-
430   WRITE(*,1100)
      TRN=CORP
      DO 431 I=1,CORP
         WRITE(*,1027)'Partition coefficient of ',NOM$(I),
     +' at 25 C (',CSAT(I),') :'
      READ(*,*,err=431)REP
      CSAT(I)=REP
431   CONTINUE

      MENU=4
      CALL SAVECONF(MENU)
      GOTO 400


C-----------------------------------------------------------------------
C-- GAS LIQUID TRANSFER COEEFICIENT
C-
440   WRITE(*,1100)
      TRN=CORP
```

```
      DO 441 I=1,CORP
         WRITE(*,1027)'Gaz-Liquide transfer coefficient of '
     +,NOM$(I),' at 25 C (',KLGAZ(I),') :'
      READ(*,*,err=441)REP
      KLGAZ(I)=REP
441   CONTINUE

      MENU=4
      CALL SAVECONF(MENU)
      GOTO 400


C--------------------------------------------------------------------------
C-- LIQUIDE BIOFILM TRANSFER COEFFICIENTS
C-
450   WRITE(*,1100)
      DO 451 I=1,CORP
         WRITE(*,1027)'Liq-Biofilm transfer coefficient of '
     +,NOM$(I),' at 25 C (',KLBIO(I),') :'
      READ(*,*,err=451)REP
      KLBIO(I)=REP
451   CONTINUE
      GOTO 400


C--------------------------------------------------------------------------
C-- BIOFILM DIFFUSION COEFFICIENT
C-
460   WRITE(*,1100)
      DO 461 I=1,CORP
         WRITE(*,1027)'Biofilm diffusion coefficient of ',
     +NOM$(I),' at 25 C (',D(I),') :'
      READ(*,*,err=461)REP
      D(I)=REP
461   CONTINUE

      MENU=4
      CALL SAVECONF(MENU)
      GOTO 400

C////////////////////////////////////////////////////////////////////////
C///////                      FORMATS                             ///////
C////////////////////////////////////////////////////////////////////////
1001  FORMAT(1x,'É',77('Í'),'»')
1002  FORMAT(1x,'°',T10,A,T80,'°')
1003  FORMAT(1x,'È',77('Í'),'¼')
1004  FORMAT(1x,'°',T80,'°')
1005  FORMAT(1x,5X,'Choice : ',\)
1007  FORMAT(1x,'°',T3,A20,T25,G8.3,T34,A5,T40,'³',T43,A20,T65,G8.3,T74,
     +A5,T80,'°')
1006  FORMAT(1x,'Ì',10('Í'),A11,57('Í'),T80,'¹')
1008  FORMAT(1x,A,F10.5,A,\)
1009  FORMAT(1x,'   WARNING !!!    THIS VALUE CAN NOT BE ',A)
1010  FORMAT(1x,'É',10('Í'),A21,32('Í'),'»')
1011  FORMAT(1x,'°',T3,A6,T10,'³',A12,T24,'³',A12,T38,'³',A12,T52,
     +'³',A12,T66,'°')
1012  FORMAT(1x,'°',T3,A6,\)
1013  FORMAT(' ³ ',G10.4,' ',\)
1014  FORMAT(' ³',A12,\)
1015  FORMAT(1X,A)
1016  FORMAT(1x,a,A,A,I2,A,G10.4,A,\)
1017  FORMAT(1x,'È',63('Í'),'¼')
1018  FORMAT(1x,A,I2,A,F10.5,A,\)
1019  FORMAT(1x,A,I2,A,A,A,\)
1020  FORMAT(1x,'°',T3,A6,T10,'³',T66,'°')
1023  FORMAT(1x,A,\)
1025  FORMAT(1x,'É',10('Í'),A21,46('Í'),'»')
1026  FORMAT(1x,'°',T3,A6,T10,'³',A12,T24,'³',A12,T38,'³',A12,T52,
     +'³',A12,T66,'³',A12,T80,'°')
1027  FORMAT(1x,A,A6,A,G10.4,A,\)
1100  FORMAT(24(/))

500   RETURN
      END
```

```
      SUBROUTINE BIOFILM()
C////////////////////////////////////////////////////////////////////////
C/                                                                      /
C/        CACULATION OF THE DYNAMIC BEHAVIOUR OF A FIXED BED            /
C/        NIRIFYING COLUMN                                              /
C/        BIOFILM MODEL BASED ON TN 27.1 27.2 27.3 and 32.1             /
C/        Calculation for a defined concentration profile inside the    /
C/        column and the bed                                            /
C/        Require an initialisation file                                /
C/                              NITRISIM                                /
C/ V 2.3                                                 UPDATE 04/97   /
C////////////////////////////////////////////////////////////////////////

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      INTEGER NMAX,CORPMAX,TMPSMAX,RMAX
      PARAMETER(NMAX=20)
      PARAMETER(CORPMAX=30)
      PARAMETER(TMPSMAX=300)
      PARAMETER(RMAX=4)
      PARAMETER(KMAX=30)

      INTEGER N,CORP,TMPS,REAC
      CHARACTER*13 FILE$,FILER$
      CHARACTER*50 DESCRIP$
      REAL*8 REHL,REHG,EPSL,EPSG,EPS,F,G,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      REAL*8 LONG,SECT,DELTAP,DCOL,DH
      REAL*8 T,P,PHINI,PHA,PHC
      REAL*8 MUMAX,KI,KS,KA,KLGAZ,KLBIO
      REAL*8 RO
      DIMENSION C0(60),C(CORPMAX)
      DIMENSION HBIO(NMAX),RNBIO(NMAX)
      DIMENSION CSAT(CORPMAX)
      REAL*8 ASPGAZ,ASPBIO
      DIMENSION D(CORPMAX),KLBIO(CORPMAX),KLGAZ(CORPMAX)
      DIMENSION KS(RMAX,CORPMAX),KI(RMAX,CORPMAX)
      DIMENSION MUMAX(RMAX)
      DIMENSION KA(CORPMAX)
      DIMENSION STO(RMAX,CORPMAX)
      DIMENSION RSNS(CORPMAX),RSNB(CORPMAX)
      DIMENSION YO(60),XTAB(100),YTAB(2000),DER(60)
      REAL*8 RXNS,RXNB

      CHARACTER*6 NOM$(CORPMAX)

      COMMON/REACBIO2/MUMAX,KS,KI,STO
      COMMON/PHYINI/T,P,PHINI
      COMMON/PHYPH/KA
      COMMON/PHYPH2/PHA,PHC,PHB
      COMMON/PHYTRANS/D,KLBIO,KLGAZ,ASPGAZ,ASPBIO
      COMMON/PHYTRANS2/CSAT
      COMMON/BILLE/RO,HBIO,RNBIO,BWO
      COMMON/COLON/EPSL,EPSG,EPS,FINP,GINP,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      COMMON/COLON2/REHL,REHG,LONG,DCOL,SECT,DELTAP,DH
      COMMON/TAB/N,CORP,TMPS,REAC
      COMMON/TAB2/NOM$
      COMMON/BIOFIX/CBNS,CBNB
      PMBIO=23.1443

      WRITE(*,1100)
      WRITE(*,1021)
      WRITE(*,1026)
      WRITE(*,1022)'Biofilm concentrations profiles'
      WRITE(*,1026)
      WRITE(*,1023)
      PRINT*
      PRINT*

C=======================================================================
C              INITIALISATION
C=======================================================================

      WRITE(*,1001)' File for column liquid profile at t : '
      READ(*,'(A,\)')FILE$
10    WRITE(*,1001)' File for results (8 letters maxi): '
      READ(*,'(A,\)')FILER$
```

```
      IDBLAN=INDEX(FILER$,' ')-1
      IF (IDBLAN.EQ.0.OR.IDBLAN.GT.8) GOTO 10
      FILER$=FILER$(1:IDBLAN)//'.bfm'
      WRITE(*,1001)' Comments : '
      READ(*,'(A,\)')DESCRIP$

      OPEN(1,FILE=FILE$,FORM='FORMATTED')
      OPEN(2,FILE=FILER$)
C-----------------------------------------------------------------------
C--  READING THE NUMBER OF SEGMENTS
C-
      READ(1,*)NSEG

C-----------------------------------------------------------------------
C--   INITIALISATION OF RESULT DATA FILE
      WRITE(2,*)'File of liquid concentrations : ',FILE$
      WRITE(2,*)'Comments : ',DESCRIP$

      WRITE(2,*)NSEG
      DO 40 I=1,CORP
         WRITE(2,1002)NOM$(I)
40    CONTINUE
         WRITE(2,1002)'Bio æm'
      WRITE(2,*)

C-----------------------------------------------------------------------
C--   Beginig of simulation
      PRINT*
      PRINT*
      WRITE(*,*)'Segment    Sum of derivaties at R=0'
      print*
      DO 50 J=1,NSEG

C-----------------------------------------------------------------------
C--   Loading raw data in format of  version1.1
C-    For one segment of the bed J
C-

         READ(1,*)C0(1),c0(2),c0(3),c0(4),C0(5),c0(6),c0(7),c0(8),
     +C0(9),c0(10),c0(11),c0(12),C0(13),c0(14),c0(15),c0(16),C0(17),
     +C0(18),c0(19),c0(20),c0(21),C0(22),c0(23)

C-----------------------------------------------------------------------
C--   Calcul des concentrations reeelle avec PH
C-
      CTNH3=C0(1)
      CTNO3H=C0(2)
      CTNO2H=C0(3)
      CTCO2=C0(4)
      CTO2=C0(5)
      CTH2O=C0(6)
      CTH2SO=C0(7)
      CTH3PO=C0(8)
      CTNS=C0(9)
      CTNB=C0(10)
      CNS=C0(21)
      CNB=C0(22)

C-----------------------------------------------------------------------
C--   CALCULATION OF PH EQUILIBRIA FOR A FIXED PHINI
C-
      C(11)=10**(-PHINI)
      C(12)=10**(PHINI-14)

      C(1)=CTNH3*C(12)/(KA(1)+c(12))
      C(13)=CTNH3-C(1)

      C(2)=CTNO3H*C(11)/(KA(2)+c(11))
      C(14)=CTNO3H-C(2)

      C(3)=CTNO2H*C(11)/(KA(3)+c(11))
      C(15)=CTNO2H-C(3)

      C(16)=CTCO2/(10**(-PHINI)/KA(4)+1+KA(16)/10**(-PHINI))
      C(17)=C(16)*KA(16)/10**(-PHINI)
      C(4)=CTCO2-C(16)-C(17)
```

```
      C(18)=CTH2SO/(10**(-PHINI)/KA(7)+1+KA(18)/10**(-PHINI))
      C(19)=C(18)*KA(18)/10**(-PHINI)
      C(7)=CTH2SO-C(18)-C(19)


      C(21)=CTH3PO/((10**(-2*PHINI)/(KA(8)*KA(20)))+
     +10**(-PHINI)/KA(20)+1+KA(21)/10**(-PHINI))
      C(22)=KA(21)*C(21)/10**(-PHINI)
      C(20)=C(21)*10**(-PHINI)/KA(20)
      C(8)=CTH3PO-C(21)-C(20)-C(22)

      C(5)=CTO2
      C(6)=CTH2O
      C(9)=CTNS
      C(10)=CTNB

C-------------------------------------------------------------------------
C--   CONVERSION OF BIOMASS CONCENTRATION INTO BIOFILM VOLUME
C--   270.3 g dry biomass / l biofilm
C-
      CBNS=CNS/(CNS+CNB)*270.3
      CBNB=CNB/(CNS+CNB)*270.3

C-------------------------------------------------------------------------
C--   INITIALISATION FOR INTEGRATION IN RKMER2
C-    HBIO et HINI: RANGE OF THE BIOFILM
C-    ITAB: NUMBER OF DATA CALCULATED IN BIOFILM (PROFILE)
C-    YO: INITIALISATION AT HBIO FOR
C-                CONCENTRATTION (INDEX:1-CORP)
C-                DERIVATIES (INDEX:CORP+1 - 2*CORP)
C-    NEQ=NUMBER OF DIFFERENTIAL EQUATIONS
C-   Ro DIAMETER OF BEADS IN m

      HBIO(J)=(CNS+CNB)*EPSL*RO/2/((1-eps)*3)
      HBIO(J)=HBIO(J)*3.7e-3
      HINI=0.
      ITAB=15
      NEQ=2*CORP

      DO 100 I=1,CORP
         YO(I)=C(I)

         SUBSLIMNS=1
         SUBSLIMNB=1
         IF (STO(1,I).LT.0) THEN
         IF (KS(1,I).NE.0..OR.KI(1,I).LE.1000) THEN
            if (C(I).lt.0.) then
               C(I)=0.
            endif
          SUBSLIMNS=SUBSLIMNS*C(I)/((KS(1,I)+C(I))*
     +(1+C(I)/KI(1,I)))
         ENDIF
         ELSEIF (STO(1,I).GT.0) THEN
         IF (KS(1,I).NE.0..OR.KI(1,I).LE.1000) THEN
            if (C(I).lt.0.) then
               C(I)=0.
            endif
          SUBSLIMNS=SUBSLIMNS/(1+C(I)/KI(1,I))
         ENDIF
         ENDIF

         IF (STO(2,I).LT.0.) THEN
         IF (KS(2,I).NE.0..OR.KI(2,I).LE.1000) THEN
            if (C(I).lt.0.) then
                C(I)=0.
             endif
          SUBSLIMNB=SUBSLIMNB*C(I)/((KS(2,I)+C(I))*
     +(1+C(I)/KI(2,I)))
         ENDIF
         ELSEIF (STO(2,I).GT.0.) THEN
         IF (KS(2,I).NE.0..OR.KI(2,I).LE.1000) THEN
            if (C(I).lt.0.) then
                C(I)=0.
             endif
          SUBSLIMNB=SUBSLIMNB/(1+C(I)/KI(2,I))
```

```
         ENDIF
         ENDIF
100    CONTINUE


C-----------------------------------------------------------------------
C-- FIXED BIOMASS GROWTH-- BEWARE OF THE REFERENCE SUBSTRATE
C-
C          1-NH3  ---------> Reference for Ns
C          2-HNO3
C          3-HNO2
C          4-CO2
C          5-O2
C          6-H2O
C          7-H2SO4
C          8-H3PO4
C          9-BIOMASSE LIBRE NS
C          10-BIOMASSE LIBRE NB
C          11-H+
C          12-OH-
C          13-NH4+
C          14-NO3-
C          15-NO2-  -------> reference for Nb
C          16-HCO3-
C          17-CO32-
C          18-HSO4-
C          19-SO42-
C          20-H2PO4-
C          21-HPO4-
C          22-PO42-

       RXNS=MUMAX(1)*SUBSLIMNS*CBNS+
      +(SUBSLIMNS-1)*(STO(1,9)*PMBIO/(-1.*STO(1,1)))*MUMAX(3)*CBNS

       RXNB=MUMAX(2)*SUBSLIMNB*CBNB+
      +(SUBSLIMNB-1)*(STO(2,10)*PMBIO/(-1.*STO(2,15)))*MUMAX(4)*CBNB

C-----------------------------------------------------------------------
C-- SUBSTRATES AND products  kinetics
C-

       DO 101 I=1,8
          RSNS(I)=(STO(1,I)/(PMBIO*STO(1,9)))*RXNS+
      +STO(3,I)/(-1.*STO(3,1))*MUMAX(3)*CBNS

          RSNB(I)=(STO(2,I)/(PMBIO*STO(2,10)))*RXNB+
      +STO(4,I)/(-1.*STO(4,15))*MUMAX(4)*CBNB

101    CONTINUE

       DO 103 I=11,CORP

          RSNS(I)=(STO(1,I)/(PMBIO*STO(1,9)))*RXNS+
      +STO(3,I)/(-1.*STO(3,1))*MUMAX(3)*CBNS

          RSNB(I)=(STO(2,I)/(PMBIO*STO(2,10)))*RXNB+
      +STO(4,I)/(-1.*STO(4,15))*MUMAX(4)*CBNB

103    CONTINUE

       RSNS(9)=0.
       RSNB(10)=0.

       DO 104 I=CORP+1,2*CORP
           Yo(i)=0.
104    CONTINUE


C=======================================================================
C--  CALCULATION OF THE BIOFILM PROFILE (MODELE 2) - NO TRANSFER RESISTANCE
C--  CALL OF RKMER2
C=======================================================================
149    CALL RKMER2(HBIO(J),HINI,YO,NEQ,ITAB,XTAB,YTAB)


C-----------------------------------------------------------------------
```

```
C--   Tests on outputs from RKMER2 in order to verify
C--     the nul derivaty in HINI
C-      and  REDEFINITION Initial condition on derivaties

       ITER=0
       SDER=0.
       DO 150 I=1,CORP
          IND=(ITAB-1)*NEQ+CORP
          DER(I)=YTAB(IND+I)
          SDER=SDER+DER(I)
          IF (DABS(DER(I)).GT.1E-8) THEN
             ITER=ITER+1
             YO(CORP+I)=YO(CORP+I)-DER(I)/10.
          ENDIF
150    CONTINUE
       IF (ITER.GT.0) THEN
          WRITE(*,1020)J,SDER
          GOTO 149
       ENDIF

C-------------------------------------------------------------------------
C--   Storage of results
C-
       DO 200 K=1,ITAB
          IND=(K-1)*2*CORP
       DO 201 I=1,CORP
             WRITE(2,1003)YTAB(IND+I)
201       CONTINUE
          WRITE(2,1004)XTAB(K)*1E6
200    CONTINUE
       WRITE(2,*)


C-------------------------------------------------------------------------
C--    Next segment of the bed
C-

50     CONTINUE

       CLOSE(1)
       CLOSE(2)

C=========================================================================
C      FORMATS
C=========================================================================
1001   FORMAT(A)
1002   FORMAT(A10,'    ',\)
1003   FORMAT(G10.4,'    ',\)
1004   FORMAT(G10.4)
1020   FORMAT('+',',    ',i2,'       ',G10.4)
1021   FORMAT(1x,'É',77('Í'),'»')
1022   FORMAT(1x,'°',T10,A,T80,'°')
1023   FORMAT(1x,'È',77('Í'),'¼')
1026   FORMAT(1x,'°',T80,'°')
1100   FORMAT(24(/))
       RETURN
       END



C=========================================================================
C=========================================================================
C=========================================================================
C      MODEL SUBROUTINE FOR BIOFILM DIFFUSION
C=========================================================================
C=========================================================================
C=========================================================================
       SUBROUTINE DERIVBIO(Y,X,FCT)
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)

       INTEGER NMAX,CORPMAX,TMPSMAX,RMAX
       PARAMETER(NMAX=20)
       PARAMETER(CORPMAX=30)
       PARAMETER(TMPSMAX=300)
       PARAMETER(RMAX=4)
       PARAMETER(KMAX=30)
```

```
      INTEGER N,CORP,TMPS,REAC
      CHARACTER*10 FILE$,FILER$
      REAL*8 REHL,REHG,EPSL,EPSG,EPS,F,G,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      REAL*8 LONG,SECT,DELTAP,DCOL,DH
      REAL*8 T,P,PHINI,PHA,PHC
      REAL*8 MUMAX,KI,KS,KA,KLGAZ,KLBIO
      REAL*8 RO
      DIMENSION C(CORPMAX)
      DIMENSION HBIO(NMAX),RNBIO(NMAX)
      DIMENSION CSAT(CORPMAX)
      REAL*8 ASPGAZ,ASPBIO
      DIMENSION D(CORPMAX),KLBIO(CORPMAX),KLGAZ(CORPMAX)
      DIMENSION KS(RMAX,CORPMAX),KI(RMAX,CORPMAX)
      DIMENSION MUMAX(RMAX)
      DIMENSION KA(CORPMAX)
      DIMENSION STO(RMAX,CORPMAX)
      DIMENSION RSNS(CORPMAX),RSNB(CORPMAX)
      DIMENSION Y(650),FCT(60)
      REAL*8 RXNS,RXNB
      REAL*8 CBNS,CBNB
      REAL*8 X
      CHARACTER*6 NOM$(CORPMAX)

      COMMON/REACBIO2/MUMAX,KS,KI,STO
      COMMON/PHYINI/T,P,PHINI
      COMMON/PHYPH/KA
      COMMON/PHYPH2/PHA,PHC,PHB
      COMMON/PHYTRANS/D,KLBIO,KLGAZ,ASPGAZ,ASPBIO
      COMMON/PHYTRANS2/CSAT
      COMMON/BILLE/RO,HBIO,RNBIO,BWO
      COMMON/COLON/EPSL,EPSG,EPS,FINP,GINP,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      COMMON/COLON2/REHL,REHG,LONG,DCOL,SECT,DELTAP,DH
      COMMON/TAB/N,CORP,TMPS,REAC
      COMMON/TAB2/NOM$
      COMMON/BIOFIX/CBNS,CBNB
      PMBIO=23.0438
C-------------------------------------------------------------------------
C--   CALCULATION OF RS In every point of the biofilm
C-------------------------------------------------------------------------
      DO 100 I=1,CORP

         SUBSLIMNS=1
         SUBSLIMNB=1
         IF (Y(I).LT.0) THEN
             Y(I)=0
         ENDIF
         IF (STO(1,I).LT.0) THEN
           IF (KS(1,I).NE.0..OR.KI(1,I).LE.1000) THEN
            SUBSLIMNS=SUBSLIMNS*Y(I)/((KS(1,I)+Y(I))*
     +(1+Y(I)/KI(1,I)))
           ENDIF
         ELSEIF (STO(1,I).GT.0) THEN
           IF (KS(1,I).NE.0..OR.KI(1,I).LE.1000) THEN
             SUBSLIMNS=SUBSLIMNS/(1+Y(I)/KI(1,I))
           ENDIF
         ENDIF

         IF (STO(2,I).LT.0.) THEN
           IF (KS(2,I).NE.0..OR.KI(2,I).LE.1000) THEN
              SUBSLIMNB=SUBSLIMNB*Y(I)/((KS(2,I)+Y(I))*
     +(1+Y(I)/KI(2,I)))
           ENDIF
         ELSEIF (STO(2,I).GT.0.) THEN
             IF (KS(2,I).NE.0..OR.KI(2,I).LE.1000) THEN
               SUBSLIMNB=SUBSLIMNB/(1+Y(I)/KI(2,I))
            ENDIF
         ENDIF
100   CONTINUE

C-------------------------------------------------------------------------
C-- Fixed biomass growth -- Beware of the refernce substrate
C-
C          1-NH3 ------> reference for Ns
C          2-HNO3
C          3-HNO2
```

```
C          4-CO2
C          5-O2
C          6-H2O
C          7-H2SO4
C          8-H3PO4
C          9-BIOMASSE LIBRE NS
C          10-BIOMASSE LIBRE NB
C          11-H+
C          12-OH-
C          13-NH4+
C          14-NO3-
C          15-NO2- -----> refernece for Nb
C          16-HCO3-
C          17-CO32-
C          18-HSO4-
C          19-SO42-
C          20-H2PO4-
C          21-HPO4-
C          22-PO42-

       RXNS=MUMAX(1)*SUBSLIMNS*CBNS+
     +(SUBSLIMNS-1)*(STO(1,9)*PMBIO/(-1.*STO(1,1)))*MUMAX(3)*CBNS

       RXNB=MUMAX(2)*SUBSLIMNB*CBNB+
     +(SUBSLIMNB-1)*(STO(2,10)*PMBIO/(-1.*STO(2,15)))*MUMAX(4)*CBNB

C-------------------------------------------------------------------------
C-- Substrate and products kinetics (Rs)
C-

       DO 101 I=1,8
          if (y(i).eq.0.and.sto(1,i).LT.0.) then
             rsns(i)=0.
          elseif (y(i).eq.0.and.sto(2,i).LT.0.) then
             rsnb(i)=0.
          else

          RSNS(I)=(STO(1,I)/(PMBIO*STO(1,9)))*RXNS+
     +STO(3,I)/(-1.*STO(3,1))*MUMAX(3)*CBNS

          RSNB(I)=(STO(2,I)/(PMBIO*STO(2,10)))*RXNB+
     +STO(4,I)/(-1.*STO(4,15))*MUMAX(4)*CBNB
           endif

101    CONTINUE

       DO 103 I=11,CORP
          if (y(i).eq.0.and.sto(1,i).LT.0) then
             rsns(i)=0.
          elseif (y(i).eq.0.and.sto(2,i).LT.0) then
             rsnb(i)=0.
          else

          RSNS(I)=(STO(1,I)/(PMBIO*STO(1,9)))*RXNS+
     +STO(3,I)/(-1.*STO(3,1))*MUMAX(3)*CBNS

          RSNB(I)=(STO(2,I)/(PMBIO*STO(2,10)))*RXNB+
     +STO(4,I)/(-1.*STO(4,15))*MUMAX(4)*CBNB
          endif
103    CONTINUE

       RSNS(9)=0.
       RSNB(10)=0.

C-------------------------------------------------------------------------
C--    MODEL for integration
C-     D in m2/s

       DO 150 I=CORP+1,2*CORP
          FCT(I)=-1/(D(I-CORP)*3600)*(RSNS(I-CORP)+RSNB(I-CORP))
150    CONTINUE
       DO 151 I=1,CORP
          if (y(i).eq.0.) then
             fct(i)=0.
          else
             FCT(I)=Y(I+CORP)
```

```
        endif
151   CONTINUE

      RETURN
      END




C=======================================================================
C=======================================================================
C=======================================================================
C     SUBROUTINE FOR THE INTEGRATION - RKMER2
C=======================================================================
C=======================================================================
C=======================================================================
CDEB  drkmer
      SUBROUTINE RKMER2(X0,XF,Y0,N,ITAB,XTAB,YTAB)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION YI(100),ERR(100),F(100),ERRMAX(100)
      DIMENSION Y0(1),XTAB(1),YTAB(1)
      COMMON/RKMERY/XIM1,YIM1(100)
      COMMON/RKMERZ/ISTOP
      DOUBLE PRECISION   K1(100),K3(100),K4(100),K5(100)
      DATA ERRMAX/100*0.01/
      DATA H/-1e-6/MODTAB/1/

      ISTOP=0
      NMAX=100
      IF(N.GT.NMAX) GO TO 999
      NHS2=0
C
C      INITIALISATION
C
      IK=0
      HTAB=(XF-X0)/FLOAT(ITAB-1)
      IF(MODTAB.EQ.0) GO TO 2
      DO 1 I=1,ITAB
    1 XTAB(I)=X0+HTAB*FLOAT(I-1)
    2 CONTINUE
      DO 3 J=1,N
      YTAB(J)=Y0(J)
      YIM1(J)=Y0(J)
    3 YI(J)=Y0(J)
      XIM1=X0
      KTAB=2
    4 CONTINUE
      IF( ABS(H).LT. ABS(HTAB)) GO TO 5
      H=H/2.
      GO TO 4
    5 XI=XIM1+H
C
C      TEST FOR THE TEMPORARY STORAGE OF RESULTS Y
C
      XFI0=(XTAB(KTAB)-XI)/(XF-X0)
      IF (XFI0.LT.1..AND.XFI0.GT.0.) GOTO 10
      H0=H
      H=XTAB(KTAB)-XIM1
      IK=1

C
C      ALGORITHME OF RUNGE KUTTA  MERSON OF THE 4th ORDER
C
   10 XI=XIM1
      CALL DERIVBIO(YI,XI,K1)
      XI=XIM1+H/3.
      DO 21 J=1,N
   21 YI(J)=YI(J)+K1(J)*H/3.
      CALL DERIVBIO(YI,XI,K3)
      DO 23 J=1,N
   23 YI(J)=YIM1(J) +(K1(J)+K3(J))/2.*H/3.
      CALL DERIVBIO(YI,XI,K3)
```

```
      XI=XIM1+H/2.
      DO 25 J=1,N
   25 YI(J)=YIM1(J) +3./8.*(K1(J)+3.*K3(J))*H/3.
      CALL DERIVBIO(YI,XI,K4)
      XI=XIM1+H
      DO 27 J=1,N
   27 YI(J)=YIM1(J) +3./2.*(K1(J)-3.*K3(J)+K4(J)*4.) *H/3.
      CALL DERIVBIO(YI,XI,K5)
      DO 30 J=1,N
      YI(J)=YIM1(J)  +(K1(J)+4.*K4(J)+K5(J))/2. *H/3.
      E=K1(J)-9./2.*K3(J)+4.*K4(J)-K5(J) /2.
       E=E*H/3.
      ERR(J)=0.
      IF(ABS(YI(J) ).GT.1.E-20) ERR(J)=ABS(E/YI(J))
   30 CONTINUE

C
C      New integration step
C
      IF(ISTOP.NE.1) GO TO 40
      XF=XI
      DO 35 J=1,N
   35 Y0(J)=YI(J)
      RETURN

   40 CONTINUE

      IC=0
      DO 50 J=1,N
      IF(ERR(J).LT.ERRMAX(J)) GO TO 45
      NHS2=NHS2+1
      IF(NHS2.GT.20) GO TO 995
      H=H/2.
      DO 42 JP=1,N
42    YI(JP)=YIM1(JP)
      IK=0
      GO TO 10
   45 IF(ERR(J).GT.ERRMAX(J)/16.) IC=1
   50 CONTINUE
      IF(IK.EQ.1) GO TO 60
      IF(IC+NHS2.EQ.0) H=H*2
      NHS2=0
C
C      REINITIALISATION for the next step
C
      XIM1=XI
      DO 55 J=1,N
   55 YIM1(J)=YI(J)
      GO TO 5
C
C      Storage in memory of the calculated results
C      and REINITIALISATION
C
   60 IK=0
      H=H0
      DO 65 J=1,N
      JKTAB=N*(KTAB-1)+J
      YTAB(JKTAB)=YI(J)
   65 YIM1(J)=YI(J)
      XIM1=XTAB(KTAB)
      KTAB=KTAB+1
      IF(KTAB.LE.ITAB) GO TO 5
      RETURN
995   WRITE(*,996) NHS2
996   FORMAT(1X,5(1H*),'Stop in RKMER2 after ',I3,'DIVISIONS ',
     1   'by 2 of the step')
      CALL DERIV(YIM1,XIM1,K5)
      WRITE(*,*) X,H,(YIM1(I),I=1,N)
      WRITE(*,*) (K5(I),I=1,N)
      STOP
999   WRITE(*,9991) NMAX
9991  FORMAT(1X,5(1H*),'ERROR in RKMER2 * more than ',I3,' EQUATIONS',
     1   50(1H*))
      STOP
      END
```

```
C////////////////////////////////////////////////////////////////////////
C/    MAIN PROGRAM FOR THE SIMULATION OF A FIXED BED COLUMN             /
C/                          FOR NITRIFICATION                           /
C/                            NITRISIM SOFT                             /
C/                                                                      /
C/                       PROJET M.E.L.i.S.S.A.                          /
C/                                                                      /
C/ L. POUGHON LAST UPDATE 03/97                                         /
C/ VERSION 2.3 PC DOS SYSTEM                                            /
C////////////////////////////////////////////////////////////////////////


C-----------------------------------------------------------------------
C          BASIC ASSUMPTION OF THE AUTOTROPHIC NITRIFYING MODEL
C                    FOR FIXED BED COLUMN
C          Report to TN 27.1 27.2 27.3 32.1
C
C          Cylindrical column divided into 3 parts
C               A: Bottom of column - mixing of entries - acquisition zone
C               B: Fixed bed - active zone
C               C: Top of the column - output fluxes - acquisition zone
C
C          Column assimilated to a N-tanks in serie process
C          with a back-mixing term
C
C          Free biomass can not be fixed to the beads
C
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
C                           Program Variables
C-----------------------------------------------------------------------
C     SCALE-UP VARIABLES
C          RMAX=MAXIMAL NUMBER OF REACTIONS (4)
C          NMAX=Maximal number of tanks for the fixed bed (50)
C          TMPSMAX=MAXIMAL NUMBER OF SAVED POINTS DURING A SIMULATION (300)
C          CORPMAX=MAXIMAL NUMBER OF  COMPOUNDS INVOLVED (30)

C     FLOW VARIABLES
C          REHL=HYDROLIC LIQUID REYNOLDS                              -ADIM-
C          REHG=HYDROLIC GAS REYNOLDS                                 -ADIM-
C          EPSL=RATIO VOLUME OCCUPED BY LIQUID                        -ADIM-
C          EPSG=RATIO VOLUME OCCUPED BY GAS                           -ADIM-
C          EPS=EMPTY DEGREE OF THE FIXED BED                          -ADIM-
C          F=VOLUMIC LIQUIDE FLOW RATE                                -M3/H-
C          G=VOLUMIC GAS FLOW RATE                                    -M3/H-
C          RL=LIQUID RECYCLING RATIO                                  -ADIM-
C          RG=GAS RECYCLING RATIO                                     -ADIM-
C          VA=VOLUME PART A OF THE COLUMN                             -M3-
C          VB=VOLUME PART B OF THE COLUMN - FIXED BED                 -M3
C          VC=VOLUME PART C OF THE COLUMN                             -M3-
C          FBAK=LIQUID BACK-MIXING RATIO                              -ADIM-
C          FBAKPRIM=GAS BACK-MIXING RATIO                             -ADIM-
C          DCOL=COLUMN DIAMETER                                       -M-
C          SECT=COLUMN CROSS SECTION                                  -M2-
C          FIN=INPUT LIQUID FLOW RATE                                 -M3/H-
C          GIN=INPUT GAS FLOW RATE                                    -M3/H-


C      PHYSICAL VARIABLES
C          T=TEMPERATURE                                              -C-
C          P=PRESSURE                                                 -ATM-
C          PHINI=FIXED PH                                             -UPH-
C          PHA=PH IN PART A                                           -UPH-
C          PHB(N)=PH IN TANK N OF PART B                              -UPH-
C          PHC=PH IN PART C                                           -UPH-
C          TMPS=TIME ITERATION                                        -ADIM-
C          TMPSFIN=LENGHT OF PROCESS SIMULATION                       -H-

C      BED AND BIOFILM VARIABLES
C          RO=BEAD RADIUS                                             -M-
C          HBIO(N)=BIOFILM THICKNESS IN TANK N OF PART B              -M-
C          RNBIO(N)=O2 RESPIRATORY RATE                               -MOL
O2/Liq.H
C          THICKLIM=LIMITING BIOFILM THICKNESS FOR THE O2 TRANSFER    -M-
C          PMBIO=C-MOLAR WEIGHT OF BIOMASS                            -G/MOL-
```

```
C       CONCENTRATIONS (MOL/L execpeted BIOMASSE IN G/L)
C          C(I)=LIQUID CONCENTRATION OF COMPOUND I IN CURRENT COLUMN SEGMENT
C          CSU(I)=LIQUID CONCENTRATION OF COMPOUND I IN FOLLOWING SEGMENT
C          CPR(I)=LIQUID CONCENTRATION OF COMPOUND I IN PREVIOUS SEGMENT
C          COUT(I)=LIQUID CONCENTRATION OF COMPOUND I AT COLUMN OUTPUT
C          CSAT(I)=SATURATION CONSTANT (DEFINED AT T AND P)
C          CINL(I)=LIQUID CONCENTRATION AT COLUMN INPUT
C          CING(I)=GAS CONCENTRATION AT COLUMN INPUT

C          Y0(650)=VECTOR OF INTEGRO-DIFERENTIAL SYSTEM
C          FCT(650)=VECTOR OF DERIVATIES Y0                          -MOL/L.H-
C          FCTL=DERIVATIES OF LIQUID COMPOUND
C          FCTG=DERIVATIES OF GAS COMPOUND

C       TRANSFER LIMITATION VARIABLES
C          D(I)=BIOFILM DIFFUSION COEFFICIENT                         -M2/H-
C          KLBIO(I)=LIQUID-BIOFILM TRANSFER COEFFICIENT FOR I         -1/H.M2-
C          KLGAZ(I)=GAS-LIQUID TRANSFER COEFFICIENT FOR I             -1/H.M2-
C          ASPGAZ=SPECIFIC GAS-LIQUID EXCHANGE AREA                   -M2-
C          ASPBIO=SPECIFIC LIQUID-BIOFILM EXCHANGE AREA               -M2-

C       BIOLOGICAL REACTIONS (MOL/L.H EXEPTED BIOMASS IN G/L.H)
C          RXNS(I)=BIOMASS PRODUCTION FOR NITROSOMONAS
C          RXNS(I)=BIOMASS PRODUCTION FOR NITROBACTER
C          RSNS(I)=REACTION RATE OF I FOR NITROSOMONAS
C          RSNB(I)=REACTION RATE OF I FOR NITROBACTER
C          MUMAX(R)=SPECIFIC GROW RATE OR MAINTENANCE FOR REACTION R
C          KSNS(I)=SATURATION CONSTANT FOR I IN NITROSOMONAS
C          KSNB(I)=SATURATION CONSTANT FOR I IN NITROBACTER
C          KINS(I)=INHIBITORY CONSTANT FOR I IN NITROSOMONAS
C          KINB(I)=INHIBITORY CONSTANT FOR I IN NITROBACTER
C          BWO=FRACTION OF BIOMASS REALEASED FROM THE BEADS

C       PH EQUILIBRIA  -ADIM-
C          KA(I)=EQUILIBRIUM CONSTANT FOR I (DEFINED AT T)


C       STOECHIOMETRIES -MOL-
C          STO(R,I)=STOICHIOMETRIC COEFICIENT OF REACTION R
C                          (<0 FOR SUBSTRATES AND >0 FOR PRODUCTS)


C-------------------------------------------------------------------------
C                          COMPOUNDS INVOLVED
C-------------------------------------------------------------------------
C NON EXHAUTIVE LIST (WARNING: NUMBERS ARE THOSE USED TO IDENTIFIED THE
C COMPOUND IN CALCOL AND FILM SUBROUTINE)
C          1-NH3
C          2-HNO3
C          3-HNO2
C          4-CO2
C          5-O2
C          6-H20
C          7-H2SO4
C          8-H3PO4
C          9-FREE BIOMASSE NS
C          10-FREE BIOMASSE NB
C          11-H+
C          12-OH-
C          13-NH4+
C          14-NO3-
C          15-NO2-
C          16-HCO3-
C          17-CO32-
C          18-HSO4-
C          19-SO42-
C          20-H2PO4-
C          21-HPO4-
C          22-PO42-
C          23 A 30 NOT USED ACTUALLY

C-------------------------------------------------------------------------
C              BIOLOGICAL REACTIONS INVOLVED
C-------------------------------------------------------------------------
```

```
C          1-CROISSANCE NS  (ANABOLISM+ENERGY)
C          2-MAINTENANCE NS(MAINTENANCE+ENERGY)
C          3-CROISSANCE NB  (ANABOLISM+ENERGY)
C          4-MAINTENANCE NB(MAINTENANCE+ENERGY)


C////////////////////////////////////////////////////////////////////////
C////////                    VARIABLE DECLARATION                  ///////
C////////////////////////////////////////////////////////////////////////
      IMPLICIT REAL*8 (A-H,O-Z)

      INTEGER NMAX,CORPMAX,TMPSMAX,RMAX
      PARAMETER(NMAX=20)
      PARAMETER(CORPMAX=30)
      PARAMETER(TMPSMAX=300)
      PARAMETER(RMAX=4)

      INTEGER N,CORP,TMPS,REAC
      INTEGER CHOIX,MENU

      REAL*8 REHL,REHG,EPSL,EPSG,EPS,F,G,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      REAL*8 LONG,SECT,DELTAP,DCOL,DH
      REAL*8 T,P,PHINI,PHA,PHC
      REAL*8 PHB(NMAX)
      REAL*8 MUMAX,KI,KS,KA,KLGAZ,KLBIO
      REAL*8 RO
      REAL*8 ASPGAZ,ASPBIO


      DIMENSION HBIO(NMAX),RNBIO(NMAX)
      DIMENSION CSAT(CORPMAX)
      DIMENSION D(CORPMAX),KLBIO(CORPMAX),KLGAZ(CORPMAX)
      DIMENSION KS(RMAX,CORPMAX),KI(RMAX,CORPMAX)
      DIMENSION MUMAX(RMAX)
      DIMENSION KA(CORPMAX)
      DIMENSION STO(RMAX,CORPMAX)
      DIMENSION CINL(CORPMAX),CING(CORPMAX)
      DIMENSION YIM1(650)

      CHARACTER*6 NOM$(CORPMAX)

C////////////////////////////////////////////////////////////////////////
C////////                      COMMON LIST                         ///////
C////////////////////////////////////////////////////////////////////////

      COMMON/REACBIO2/MUMAX,KS,KI,STO
      COMMON/PHYINI/T,P,PHINI
      COMMON/PHYPH/KA
      COMMON/PHYPH2/PHA,PHC,PHB
      COMMON/PHYTRANS/D,KLBIO,KLGAZ,ASPGAZ,ASPBIO
      COMMON/PHYTRANS2/CSAT
      COMMON/BILLE/RO,HBIO,RNBIO,BWO
      COMMON/COLON/EPSL,EPSG,EPS,FINP,GINP,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      COMMON/COLON2/REHL,REHG,LONG,DCOL,SECT,DELTAP,DH
      COMMON/TAB/N,CORP,TMPS,REAC
      COMMON/TAB2/NOM$
      COMMON/CINI/CINL,CING
      COMMON/RKMERY/XIM1,YIM1

C////////////////////////////////////////////////////////////////////////
C///////                   ROUTINE OF MAIN PAGE                   ////////
C///////          MENU FOR THE MANAGEMENT OF THE PROGRAM          ////////
C///////                                                          ////////
C//////              ------- INITIALISATION --------              ////////
C////////////////////////////////////////////////////////////////////////

98    WRITE(*,1100)
      WRITE(*,1001)
      WRITE(*,1002)'            Fixed Bed Nitrification Simulation'
      WRITE(*,1006)
      WRITE(*,1002)'     **    *  * ******* ******  * ****** * **    ** '
      WRITE(*,1002)'     * *   *  *    *     *     *  * **     * * *** * '
      WRITE(*,1002)'     *  *  *  *    *     ******  *  ***    * *     * '
      WRITE(*,1002)'     *   * *  *    *     *     * *     ** * *     * '
      WRITE(*,1002)'     *    **  *    *     *     * * ****** * *     * '
      WRITE(*,1006)
```

```
      WRITE(*,1002)'  SKYSoft                              Version 2.3 PC'
      WRITE(*,1002)'  Laboratoire Genie Chimique Biologique                '
      WRITE(*,1003)
      PRINT*
      PRINT*
      WRITE(*,1001)
      WRITE(*,1002)'1 - Process simulation [No biofilm limitation]'
      WRITE(*,1002)'2 - Biofilm profile in steady state conditions'
      WRITE(*,1002)'3 - Column and kinetic parameters management '
      WRITE(*,1002)'4 - Quit NitriSim'
      WRITE(*,1003)
      PRINT*
      MENU=1
      CALL CONFIGSIM(MENU)
      WRITE(*,1005)
99    READ*,CHOIX
      GOTO (100,400,200,300),CHOIX
      GOTO 99


C//////////////////////////////////////////////////////////////////////////
C///////  Call Simulation routine                                       ///////
C//////////////////////////////////////////////////////////////////////////
100   CALL CALCCOL()
      GOTO 98


C//////////////////////////////////////////////////////////////////////////
C///////        Call listing and management routines            ////////
C//////////////////////////////////////////////////////////////////////////
200   CALL EDITCONF()
      GOTO 98

C//////////////////////////////////////////////////////////////////////////
C///////            Call biofilm simulation routine             ////////
C//////////////////////////////////////////////////////////////////////////
400   CALL BIOFILM()
      GOTO 98


C//////////////////////////////////////////////////////////////////////////
C///////                        FORMATS                           ///////
C//////////////////////////////////////////////////////////////////////////
1001  FORMAT(1x,'É',77('Í'),'»')
1002  FORMAT(1x,'°',T15,A,T80,'°')
1003  FORMAT(1x,'È',77('Í'),'¼')
1004  FORMAT()
1005  FORMAT(1x,5X,'Choice : ',\)
1006  FORMAT(1x,'°',T80,'°')
1100  FORMAT(24(/))


300   END
```

```
       SUBROUTINE SAVECONF(MENU)
C////////////////////////////////////////////////////////////////////////
C///////                SAVE THE DEFAULT FILE .DAT                  ///////
C///////   OF THE COLUMN WORKING  AND DESIGNPARAMETERS              ///////
C///////                        NITRISIM                            ///////
C////// V2.3                                          UPDATE 03/97 ////////
C////////////////////////////////////////////////////////////////////////
C-----------------------------------------------------------------------
C            DECLARATIONS
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)

      INTEGER NMAX,CORPMAX,TMPSMAX,RMAX
      PARAMETER(NMAX=20)
      PARAMETER(CORPMAX=30)
      PARAMETER(TMPSMAX=300)
      PARAMETER(RMAX=4)

      INTEGER N,CORP,TMPS,REAC
      INTEGER CHOIX,MENU
      REAL*8 REHL,REHG,EPSL,EPSG,EPS,FINP,GINP,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      REAL*8 LONG,SECT,DELTAP,DCOL,DH
      REAL*8 T,P,PHINI,PHA,PHC
      REAL*8 PHB(NMAX)
      REAL*8 MUMAX,KI,KS,KA,KLGAZ,KLBIO
      REAL*8 RO
      REAL*8 ASPGAZ,ASPBIO
      DIMENSION D(CORPMAX),KLBIO(CORPMAX),KLGAZ(CORPMAX)

      DIMENSION KS(RMAX,CORPMAX),KI(RMAX,CORPMAX)
      DIMENSION MUMAX(RMAX)


      DIMENSION KA(CORPMAX)
      DIMENSION CSAT(CORPMAX)
      DIMENSION STO(RMAX,CORPMAX)

      CHARACTER*6 NOM$(CORPMAX)


      COMMON/REACBIO2/MUMAX,KS,KI,STO
      COMMON/PHYINI/T,P,PHINI
      COMMON/PHYPH/KA
      COMMON/PHYPH2/PHA,PHC,PHB
      COMMON/PHYTRANS/D,KLBIO,KLGAZ,ASPGAZ,ASPBIO
      COMMON/PHYTRANS2/CSAT
      COMMON/BILLE/RO,HBIO,RNBIO,BWO
      COMMON/COLON/EPSL,EPSG,EPS,FINP,GINP,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      COMMON/COLON2/REHL,REHG,LONG,DCOL,SECT,DELTAP,DH
      COMMON/TAB/N,CORP,TMPS,REAC
      COMMON/TAB2/NOM$

C-----------------------------------------------------------------------
C     ORIGIN OF CALL
C-----------------------------------------------------------------------
       GOTO(100,200,300,400),MENU



C-----------------------------------------------------------------------
C     SAVE PARAMETERS OF CARCOL.DAT ET FLOWCOL.DAT
C-----------------------------------------------------------------------
100    OPEN (1,FILE='CARCOL.DAT',FORM='FORMATTED',STATUS='OLD')
        WRITE(1,*)LONG
        WRITE(1,*)DCOL
        WRITE(1,*)VA
        WRITE(1,*)VB
        WRITE(1,*)VC
        WRITE(1,*)EPS
        WRITE(1,*)RO
        WRITE(1,*)T
        WRITE(1,*)P
        WRITE(1,*)PHINI
        WRITE(1,*)BWO
      CLOSE(1)
      OPEN (1,FILE='FLOWCOL.DAT',FORM='FORMATTED',STATUS='OLD')
```

```
          WRITE(1,*)FINP
          WRITE(1,*)GINP
          WRITE(1,*)RL
          WRITE(1,*)RG
          WRITE(1,*)FBAK
          WRITE(1,*)FBAKPRIM
          WRITE(1,*)N
        CLOSE(1)
        RETURN


C-------------------------------------------------------------------------
C     SAVE PARAMETERS OF STOIC.DAT
C-------------------------------------------------------------------------

200   OPEN (1,FILE='STOIC.DAT',FORM='FORMATTED',STATUS='OLD')
        WRITE(1,*)REAC
        DO 97 I=1,REAC
          DO 96 J=1,CORP
           WRITE(1,*)STO(I,J)
96        CONTINUE
97      CONTINUE
      CLOSE(1)
      RETURN


C-------------------------------------------------------------------------
C     SAVE PARAMETERS OF KINETICS LAWS
C-------------------------------------------------------------------------

300   OPEN (1,FILE='CINET.DAT',FORM='FORMATTED',STATUS='OLD')
        DO 94 J=1,REAC
           WRITE(1,*)MUMAX(J)
        DO 93 I=1,CORP
           WRITE(1,*)KS(J,I)
           WRITE(1,*)KI(J,I)
93      CONTINUE
94    CONTINUE
      CLOSE(1)
      RETURN


C-------------------------------------------------------------------------
C     SAVE PARAMETERS OF CORPS.DAT AND PHYTPH/TRANS.DAT
C-------------------------------------------------------------------------

400   OPEN (1,FILE='PHYPH.DAT',FORM='FORMATTED',STATUS='OLD')
        DO 98 I=1,CORP
           WRITE(1,*)KA(I)
98      CONTINUE
      CLOSE(1)



      OPEN (1,FILE='PHYTRANS.DAT',FORM='FORMATTED',STATUS='OLD')
        DO 95 I=1,CORP
           WRITE(1,*)D(I)
           WRITE(1,*)KLBIO(I)
           WRITE(1,*)KLGAZ(I)
           WRITE(1,*)CSAT(I)
95      CONTINUE
      CLOSE(1)

      OPEN (1,FILE='CORPS.DAT',FORM='FORMATTED',STATUS='OLD')
        WRITE(1,*)CORP
        DO 99 I=1,CORP
          WRITE(1,'(A)')NOM$(I)
99      CONTINUE
      CLOSE(1)

C-------------------------------------------------------------------------
       RETURN
       END
```

```
      SUBROUTINE SAVEPARA()
C////////////////////////////////////////////////////////////////////
C///////                SAVE RESULTS FILE .CNF                  ////////
C///////              PARAMETERS OF THE SIMULATION              ////////
C///////                        NITRISIM                        ////////
C/////// V2.3                                  UPDATE 03/97 /////////
C////////////////////////////////////////////////////////////////////
C--------------------------------------------------------------------
C            DECLARATIONS
C--------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)

      INTEGER NMAX,CORPMAX,TMPSMAX,RMAX
      PARAMETER(NMAX=20)
      PARAMETER(CORPMAX=30)
      PARAMETER(TMPSMAX=300)
      PARAMETER(RMAX=4)

      INTEGER N,CORP,TMPS,REAC

      INTEGER INI

      REAL*8 REHL,REHG,EPSL,EPSG,EPS,FINP,GINP,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      REAL*8 LONG,SECT,DELTAP,DCOL,DH
      REAL*8 T,P,PHINI,PHA,PHC
      REAL*8 PHB(NMAX)
      REAL*8 MUMAX,KI,KS,KA,KLGAZ,KLBIO
      REAL*8 RO,TN

      REAL*8 ASPGAZ,ASPBIO
      DIMENSION D(CORPMAX),KLBIO(CORPMAX),KLGAZ(CORPMAX)

      DIMENSION KS(RMAX,CORPMAX),KI(RMAX,CORPMAX)
      DIMENSION MUMAX(RMAX)


      DIMENSION KA(CORPMAX)
      DIMENSION CSAT(CORPMAX),CING(CORPMAX),CINL(CORPMAX)
      DIMENSION STO(RMAX,CORPMAX)

      CHARACTER*6 NOM$(CORPMAX)


      COMMON/REACBIO2/MUMAX,KS,KI,STO
      COMMON/PHYINI/T,P,PHINI
      COMMON/PHYPH/KA
      COMMON/PHYPH2/PHA,PHC,PHB
      COMMON/PHYTRANS/D,KLBIO,KLGAZ,ASPGAZ,ASPBIO
      COMMON/PHYTRANS2/CSAT
      COMMON/BILLE/RO,HBIO,RNBIO,BWO
      COMMON/COLON/EPSL,EPSG,EPS,FINP,GINP,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      COMMON/COLON2/REHL,REHG,LONG,DCOL,SECT,DELTAP,DH
      COMMON/TAB/N,CORP,TMPS,REAC
      COMMON/TAB2/NOM$
      COMMON/CINI/CINL,CING


       WRITE(3,*)'#### -- New simulations Conditions -- ####'
       WRITE(3,*)
       WRITE(3,*)'#### Column design ####'
        WRITE(3,1000)'Height', LONG,'m'
        WRITE(3,1000)'Diameter',DCOL,'m'
        WRITE(3,1000)'Bottom Volume',VA,'m3'
        WRITE(3,1000)'Bed volume',VB,'m3'
        WRITE(3,1000)'Top volume',VC,'m3'
        WRITE(3,1000)'Temperature',T,'°C'
        WRITE(3,1000)'Pressure',P,'atm'
        WRITE(3,1000)'PH',PHINI,' '
        WRITE(3,1000)'Biomass Washinng',BWO,' '
        WRITE(3,*)
        WRITE(3,*)

       WRITE(3,*)'#### Flow rates ####'
        WRITE(3,1000)'Liquid Flow Rate',FINP,'m3/h'
        WRITE(3,1000)'Gas Flow Rate',GINP,'m3/h'
        WRITE(3,1000)'Liquid recycling ratio',RL,' '
```

```
          WRITE(3,1000)'Gas recycling ratio',RG,' '
          WRITE(3,*)
          WRITE(3,*)

         WRITE(3,*)'#### Fixed Bed design ####'
          WRITE(3,1000)'Liquid Backmixing',FBAK,' '
          WRITE(3,1000)'Gas Backmixing',FBAKPRIM,' '
          TN=float(N)
          WRITE(3,1000)'Number of tanks',TN,' '
          WRITE(3,1000)' Bead radius',RO,'m'
          WRITE(3,1000)'column empty degree',EPS,' '
          WRITE(3,*)
          WRITE(3,*)

         WRITE(3,*)'#### Liquid Input composition mol/l ####'
          DO 90 I=1,CORP
             WRITE(3,1001)NOM$(I)
90        CONTINUE
          WRITE(3,*)
          DO 91 I=1,CORP
             WRITE(3,1002)CINL(I)
91        CONTINUE
          WRITE(3,*)
          WRITE(3,*)
          WRITE(3,*)

         WRITE(3,*)'#### Gas Input composition fraction ####'
          DO 92 I=1,CORP
             WRITE(3,1001)NOM$(I)
92        CONTINUE
          WRITE(3,*)
          DO 93 I=1,CORP
             WRITE(3,1002)CING(I)
93        CONTINUE
          WRITE(3,*)
          WRITE(3,*)


1000    FORMAT(A20,'  ',G13.6,'  ',A5)
1001    FORMAT(A8,'    ',\)
1002    FORMAT(F10.3,'  ',\)
C-------------------------------------------------------------------------------
        RETURN
        END
```

```
      SUBROUTINE SAVESIM(X,Y,K)
C////////////////////////////////////////////////////////////////////
C///////                SAVE RESULTS IN FILE .RES                ////////
C///////              RESULTS OF CURRENT SIMULATION              ////////
C///////     X=TIME (H)     Y(I)=RESULT VECTOR (CALCOL.FOR)      ////////
C//////// V 2.3                                    UPDATE 03/97 ////////
C////////////////////////////////////////////////////////////////////
C--------------------------------------------------------------------
C            DECLARATIONS
C--------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER NMAX,CORPMAX
      PARAMETER (NMAX=20)
      PARAMETER(CORPMAX=30)
      REAL*8 LONG
      REAL*8 RO
      REAL*8 THICKLIM

      REAL*8 ASPGAZ,ASPBIO
      DIMENSION D(CORPMAX),KLBIO(CORPMAX),KLGAZ(CORPMAX)

      INTEGER N,CORP,TMPS,REAC
      DIMENSION Y(650)
      DIMENSION HBIO(NMAX),RNBIO(NMAX)
      COMMON/COLON/EPSL,EPSG,EPS,FINP,GINP,RL,RG,VA,VB,VC,FBAK,FBAKPRIM
      COMMON/COLON2/REHL,REHG,LONG,DCOL,SECT,DELTAP,DH
      COMMON/TAB/N,CORP,TMPS,REAC
      COMMON/PHYINI/T,P,PHINI
      COMMON/BILLE/RO,HBIO,RNBIO,BWO
      COMMON/PHYTRANS/D,KLBIO,KLGAZ,ASPGAZ,ASPBIO
c--------------------------------------------------------------------
C-- UNIT A
C-

      COLIND=0.
      WRITE(1,1001)X
      WRITE(1,1001)COLIND+1.
      DO 100 I=1,10
         WRITE(1,1001)Y(I)
100    CONTINUE
      DO 101 I=1,10
         WRITE(1,1001)Y(10+I)*(8.314*(273.15+T)/(P*101.3))
101    CONTINUE
      WRITE(1,1001)0.00
      WRITE(1,1001)0.00
      WRITE(1,1001)Y(10*2*(N+2)+2*N+1)
      WRITE(1,1001)VA/(VA+VB+VC)*LONG
      WRITE(1,*)

C--------------------------------------------------------------------
C-- SEGMENT N
C-
      BIOTOT=0.
      BIOTOTNS=0.
      DO 102 J=1,N
         INDIC=10*2*J
         INDBIO=10*2*(N+2)
         WRITE(1,1001)X
         WRITE(1,1001)COLIND+FLOAT(J)+1.
         DO 103 I=1,10
           WRITE(1,1001)Y(INDIC+I)
103      CONTINUE
         DO 104 I=1,10
           WRITE(1,1001)Y(INDIC+10+I)*(8.314*(273.15+T)/(P*101.3))
104      CONTINUE
         WRITE(1,1001)Y(INDBIO+J)
         WRITE(1,1001)Y(INDBIO+N+J)
         WRITE(1,1001)Y(10*2*(N+2)+2*N+J+1)
         WRITE(1,1001)VA/(VA+VB+VC)*LONG+J*(VB/(VA+VB+VC)*LONG)/N
         HBIO(J)=(Y(INDBIO+J)+Y(INDBIO+J+N))*EPSL*RO/2/((1-eps)*3)*1e6
         HBIO(J)=HBIO(J)*3.7e-3
         WRITE(1,1001)HBIO(J)

C- BIOFILM THICKNEES FOR OXYGEN LIMITATION
C- diffusion o2 : en m2 s-1
C- Thicklim given in m
```

```
          IF (RNBIO(J).NE.0.) THEN
             THICKLIM=DSQRT(2*Y(INDIC+5)*D(5)*3600/DABS(RNBIO(J)))
             THICKLIM=THICKLIM/3
          ELSE
             THICKLIM=999.
          ENDIF
          WRITE(1,1001)THICKLIM*1e6
          WRITE(1,*)

          BIOTOT=BIOTOT+Y(INDBIO+J)+Y(INDBIO+N+J)
          BIOTOTNS=Y(INDBIO+J)+BIOTOTNS
102    CONTINUE
          BIOTOT=BIOTOT/N
C-------------------------------------------------------------------------
C-- UNIT C
C-
       WRITE(1,1001)X
       WRITE(2,1001)X
       WRITE(1,1001)COLIND+FLOAT(N)+2.0
       INDIC=10*2*(N+1)
       DO 110 I=1,10
          WRITE(1,1001)Y(INDIC+I)
          WRITE(2,1001)Y(INDIC+I)
110    CONTINUE
       DO 111 I=1,10
          WRITE(1,1001)Y(INDIC+10+I)*(8.314*(273.15+T)/(P*101.3))
          WRITE(2,1001)Y(INDIC+10+I)*(8.314*(273.15+T)/(P*101.3))
111     CONTINUE
       WRITE(1,1001)0.00
       WRITE(1,1001)0.00
       WRITE(2,1001)BIOTOT
       WRITE(2,1001)BIOTOTNS*100/BIOTOT/N
       WRITE(1,1001)Y(10*2*(N+2)+2*N+N+2)
       WRITE(2,1001)Y(10*2*(N+2)+2*N+N+2)
       WRITE(1,1001)LONG
       WRITE(1,*)
       WRITE(1,*)
       WRITE(2,*)

1001   FORMAT(G10.4,'    ',\)


       RETURN
       END
```

```
C////////////////////////////////////////////////////////////////////////
C                  FRENCH VERSION OF RKMER WRITTEN BY ENSIGC
C                  MODIFIED FOR NITRISIM V 2.3
C////////////////////////////////////////////////////////////////////////
      SUBROUTINE DRKMER(X0,XF,Y0,N,ITAB)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION YI(650),ERR(650)
      DIMENSION Y0(1),ERRMAX(650),XTAB(300)
      REAL*8 K1(650),K3(650),K4(650),K5(650)
      REAL*8 YIM1(650)
      REAL*8 HITER
      COMMON/RKMERY/XIM1,YIM1,HITER

      DATA ERRMAX/650*0.01/
      DATA MODTAB/1/
      HITER=1e-2

      NO=6
      ISTOP=0
      NMAX=2000
      IF(N.GT.NMAX) GO TO 999
      NHS2=0
      XCOMPT=X0/XF*100
      WRITE(*,1010)XCOMPT

C
C       INITIALISATION
C

C       OPEN(5,FILE='RKMER.RES')
C       write(5,*)N

      IK=0
      HTAB=(XF-X0)/FLOAT(ITAB-1)
      IF(MODTAB.EQ.0) GO TO 2
      DO 1 I=1,ITAB
    1 XTAB(I)=X0+HTAB*FLOAT(I-1)
    2 CONTINUE
      DO 3 J=1,N
        YIM1(J)=Y0(J)
    3   YI(J)=Y0(J)
        XIM1=X0
        KTAB=2
        CALL SAVESIM(XIM1,YI,KTAB)
    4 CONTINUE
      IF( ABS(HITER).LT. ABS(HTAB)) GO TO 5
      HITER=HITER/2.
      GO TO 4
    5 XI=XIM1+HITER

C
C     print % Processing
      IF ((XI/XF*100-XCOMPT).GT.0.05) then
        WRITE(*,1010)XI/XF*100
        XCOMPT=XI/XF*100
      ENDIF


C
C       TEST FOR THE STORAGE OF RESULTS
C
      XFI0=(XTAB(KTAB)-XI)/(XF-X0)
      IF(XFI0.LT.1..AND.XFI0.GT.0.)GOTO10
      H0=HITER
      HITER=XTAB(KTAB)-XIM1
      IK=1
C
C       ALGORYTHM OF RUNGE KUTTA MERSON OF th ORDER
C
   10 XI=XIM1
      CALL DERIV(YI,XI,K1)
      XI=XIM1+HITER/3.
      DO 21 J=1,N
         YI(J)=YI(J)+K1(J)*HITER/3.
   21    IF(YI(J).LT.0.) YI(J)=1e-15
```

```
      CALL DERIV(YI,XI,K3)
      DO 23 J=1,N
        YI(J)=YIM1(J) +(K1(J)+K3(J))/2.*HITER/3.
   23   IF(YI(J).LT.0.) YI(J)=1e-21

      CALL DERIV(YI,XI,K3)
      XI=XIM1+HITER/2.
      DO 25 J=1,N
        YI(J)=YIM1(J) +3./8.*(K1(J)+3.*K3(J))*HITER/3.
   25   IF(YI(J).LT.0.) YI(J)=1e-21

      CALL DERIV(YI,XI,K4)
      XI=XIM1+HITER
      DO 27 J=1,N
         YI(J)=YIM1(J) +3./2.*(K1(J)-3.*K3(J)+K4(J)*4.) *HITER/3.
   27    IF(YI(J).LT.0.) YI(J)=1e-21

      CALL DERIV(YI,XI,K5)
      DO 30 J=1,N
        YI(J)=YIM1(J)  +(K1(J)+4.*K4(J)+K5(J))/2. *HITER/3.
        IF(YI(J).LT.0.) YI(J)=1e-21
        E=K1(J)-9./2.*K3(J)+4.*K4(J)-K5(J) /2.
        E=E*HITER/3.
        ERR(J)=0.
        IF(ABS(YI(J) ).GT.1.E-20) ERR(J)=ABS(E/YI(J))
 30   CONTINUE

C
C      NEW STEP
C
      IF(ISTOP.NE.1) GO TO 40
      XF=XI
      DO 35 J=1,N
 35   Y0(J)=YI(J)
      RETURN

   40 CONTINUE
      IC=0
      DO 50 J=1,N
      IF(ERR(J).LT.ERRMAX(J).OR.YI(J).LT.1e-15) GO TO 45
      NHS2=NHS2+1
      IF(NHS2.GT.20) GO TO 995
      HITER=HITER/2.

C      write(5,*)HITER,J

      DO 42 JP=1,N
42    YI(JP)=YIM1(JP)
      IK=0
      GO TO 10

   45 IF(ERR(J).GT.ERRMAX(J)/16. .AND.YI(J).GT.1e-15) IC=1
   50 CONTINUE

      IF(IK.EQ.1) GO TO 60
      IF(IC+NHS2.EQ.0) HITER=HITER*2
      NHS2=0

C
C      REINITIALISATION FOR THE NEXT STEP
C
      XIM1=XI
      DO 55 J=1,N
         YIM1(J)=YI(J)

C---  Verification of negative values (concentrations)
         IF(YI(J).LT.0.) YI(J)=1e-21
55    CONTINUE

      GO TO 5
C
C      Storage of results
C      and reinitialisation
C
   60 IK=0
      HITER=H0
```

```
      DO 65 J=1,N
   65 YIM1(J)=YI(J)
      XIM1=XTAB(KTAB)
      CALL SAVESIM(XIM1,YI,KTAB)
      KTAB=KTAB+1
      IF(KTAB.LE.ITAB) GO TO 5

C-----------------------------------------------------------------------
C-- Reinitialisation of Y0 for a new simulation
C-
      DO 100 j=1,N
         Y0(J)=Yi(J)
100   CONTINUE

C      close(5)

      RETURN
995   WRITE(NO,996) NHS2
996   FORMAT(1X,5(1H*),'Stop in KYKMER after ',I3,'DIVISIONS ',
     1   'of the step')
      CALL DERIV(YIM1,XIM1,K5)
      WRITE(NO,997) X,H,(YIM1(I),I=1,N)
      WRITE(NO,998) (K5(I),I=1,N)
997   FORMAT(' last value of X',G12.4,' last value of Y',/,
     1   5(10G12.4,/))
998   FORMAT(' Last values of derivaties',/,5(10G12.4,/))
      STOP
999   WRITE(NO,9991) NMAX
9991  FORMAT(1X,5(1H*),'ERROR IN RKMER * MORE THAN ',I3,' EQUATIONS',
     1   50(1H*))
1010  FORMAT('+',F5.2,t8,'%')
      STOP
      END
```

```
      SUBROUTINE WEG(X,Y,XMIN,XMAX,XA,YA,NR,NC,PRE,SCRIT,NCRIT,INDIC)
C*************************************************************************
C     This subroutine solves a system a N equations of the form
C                              X=f(X)
C     The algorythm used is this of WEGSTEIN
C*************************************************************************
C     ENTRY :
C     --------------------
C     Y : Vector of the N equation F(X).
C     PRE : PRECISION FOR THE TEST OF CONVERGENCE.
C     NR : Number of equations.
C*************************************************************************
C     OUTPUTS :
C     ---------------------
C     NC : Convergence (=1) or not
C*************************************************************************
C     INPUTS/OUTPUTS :
C     ---------------------------
C     X : Vector of the variables to fit
C     XA,YA : working vector
C*************************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER NR,NC,I,INDIC
      REAL*8 X(650),Y(650),XA(650),YA(650),PRE,XT,XMIN(650),
     &XMAX(650)

      SCRIT=0.
      NCRIT=0

C
C     CONVERGENCE TEST
C
      INDIC=0

      DO 10 I=1,NR
      IF(X(I).EQ.0.AND.Y(I).EQ.0) THEN
      INDIC=INDIC+1
      ELSE
        CRIT=DABS((X(I)-Y(I))/(X(I)+Y(I)))
        IF (CRIT.LT.PRE) INDIC=INDIC+1
        IF (CRIT.GT.SCRIT) THEN
           SCRIT=CRIT
           NCRIT=I
        ENDIF
      ENDIF
  10  CONTINUE

C
C     Output test
C
      IF(INDIC.EQ.NR) THEN
      NC=1
      DO 20 I=1,NR
      X(I)=Y(I)
  20  CONTINUE
      RETURN
      ENDIF
C
C     Tesy for the first step
C
      IF(NC.LE.1) THEN
      DO 30 I=1,NR
      XA(I)=X(I)
      YA(I)=Y(I)
      X(I)=Y(I)
  30  CONTINUE
      NC=2
      RETURN
      ELSE
C
C     CONVERGENCE
C
      DO 40 I=1,NR
      IF((XA(I)-X(I)+Y(I)-YA(I)).NE.0) THEN
      XT=(XA(I)*Y(I)-YA(I)*X(I))/(XA(I)-X(I)+Y(I)-YA(I))
C
```

```
C     TEST DE VALEUR BORNEE
C     SPECIFIQUE AU PROBLEME
C     X(I)>=0. (CONCENTRATION)
C
        IF (XT.LT.XMIN(I)) XT=(X(I)+XMIN(I))/2
        IF (XT.GT.XMAX(I)) XT=(X(I)+XMAX(I))/2
C
C     End of TEST
C
      XA(I)=X(I)
      YA(I)=Y(I)
      X(I)=XT
      ELSE
      X(I)=Y(I)
      ENDIF
   40 CONTINUE
      ENDIF
      RETURN
      END
```

APPENDIX 2

Results files of NitriSim

File base1500.RES

# standart	1500h v2.3												
# Time	Unit	LiqNH3	LiqHNO3	LiqHNO2	LiqCO2	LiqO2	LiqH2O	LiqH2SO4	LiqH3PO4	LiqBioNS	LiqBioNB	GasNH3	GasHNO3
0.0000E+00	1.000	0.7140E-02	0.0000E+00	0.0000E+00	0.6035E-02	0.2731E-03	55.56	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
0.0000E+00	2.000	0.7140E-02	0.0000E+00	0.0000E+00	0.6035E-02	0.2731E-03	55.56	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
0.0000E+00	3.000	0.7140E-02	0.0000E+00	0.0000E+00	0.6035E-02	0.2731E-03	55.56	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
0.0000E+00	4.000	0.7140E-02	0.0000E+00	0.0000E+00	0.6035E-02	0.2731E-03	55.56	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
0.0000E+00	5.000	0.7140E-02	0.0000E+00	0.0000E+00	0.6035E-02	0.2731E-03	55.56	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
0.0000E+00	6.000	0.7140E-02	0.0000E+00	0.0000E+00	0.6035E-02	0.2731E-03	55.56	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
0.0000E+00	7.000	0.7140E-02	0.0000E+00	0.0000E+00	0.6035E-02	0.2731E-03	55.56	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
[Intermétiates values deleted]													
792.9	1.000	0.7977E-03	0.6216E-02	0.1261E-03	0.1133E-02	0.2220E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
792.9	2.000	0.4704E-03	0.6518E-02	0.1510E-03	0.1147E-02	0.1421E-03	55.55	0.1000E-03	0.9999E-04	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
792.9	3.000	0.4242E-03	0.6595E-02	0.1205E-03	0.1160E-02	0.2005E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
792.9	4.000	0.4084E-03	0.6623E-02	0.1090E-03	0.1168E-02	0.2173E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
792.9	5.000	0.4009E-03	0.6636E-02	0.1037E-03	0.1175E-02	0.2218E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
792.9	6.000	0.3974E-03	0.6642E-02	0.1012E-03	0.1179E-02	0.2231E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
792.9	7.000	0.3974E-03	0.6642E-02	0.1012E-03	0.1181E-02	0.2245E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
[Intermétiates values deleted]													
1500.	1.000	0.8049E-03	0.6207E-02	0.1282E-03	0.1133E-02	0.2220E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
1500.	2.000	0.4656E-03	0.6530E-02	0.1447E-03	0.1148E-02	0.1346E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
1500.	3.000	0.4392E-03	0.6574E-02	0.1266E-03	0.1159E-02	0.2047E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
1500.	4.000	0.4318E-03	0.6587E-02	0.1215E-03	0.1167E-02	0.2203E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
1500.	5.000	0.4286E-03	0.6592E-02	0.1195E-03	0.1173E-02	0.2235E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
1500.	6.000	0.4271E-03	0.6594E-02	0.1186E-03	0.1178E-02	0.2243E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
1500.	7.000	0.4271E-03	0.6594E-02	0.1186E-03	0.1180E-02	0.2249E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00

Data to use for initialisation of a process from a file
[copy and paste into a new file]

GasHNO2	GasCO2	GasO2	GasH2O	GasH2SO4	GasH3PO4	GasBioNS	GasBioNB	Ns fix	Nb fix	H+	Height	Bio Thick	Thick Lim
0.0000E+00	0.4000E-02	0.2100	0.3100E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E-07	0.1326		
0.0000E+00	0.4000E-02	0.2100	0.3100E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.7000E-01	0.3000E-01	0.1000E-07	0.2414	0.1343	0.9990E+08
0.0000E+00	0.4000E-02	0.2100	0.3100E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.7000E-01	0.3000E-01	0.1000E-07	0.3501	0.1343	0.9990E+08
0.0000E+00	0.4000E-02	0.2100	0.3100E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.7000E-01	0.3000E-01	0.1000E-07	0.4589	0.1343	0.9990E+08
0.0000E+00	0.4000E-02	0.2100	0.3100E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.7000E-01	0.3000E-01	0.1000E-07	0.5677	0.1343	0.9990E+08
0.0000E+00	0.4000E-02	0.2100	0.3100E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.7000E-01	0.3000E-01	0.1000E-07	0.6764	0.1343	0.9990E+08
0.0000E+00	0.4000E-02	0.2100	0.3100E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1000E-07	0.7162		
0.0000E+00	0.8082E-03	0.1732	0.3065E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.6342E-02	0.1326		
0.0000E+00	0.8040E-03	0.1730	0.3069E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.8041	0.2834	0.6669E-02	0.2414	1.461	0.1610
0.0000E+00	0.8008E-03	0.1729	0.3072E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1056	0.7635E-01	0.6717E-02	0.3501	0.2444	0.1989
0.0000E+00	0.7985E-03	0.1729	0.3075E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.3320E-01	0.2671E-01	0.6733E-02	0.4589	0.8048E-0	1 0.2109
0.0000E+00	0.7967E-03	0.1729	0.3079E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1594E-01	0.1294E-01	0.6741E-02	0.5677	0.3881E-0	1 0.2150
0.0000E+00	0.7954E-03	0.1729	0.3079E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1067E-01	0.8600E-02	0.6745E-02	0.6764	0.2588E-0	1 0.2167
0.0000E+00	0.7943E-03	0.1729	0.3085E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.6745E-02	0.7162		
0.0000E+00	0.8078E-03	0.1734	0.3065E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.6335E-02	0.1326		
0.0000E+00	0.8036E-03	0.1731	0.3069E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.8773	0.3401	0.6674E-02	0.2414	1.636	0.1581
0.0000E+00	0.8005E-03	0.1731	0.3072E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.6137E-01	0.4508E-01	0.6701E-02	0.3501	0.1430	0.1984
0.0000E+00	0.7982E-03	0.1731	0.3075E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1536E-01	0.1156E-01	0.6709E-02	0.4589	0.3618E-0	1 0.2072
0.0000E+00	0.7964E-03	0.1730	0.3078E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.6603E-02	0.4837E-02	0.6712E-02	0.5677	0.1537E-0	1 0.2094
0.0000E+00	0.7950E-03	0.1730	0.3080E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.4190E-02	0.3001E-02	0.6714E-02	0.6764	0.9661E-0	2 0.2101
0.0000E+00	0.7938E-03	0.1730	0.3082E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.6714E-02	0.7162		

Data to use for initialisation of a process from a file
[copy and paste into a new file]

File BFM1500.DAT

[File for initialisation of biofilm profile taken from file BASE1500.RES]

Number of tanks for the fixed bed (part B)

5

0.4656E-03	0.6530E-02	0.1447E-03	0.1148E-02	0.1346E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
0.4392E-03	0.6574E-02	0.1266E-03	0.1159E-02	0.2047E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
0.4318E-03	0.6587E-02	0.1215E-03	0.1167E-02	0.2203E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
0.4286E-03	0.6592E-02	0.1195E-03	0.1173E-02	0.2235E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
0.4271E-03	0.6594E-02	0.1186E-03	0.1178E-02	0.2243E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00

Same as file INI1500.DAT without the first and the last line (i.e. descritoion of bottom and top of the column)

0.0000E+00	0.8036E-03	0.1731	0.3069E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.8773	0.3401	0.6674E-02
0.0000E+00	0.8005E-03	0.1731	0.3072E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.6137E-01	0.4508E-01	0.6701E-02
0.0000E+00	0.7982E-03	0.1731	0.3075E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.1536E-01	0.1156E-01	0.6709E-02
0.0000E+00	0.7964E-03	0.1730	0.3078E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.6603E-02	0.4837E-02	0.6712E-02
0.0000E+00	0.7950E-03	0.1730	0.3080E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.4190E-02	0.3001E-02	0.6714E-02

File INI1500.DAT

[File for initialisation of a dynamic simulation taken from file BASE1500.RES]

0.8049E-03	0.6207E-02	0.1282E-03	0.1133E-02	0.2220E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00
0.4656E-03	0.6530E-02	0.1447E-03	0.1148E-02	0.1346E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00
0.4392E-03	0.6574E-02	0.1266E-03	0.1159E-02	0.2047E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00
0.4318E-03	0.6587E-02	0.1215E-03	0.1167E-02	0.2203E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00
0.4286E-03	0.6592E-02	0.1195E-03	0.1173E-02	0.2235E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00
0.4271E-03	0.6594E-02	0.1186E-03	0.1178E-02	0.2243E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00
0.4271E-03	0.6594E-02	0.1186E-03	0.1180E-02	0.2249E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00

0.0000E+00	0.8078E-03	0.1734	0.3065E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.6335E-02
0.0000E+00	0.8036E-03	0.1731	0.3069E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.8773	0.3401	0.6674E-02
0.0000E+00	0.8005E-03	0.1731	0.3072E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.6137E-01	0.4508E-01	0.6701E-02
0.0000E+00	0.7982E-03	0.1731	0.3075E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.1536E-01	0.1156E-01	0.6709E-02
0.0000E+00	0.7964E-03	0.1730	0.3078E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.6603E-02	0.4837E-02	0.6712E-02
0.0000E+00	0.7950E-03	0.1730	0.3080E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.4190E-02	0.3001E-02	0.6714E-02
0.0000E+00	0.7938E-03	0.1730	0.3082E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.6714E-02

File Base1500.FIN

# standart	1500h v2.3										
# Time	LiqNH3	LiqHNO3	LiqHNO2	LiqCO2	LiqO2	LiqH2O	LiqH2SO4	LiqH3PO4	LiqBioNS	LiqBioNB	GasNH3
0.0000E+00	0.7140E-02	0.0000E+00	0.0000E+00	0.6035E-02	0.2731E-03	55.56	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00
[Intermediate values deleted]											
792.9	0.3974E-03	0.6642E-02	0.1012E-03	0.1181E-02	0.2245E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00
[Intermediate values deleted]											
1500.	0.4271E-03	0.6594E-02	0.1186E-03	0.1180E-02	0.2249E-03	55.55	0.1000E-03	0.1000E-03	0.0000E+00	0.0000E+00	0.0000E+00

GasHNO3	GasHNO2	GasCO2	GasO2	GasH2O	GasH2SO4	GasH3PO4	GasBioNS	GasBioNB	Bio mean	Pop mean	H+
0.0000E+00	0.0000E+00	0.4000E-02	0.2100	0.3100E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.1000	70.00	0.1000E-07
0.0000E+00	0.0000E+00	0.7943E-03	0.1729	0.3085E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.2755	70.38	0.6745E-02
0.0000E+00	0.0000E+00	0.7938E-03	0.1730	0.3082E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.2739	70.45	0.6714E-02

-- New simulations Conditions --

Column design
Height 0.716200 m
Diameter 0.120000 m
Bottom Volume 0.150000E-02 m3
Bed volume 0.615005E-02 m3
Top volume 0.450000E-03 m3
Temperature 25.0000 °C
Pressure 1.00000 atm
PH 8.00000
Biomass Washinng 0.000000E+00

Flow rates
Liquid Flow Rate 0.200000E-03 m3/h
Gas Flow Rate 0.180000E-02 m3/h
Liquid recycling rat 6.42860
Gas recycling ratio 99.0000

Fixed Bed design
Liquid Backmixing 1.55000
Gas Backmixing 0.000000E+00
Number of tanks 5.00000
Bead radius 0.410000E-02 m
column empty degree 0.370000

Liquid Input composition mol/l

NH3	HNO3	HNO2	CO2	O2	H2O	H2SO4	H3PO4	BioNS	BioNB	H+	OH-	NH4+	NO3-	NO2-	HCO3-	CO32-	HSO4-	SO42-	H2PO4-	HPO42-	PO43-
0.007	0.000	0.000	0.000	0.000	55.556	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Gas Input composition fraction

NH3	HNO3	HNO2	CO2	O2	H2O	H2SO4	H3PO4	BioNS	BioNB	H+	OH-	NH4+	NO3-	NO2-	HCO3-	CO32-	HSO4-	SO42-	H2PO4-	HPO42-	PO43-
0.000	0.000	0.000	0.000	0.009	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

GasHNO2	GasCO2	GasO2	GasH2O	GasH2SO4	GasH3PO4	GasBioNS	GasBioNB	Ns fix	Nb fix	H+	Height	Bio Thick	Thick Lim
0,00E+00	2,89E-03	0,175	3,10E-02	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,1326		
0,00E+00	2,89E-03	0,175	3,10E-02	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,7804	0,3159	0,00E+00	0,2414	1,473	0,206
0,00E+00	2,89E-03	0,175	3,10E-02	0,00E+00	0,00E+00	0,00E+00	0,00E+00	1,00E-02	1,00E-02	0,00E+00	0,3501	2,69E-02	0,2063
0,00E+00	2,89E-03	0,175	3,10E-02	0,00E+00	0,00E+00	0,00E+00	0,00E+00	1,00E-02	1,00E-02	0,00E+00	0,4589	2,69E-02	0,2068
0,00E+00	2,89E-03	0,175	3,10E-02	0,00E+00	0,00E+00	0,00E+00	0,00E+00	1,00E-02	1,00E-02	0,00E+00	0,5677	2,69E-02	0,2072
0,00E+00	2,89E-03	0,175	3,10E-02	0,00E+00	0,00E+00	0,00E+00	0,00E+00	1,00E-02	1,00E-02	0,00E+00	0,6764	2,69E-02	0,2073
0,00E+00	2,89E-03	0,175	3,10E-02	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,7162		
0,00E+00	8,08E-04	0,1735	3,07E-02	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	6,33E-03	0,1326		
0,00E+00	8,04E-04	0,1732	3,07E-02	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,9316	0,3773	6,68E-03	0,2414	1,758	0,1552
0,00E+00	8,01E-04	0,1732	3,07E-02	0,00E+00	0,00E+00	0,00E+00	0,00E+00	1,36E-02	1,21E-02	6,69E-03	0,3501	3,44E-02	0,1976
0,00E+00	7,97E-04	0,1732	3,08E-02	0,00E+00	0,00E+00	0,00E+00	0,00E+00	7,83E-03	6,56E-03	6,69E-03	0,4589	1,93E-02	0,2042
0,00E+00	7,96E-04	0,1735	3,08E-02	0,00E+00	0,00E+00	0,00E+00	0,00E+00	5,12E-03	3,82E-03	6,69E-03	0,5677	1,20E-02	0,2057
0,00E+00	7,94E-04	0,1732	3,08E-02	0,00E+00	0,00E+00	0,00E+00	0,00E+00	4,16E-03	3,03E-03	6,69E-03	0,6764	9,65E-03	0,2063
0,00E+00	7,94E-04	0,1732	3,08E-02	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	6,69E-03	0,7162		

Result file of Steady state simulation

# manual for	standart	v2,3											
# Time	Unit		LiqNH3	LiqHNO3	LiqHNO2	LiqCO2	LiqO2	LiqH2O	LiqH2SO4	LiqH3PO4	LiqBioNS	LiqBioNB	GasNH3	GasHNO3
#Initialisalisation values														
0,00E+00		1	2,14E-04	6,78E-03	1,43E-04	4,37E-03	2,28E-04	55,56	1,00E-04	1,00E-04	0,00E+00	0,00E+00	0,00E+00	0,00E+00
0,00E+00		2	2,14E-04	6,78E-03	1,43E-04	4,37E-03	2,28E-04	55,56	1,00E-04	1,00E-04	0,00E+00	0,00E+00	0,00E+00	0,00E+00
0,00E+00		3	2,14E-04	6,78E-03	1,43E-04	4,37E-03	2,28E-04	55,56	1,00E-04	1,00E-04	0,00E+00	0,00E+00	0,00E+00	0,00E+00
0,00E+00		4	2,14E-04	6,78E-03	1,43E-04	4,37E-03	2,28E-04	55,56	1,00E-04	1,00E-04	0,00E+00	0,00E+00	0,00E+00	0,00E+00
0,00E+00		5	2,14E-04	6,78E-03	1,43E-04	4,37E-03	2,28E-04	55,56	1,00E-04	1,00E-04	0,00E+00	0,00E+00	0,00E+00	0,00E+00
0,00E+00		6	2,14E-04	6,78E-03	1,43E-04	4,37E-03	2,28E-04	55,56	1,00E-04	1,00E-04	0,00E+00	0,00E+00	0,00E+00	0,00E+00
0,00E+00		7	2,14E-04	6,78E-03	1,43E-04	4,37E-03	2,28E-04	55,56	1,00E-04	1,00E-04	0,00E+00	0,00E+00	0,00E+00	0,00E+00

[intermediate results deleted]

#Result Values														
0,00E+00		1	8,11E-04	6,20E-03	1,31E-04	1,13E-03	2,22E-04	55,55	1,00E-04	1,00E-04	0,00E+00	0,00E+00	0,00E+00	0,00E+00
0,00E+00		2	4,64E-04	6,53E-03	1,44E-04	1,15E-03	1,29E-04	55,55	1,00E-04	1,00E-04	0,00E+00	0,00E+00	0,00E+00	0,00E+00
0,00E+00		3	4,57E-04	6,55E-03	1,36E-04	1,16E-03	2,10E-04	55,55	1,00E-04	1,00E-04	0,00E+00	0,00E+00	0,00E+00	0,00E+00
0,00E+00		4	4,52E-04	6,56E-03	1,32E-04	1,17E-03	2,22E-04	55,55	1,00E-04	1,00E-04	0,00E+00	0,00E+00	0,00E+00	0,00E+00
0,00E+00		5	4,49E-04	6,56E-03	1,30E-04	1,17E-03	2,24E-04	55,55	1,00E-04	1,00E-04	0,00E+00	0,00E+00	0,00E+00	0,00E+00
0,00E+00		6	4,48E-04	6,56E-03	1,29E-04	1,18E-03	2,25E-04	55,55	1,00E-04	1,00E-04	0,00E+00	0,00E+00	0,00E+00	0,00E+00
0,00E+00		7	4,48E-04	6,56E-03	1,29E-04	1,18E-03	2,25E-04	55,55	1,00E-04	1,00E-04	0,00E+00	0,00E+00	0,00E+00	0,00E+00