WESTPOINT TREATMENT PLANT
OXYGEN PROCESS MODELING

by

Michael K. Stenstrom

Civil Engineering Department

University of California, Los Angeles

February 15, 1990

UCLA ENG 90-17

LIST OF FIGURES

Page

INTRODUCTION

The high purity oxygen activated sludge process is frequently used for large municipal applications where only a small area is available. This situation frequently occurs in mature cities where a large infrastructure has developed around existing treatment plants.

The high purity oxygen (HPO) activated sludge process is more complicated to understand and operate than a conventional process. The aeration basins are covered in order to maintain an enriched oxygen atmosphere. Under these circumstances the carbon dioxide which is normally stripped to the atmosphere is maintained in the gas headspace in quantities sometimes exceeding 0.1 ATM partial pressure. The nitrogen introduced in the high purity feed (~ 3% by mass) and the nitrogen dissolved in the liquid influent can exceed 0.5 ATM partial pressure in the vent gas. Since the driving force for oxygen transfer is strongly dependent upon the oxygen mole fraction, the total plant capacity is dependent on headspace oxygen mole fraction. The design procedures for estimating oxygen transfer are much more critical for this reason. Alternatively, the ability to manipulate headspace purity provides an extra degree of control to mitigate the impact of shock organic loads.

A number of models have been proposed for the high purity process. These models were developed in what can best be described as a continuing evolution of models from the earliest to the most current. The process was earlier described by McWhirter and Vahldieck (1970) who were associated with its development at Union Carbide. Mueller and coworkers (1973) developed a steady-state model, as well as Linden (1979). Dynamic models were developed by Cliff and Andrews (1986) and Stenstrom et al. (1989).

The aforementioned models were based upon Monod kinetics using the "defacto" standard approach developed during the late 50's and 60's by the environmental engineering research community, and documented by Lawrence and McCarty (1970). The model used in this approach represents a new development in the evolution of the HPO process models. The model combines the gas phase model developed by Stenstrom et al. (1989) with the structured activated sludge models developed over a number of years by Andrews and coworkers.

Structured models divide the biomass into separate fractions, such as active and inactive mass. The latest model developed in the series of structured Andrews models was proposed by Cliff and Andrews (1981), and recently described by Vitasovic and Andrews (1989). This model was interfaced to the HPO model developed by Stenstrom et al (1989), by replacing the unstructured model with the newer structured model.

MODEL DESCRIPTION

THEORY

Figure 1 is a block diagram of the structured model. The model considers that the influent biodegradable materials substrates are divided into two pools - particulate and soluble. The two substrates follow different pathways as they are degraded to carbon dioxide, water, or converted to cell biomass. The particulate substrate is first captured into a pool of material called stored particulate substrate. This process is analogous to biosorption whereby particulate and colloidal material are captured into the biofloc by a physical mechanism such as "sweep floc" coagulation. The stored substrate is next converted to stored biomass which can be oxidized to active biomass. This feature models the rapid removal of particulate substrate which can occur in the activated sludge process, and is extremely important for the various contact

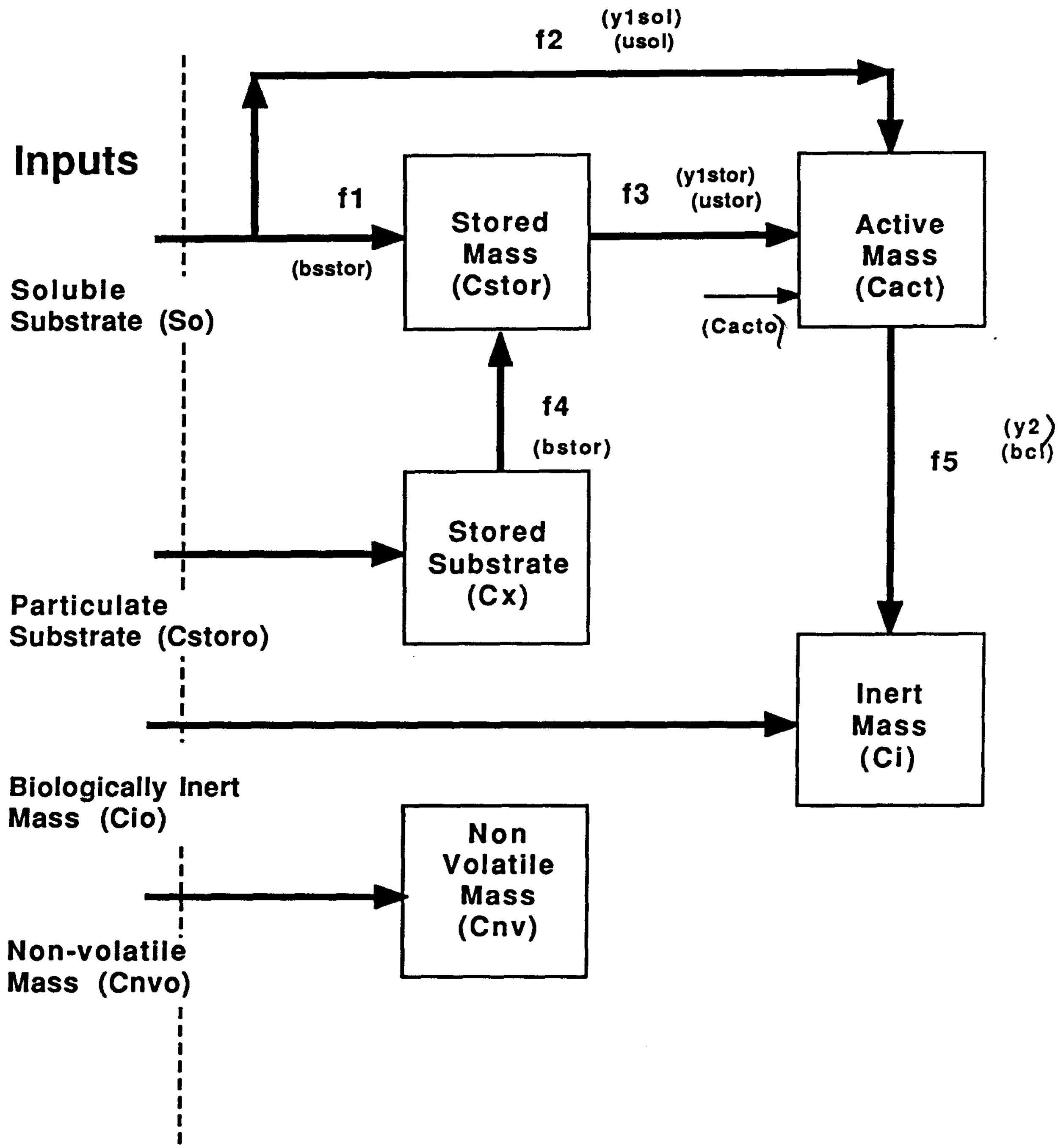

Figure 1. Model Structure

stabilization/aeration modifications. No oxygen is consumed in the conversion of influent particulate solids to stored mass.

Soluble substrate can be directly metabolized to form active mass, or it can be converted to stored mass. The two processes are competitive and the conversion depends upon the size of the stored mass pool.

Active mass decays in a fashion similar to the conventional unstructured model, but not all of the decayed material is converted to carbon dioxide and water. A fraction is converted to biologically inert material. This material is volatile but does not participate in any biochemical reactions. Non-biodegradable solids in the influent are added to this pool.

The two aspects of this model which are important to the West Point design studies are the stored mass and inert mass pools. These first feature is required to simulate contact reaeration or step feed modifications to the activated sludge process. The second feature is required to produce the high yield of volatile solids measured in the Seattle Metro pilot plant work (1989).

The kinetic equation used in the model are as follows, and correspond to the paths shown on Figure 1.

$$f_1 = \text{bsstor}\, C_{act}\, S\, (f_{cstrm} - f_{cstor}) \tag{1}$$

$$f_2 = \hat{\mu}_{sol}\, C_{act}\, S\, f_{O2} \tag{2}$$

$$f_3 = \hat{\mu}_{stor}\, C_{act}\, f_{cact}\, f_{O2} \tag{3}$$

$$f_4 = \text{bstor}\, C_{act}\, \frac{f_x}{f_x + \text{kcstor}} \tag{4}$$

$$f_5 = bci\ C_{act} \cdot f_{O2} \tag{5}$$

where

bsstor	=	rate coefficient for soluble substrate conversion to stored mass
bstor	=	rate coefficient for particulate substrate conversion to stored mass
$\hat{\mu}_{sol}$	=	rate coefficient for soluble substrate conversion to active mass
$\hat{\mu}_{stor}$	=	rate coefficient for stored mass conversion to active mass
bci	=	rate coefficient for conversion of active mass to inert mass

The "f" functions are described as follows:

$$f_{O2} = \frac{DO}{DO + K_{SDO}} \tag{6}$$

$$f_{cstor} = \frac{C_{stor}}{C_{stor} + C_{act} + C_i} \tag{7}$$

$$f_{C_{act}} = \frac{f_{cstor}}{f_{cstor} + 1} \tag{8}$$

$$f_x = \frac{C_x}{C_x + C_{act}} \tag{9}$$

where

f_{cstrm}	=	maximum fraction of biomass than can be stored mass
K_{SDO}	=	half velocity saturation coefficient for dissolved oxygen

There are three yields associated with the model. Conversion steps described by f_1 and f_4 have yields of unity. Rates f_2 and f_3 are divided by Y_{1sol} and Y_{1stor} to obtain substrate disappearance or stored mass disappearance rates. Rate f_5 is multiplied by Y_2 to obtain the inert mass production rate.

The carbonaceous model is coupled to the gas phase model through material balance and gas transfer equations described by Stenstrom et al. (1989). A copy of this paper is included in Appendix 1. Carbon dioxide production is related to oxygen consumption, with 1 mole of carbon dioxide produced per mole of oxygen consumed. The equations described in Appendix 1 are equally applicable to this problem. Oxygen consumption is described by equation 10, as follows:

$$O2_{uptk} = f_2 \frac{1-Y_{1sol}}{Y_{1sol}} K_{2sol} + f_3 \frac{1-Y_{1stor}}{Y_{1stor}} K_{2str} + f_5 (1-Y_2) K_{oex} \tag{10}$$

where $K_{2sol,}$ K_{2str} and K_{oex} are stoichiometric with units of mass O2/mass carbon. If the mass concentrations are written in terms of oxygen equivalents, these three coefficients are equal to unity. Continuity terms (flow in, flow out, both liquid and gas) are added to the above rates to produce nonsteady-state material balances. For more than one reactor in series, the terms must be subscripted.

The nomenclature used here is identical to that used in the model code and input files, with the exception of subscripts and the Greek "μ" which is represented as "u". For example μ_{sol} is written as usol in the program and input files.

The secondary clarifier was modeled using a lumped parameter approach with a one-dimensional ideal geometry. The model is identical to that described by Stenstrom (1976). The feature affects process dynamics but has little impact on oxygen transfer requirements and rates.

CODE OVERVIEW

The previously described model and the gas phase model were combined and rewritten into a single FORTRAN 77 code. The code is compatible with Microsoft FORTRAN, versions 4.1 and 5.0. The code has been thoroughly checked and has been compiled using IBM's VS FORTRAN on UCLA's 3090 using the MVS operating system. The code also has been compiled using the f77 compiler with an IBM RT/125 running AOS. All machines produced virtually identical output. Small differences in the sixth significant figure are due to the way each machine stores floating point variables.

Figure 2 shows an overview of the program. The source code is included in Appendix 2. Appendix 3 contains two sets of input files: one set for the pilot plant and one set for the anticipated full scale facility.

The governing equations developed produce a large collection of differential equations. The current configuration, including DO controllers and gas purity controller results in 68 ordinary differential equations (ODEs). Each stage requires 14 ODEs. The clarifier requires 10 and two more are required for stage 4 purity and system pressure controllers.

The model is written in a modular fashion using functions and subroutines as much as possible to produce an efficient, readable code. The model uses a total of 21 subprograms and a total of 1600 lines of source code and comments. The code design allows it to be expanded to include as many equations as necessary. There are no design aspects that limit the size of the model. Only the size and speed of the computer limits the number of equations that can be included in the model.

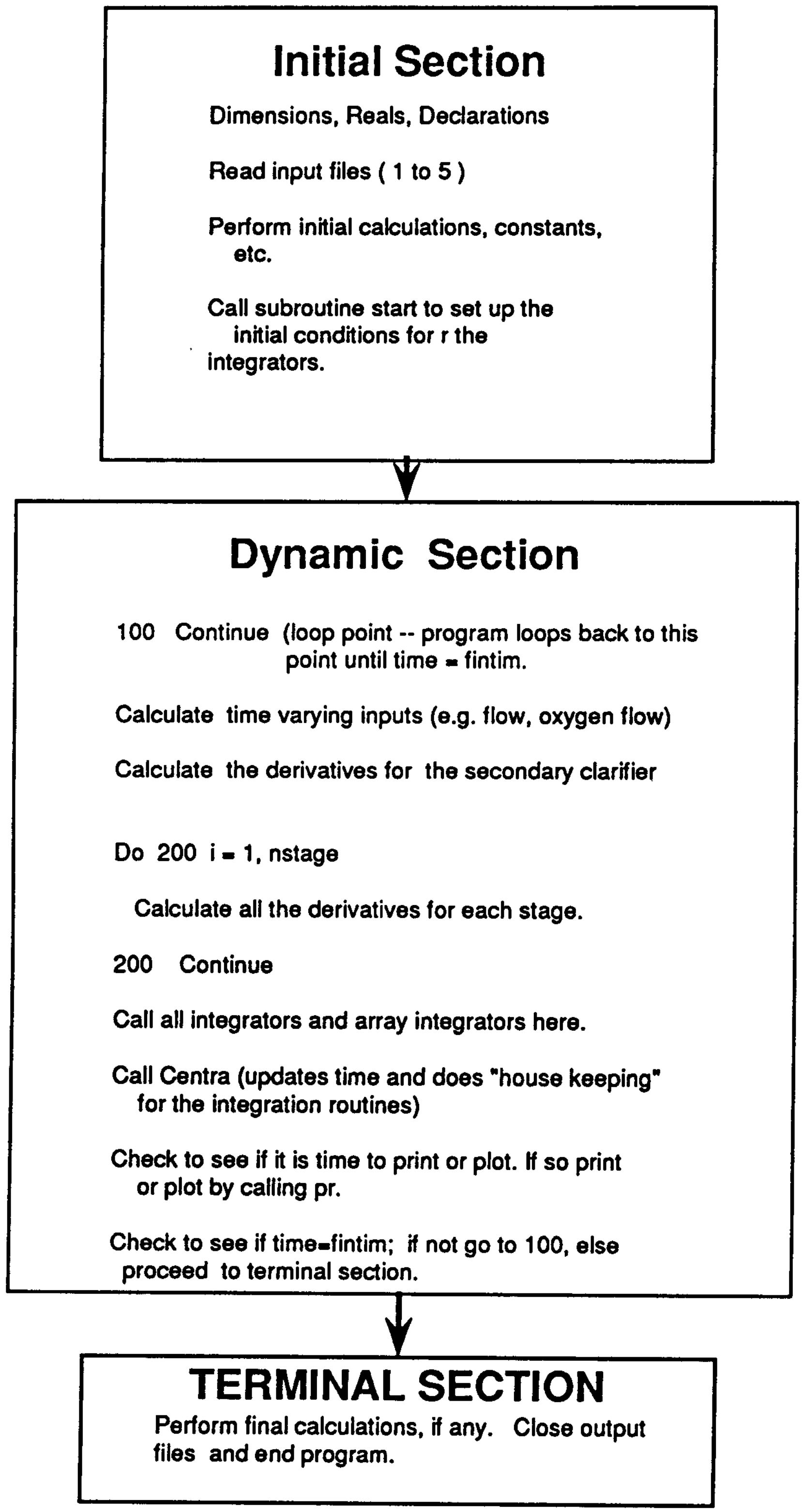

Figure 2. Model Block Diagram

Four integration methods are provided. Euler (or Rectangular), Modified Euler (or Trapezoidal), fixed step Runge Kutta (RKS), and variable step RKS. The variable step method may be the best choice for many problems and many save time. For problems with abrupt changes in the inputs, a fixed step procedure may be preferable. The procedures are common and are described in a number of textbooks, including James, et al (1985). The code was developed to resemble IBM's CSMP program; however, none of the code came from IBM.

The model uses five files for input. Four are required for all simulations. The fifth file is required only if arbitrary input functions are needed.

The TIMERS input file is read first. Table 1 shows a sample TIMERS file. The first line contains an integer which must appear in column 1. This integer defines the method and the numerical codes are shown in the table. The next line contains the integration step size. For the variable method, this value is taken as a starting point, and the step may increase or decrease in size. An internal variable DELMIN specifies the minimum step size, and this value has been arbitrarily set to DELT/10^4. It can be changed by editing the START.FOR subprogram. PRDEL and OUTDEL define the intervals for printing and plotting. The program provided in this contract does not plot but writes output to a file (OUTPUT.DAT) which is read by an AUTOCAD program written by Dr. Cello Vitasovic and staff. The ABSERR and RELERR terms like DELMIN, are used only for the variable step RKS procedure. Default values will be substituted if they are not included in the input file. This dataset is read on FORTRAN I/O unit 8. Format for this file and others is F10.0, which requires that input data be contained within the first 10 spaces and the data must included a decimal point (e.g. 10., not 10). If this program is compiled with a different compiler other than the ones used here, different rules may apply.

Table 1. Timers File

4	method (1=Euler, 2=Modified Euler, 3=RKS, 4=RKS variable)
0.0010	delt (integration time step)
0.1	prdel (print interval)
144.	outdel (plot interval)
144.	fintim (length of simulation)
0.001	abserr (absolute integration error, RKS variable only)
0.001	relerr (relative integration error, RKS variable only)

Notes: The above inputs are selected as follows:

1. Select the desired integration method: 1=Simple Euler, 1st order correct; 2=Modified Euler, 2nd order correct; 3=RKS fixed step, 4th order correct; 4=RKS variable step, 4th order correct. In general Method 4 should be used.

2. Delt is the integration interval in units of hours. Select a sufficiently small value to give stable results. For the examples here, 10^{-3} hrs is adequate.

3. Prdel and outdel are the intervals for printing and plotting. Select as desired. For diurnally varying flow, 1 hour is a good choice.

4. Fintim is the maximum time desired for simulation.

5. Abserr and relerr are the absolute and relative errors for the variable step integration. The values supplied here are adequate for the simulations shown. Decrease if the inputs are changed which result in instability. Increasing the values may result in faster execution, but will also result in less accurate solutions.

6. For a general review of integration procedures, see James et al (1985), Chapter 6.

Table 2 shows the PARAMS file. This file contains the parameters and coefficients, such as those described in equations 1 to 10. The model includes provisions for DO and stage 4 oxygen purity control. The limits for the controlled and the controller gains are included in this file as well. To restrict the values of K_La or the stage 4 oxygen purity (partial pressure), specify the appropriate limits in this file. If no control is required or wanted, set the gains to zero. The values of K_La specified in the file are initial estimates if the controllers are used. Otherwise the initial values will be used throughout the program. The method of providing changes in K_La is not considered by the program. The program assumes that K_La can be changed if the controller is active and the upper and lower limits are not equal. The format for this file is similar to the other files in that the first 10 spaces of each line are reserved for the variables, which must include a decimal point. The remaining spaces can be used for comments.

The clarifier area and depth are specified in this file as well as the reactor volumes; both liquid and gas reactor volumes are required. The clarifier area is the total area of all clarifiers. The reactor volume is the volume of a single train, stage-by-stage. The PARAMS file is read on FORTRAN I/O unit 9.

The model works by dividing the inputs equally among the specified number of trains. Therefore, the volume of only one train is required. To obtain global mass balances, the values associated with each train are multiplied by the number of trains in service. This procedure assumes that all trains operate in identical fashion. The procedure for handling the clarifier area is different. The total area must be specified, and it is divided equally among trains. This procedure is required because clarifiers may not be assigned to specific trains, and could be shared among trains.

Table 2. Params File

1.0	alpha1 (ratio of process to clean water kla's for stage 1)
1.0	alpha2 (ratio of process to clean water kla's for stage 2)
1.0	alpha3 (ratio of process to clean water kla's for stage 3)
1.0	alpha4 (ratio of process to clean water kla's for stage 4)
0.012	bci (active mass decay coefficient)
0.99	beta (ratio of process to clean water DO saturations)
0.405	BODU to BOD5 ratio
0.015	bsstor(specific rate for conversion of sol sub to stored mass)
0.500	bstor (specific rate for conversion of part sub to stored mass)
0.60	fcstorm (maximum fraction that can be stored mass)
0.05	kcstor (stored substrate fraction, dimensionless)
1.	kla1 (kla for stage 1, 1/hour) ***************************
5.	kla2 (kla for stage 2, 1/hour)
5.	kla3 (kla for stage 3, 1/hour)
3.	kla4 (kla for stage 4, 1/hour)
0.5	Lower limit for kla1 **************************
0.5	Lower limit for kla2 (upper and lower units on kla in 1/hr)
0.5	Lower limit for kla3
0.5	Lower limit for kla4
10.	Upper limit for kla1 **************************
13.	Upper limit for kla2
13.	Upper limit for kla3
13.	Upper limit for kla4
6.0	DO set point for Stage 1 ****************************
6.0	DO set point for Stage 2
6.0	DO set point for Stage 3
6.0	DO set point for Stage 4
2.0	Proportional gain for DO control (set to zero for no control)
0.2	Reset (integral) gain for DO control (set to zero for no control)
0.40	O2 purity in stage 4 setpoint (mole fraction)
1.0	Proportional gain for stage 4 purity control (1.0)
0.5	Rest (integral) gain for stage 4 purity control (1.0)
1.42	koex (o2 uptake from endogenous respiration)
1.10	ko2sol (o2 uptake from soluble substrate synthesis)
1.10	ko2str (o2 uptake from stored substrate synthesis)
2.0	kso2 (do half saturation coefficient, mg/L)
0.006	usol (maximum growth rate on soluble substrate)
0.75	ustor (maximum growth rate on stored substrate)
1.2	y1co21 (co2 pro'd per unit soluble substrate metabolized)
1.2	y1co22 (co2 pro'd per unit particulate substrate metabolized)
0.4	y1sol (active mass yield from soluble substrate)
0.4	y1stor (active mass yield from stored substrate)
0.15	y2 (biologically inert mass yield from active mass decay)
0.1239	ynh31 (ammonia consumed by active mass)
238000.	Clarifier area (ft^2)
16.	Clarifier depth (ft)
10.	Number of layers in the clarifier
1.	SRT/FM definition (2 uses clarifier sludge mass, 1 ignores it)
12544.	VG1 (gas, ft^3)

12544.	VG2 (gas, ft^3)
12544.	VG3 (gas, ft^3)
12544.	VG4 (gas, ft^3)
78400.	VL1 (liquid, ft^3)
78400.	VL2 (liquid, ft^3)
78400.	VL3 (liquid, ft^3)
78400.	VL4 (liquid, ft^3)

The INITS file is next read on FORTRAN I/O unit 10 and is shown in Table 3. This file specifies the initial conditions of all the state variables. The values are read across the page into an array. The format of the file is 5F10.0. Each entry must have a decimal point. The initial concentrations of each state variable (e.g. substrate, active mass, etc.) must be specified. At the conclusion of a normal run of the model, the final values of the state variables are written into a file called NINITS. This file may be renamed INITS in order to restart the model at the conclusion of a run. The restart using this procedure will be nearly equal to the previous simulations, with the exception of inputs that may have been changed by controllers, such as the K_La's or the oxygen feed rate.

The INPUTS file is read next on FORTRAN I/O unit 11 and is shown in Table 4. This file contains the input flows and concentrations. Input variables are read from the first 10 columns, as before, with a decimal point. The leak parameter specifies the gas loss in SCFM per inch of water pressure in the gas headspace. This feature has not been tested thoroughly for this code because no data were available. The step feed pattern is also specified in this file. The lines which have "Percent flow to Stage 1," etc. specify what percent of the influent flow is provided to each stage. The percents for all four stages should total to 100. The recycle flow is specified as a fraction (0 to 1.0) of the input flow and always enters stage 1. For cases where the flow rate is changing, the recycle flow rate will change also (ratio recycle control).

The input type parameter needs further explanation. If the input type is specified as 1, the liquid inputs are constant. If the input type is specified as 2, the liquid inputs are assumed to vary in a sinusoidal fashion with a period of 1 day. The magnitude of the variation is specified on the following lines. For example, if the flow rate Q were specified as 200 MGD with input=2 and a percent variation of 20, the flow would vary sinusoidally between a 160 MGD minimum to

Table 3. Inits File (initial conditions)

.0	.4	30.1	12.3	5.0						Soluble Substrate (mg/L)
.0	18.8	46.6	68.6	76.3						Stored Mass (mg/L)
.0	743.6	259.8	266.7	271.4						Active Mass (mg/L)
.0	1164.9	421.9	422.2	422.3						Biologically Inert Mass (mg/L)
.0	296.0	107.4	107.4	107.4						Non-volatile Mass (mg/L)
.0	2.5	46.0	22.7	8.2						Stored Substrate (mg/L)
.0	45.0	46.3	45.4	44.8						Ammonia Concentration (mg/L)
.0	5.9	5.9	6.0	6.0						Dissolved Oxygen (mg/L)
.05	.07	.08	.08							Carbon Dioxide (mole fraction)
.90	.81	.76	.72							Oxygen Purity (mole fraction)
6.21	6.27	6.23	6.20							Basin pHs
565.6	567.6	563.2	565.5	560.0	561.1	556.7	557.8	554.4	2437.0	Clarifier Solids (mg/L)

Notes:

Columns above for lines beginning with "Soluble Substrate" to "Dissolved Oxygen" represent stage concentrations. The zero in column 1 represents an internal value which is always 0. Columns 2 through 5 represent Stages 1 to 4, respectively.

For lines "Carbon Dioxide" to "Basin pHs," columns 1 to 4 represent Stages 1 to 4.

The last line represents the MLSS concentrations in the clarifier layers. Column 1 corresponds to the top layer. Column 10 corresponds to the 10th layer. A variable number of layers are possible, but the graphics package requires 10 layers.

Table 4. Inputs File

43.	Soluble BOD_5 (influent, mg/L)
0.	Conc. Influent active mass (mg/L)
50.0	Conc. Influent biologically inert mass (mg/L)
49.2	Conc. Influent ammonia (mg/L)
13.0	Conc. Influent non-volatile solids (mg/L)
0.	Conc. Influent stored mass (mg/L)
53.0	Conc. Influent particulate BOD_5 (mg/L)
200.	Conc. Influent alkalinity (as $CaCO_3$, mg/L)
0.	DO (influent, mg/L)
0.00	Leak parameter
6.	Number of Basins (trains of four stages each)
6.8	pH (influent)
143.3	Flow rate Q (mgd)
0.	Percent flow to Stage 1 (contact/reaeration)
100.	Percent flow to Stage 2 (contact/reaeration)
0.	Percent flow to Stage 3 (contact/reaeration)
0.	Percent flow to Stage 4 (contact/reaeration)
0.50	Recycle Rate (fraction of input flow rate)
1.0	SRT (set point, days)
15.0	Temperature (°C)
124.7	Oxygen feed (tons/day)
.97	Oxygen Purity (mole fraction)
3.	Input type (1=constant, 2=sinusoidal, 3=actual Randall's data)
20.	Percent sinusoidal variation in flow input (input type = 2)
20.	Percent sinusoidal variation in Particulate BOD_5 input (input type=2)
20.	Percent sinusoidal variation in Soluble BOD_5 input (input type = 2)

a 240 MGD maximum with a period of 1 day. If the input is specified as 3, the file DIURNAL is read, which allows an arbitrary input to be specified.

The file DIURNAL is read on FORTRAN I/O number 13 and is shown in Table 5. The first line of the file contains a two digit integer in columns 1 and 2 which specifies the number of flow data pairs that follow. The flow data pairs consist of a time of day (0. = midnight, 24. = midnight) and a normalized value of flow. The average of the normalized values of flow should always have a mean of 1.0. The data must be entered in ascending order and the intervals need not be constant. Interesting input functions can be created (e.g. square waves) using this procedure. The particulate BOD and soluble BOD data pairs follow in the file in a similar fashion. The time intervals need not match the flow time intervals. A minimum of two data pairs is required. A maximum of 20 pairs is allowed for each function, but this number can be increased, if desired, by changing dimensions and recompiling the main program and function AFGEN.FOR.

Two sets of input files are supplied on the floppy disks. The first set is in a directory called PILOT and are the results of the calibration shown later. The second set is in a directory called BIGPLANT and are for one of the simulations provided later. For the pilot plant data set, the controllers are turned off and for the bigplant data set the controllers are active with gains that provide reasonable control. The values of the gains for the proportional and integral (reset) functions can be determined by trial and error.

This completes the code description. It is supplied on a single 5.25" HD disk. The source for all routines (none of the AUTOCAD routines are supplied) is contained in the SOURCE directory. Routines SVS, STEP, START, FMIN, FMAX, PULSE, and LIMIT are supplied in a single file called COMBINE.FOR. The executable code is called MAIN.EXE and

Table 5. Diurnal Input File

13		number of data pairs for flow	
0.	1.1	time= 12 midnight	This file has format 2f10.0
2.	1.02	time= 2 AM	The time should appear in the
4.	0.90	time= 4 AM	first 10 columns with a decimal
6.	0.78	time= 6 AM	point. In the second 10 columns
8.	0.71	time= 8 AM	the normalized flow or BOD should
10.	0.83	time=10 AM	appear with a decimal point. For
12.	1.00	time=12 Noon	example, "2. 1.1" means that
14.	1.1	time= 2 PM	the flow or BOD was 110% of the mean
16.	1.12	time= 4 PM	at 2 AM. The number of points is
18.	1.13	time= 6 PM	specified by an integer that must
20.	1.15	time= 8 PM	appear on the first line of the file
22.	1.10	time=10 PM	and on the first line preceding the
24.	1.1	time=12 midnight	BOD data. This integer tells the
14		number of pairs	the program how many data pairs
0.	1.00		to read. The time can be entered in
1.5	1.06		any arbitrary spacing as long as it is
3.5	1.11		in ascending order. The time spacing
5.5	0.90		for flow and BOD do not have to match.
7.5	1.06		Blank lines are not permitted.
9.5	0.94		
11.5	1.06	Particulate BOD	
13.5	1.18		
15.5	1.02		
17.5	0.94		
19.5	0.92		
21.5	0.85		
23.5	0.94		
24.0	0.97		
14		Number of data pairs	
0.	1.20	Soluble BOD	
1.5	1.23		
3.5	1.58		
5.5	1.43		
7.5	1.14		
9.5	0.45		
11.5	0.39		
13.5	0.39		
15.5	0.94		
17.5	1.04		
19.5	1.17		
21.5	1.06		
23.5	1.17		
24.0	1.19		

is also contained in the SOURCE directory. The directories PILOT and BIGPLANT contain input files suitable to model the HPO pilot plant and the anticipated full scale plant. If further simulations are to be performed by either the Metro or CH2M Hill staffs, these files should be edited to include the new conditions and used as inputs.

The model requires approximately 20-30 minutes to perform a 144 hour simulation on a PS/2-70 (20 MHz) with 80387 math coprocessor installed. The time required is variable if the RKS variable step procedure is used.

MODEL CALIBRATION

The model was calibrated using several sources of data. First a set of plausible parameter values was formulated based upon the authors previous use of the models. The gas phase parameters and physical constants (e.g. Henry's Law constants) were taken directly from the author's previous publications (see Appendix 1).

Next the model was calibrated to match the HPO pilot plant results. Two documents provided by Metro were used. The first was appended to the March 16, 1989 meeting notes (Samstag, 1989). These notes contain a spreadsheet listing of the HPO data from the period of 6/26/88 to 7/15/88 when the pilot plant was operated in sludge reaeration mode. The data from this period were averaged (in some cases the spreadsheet contained the averages; in other cases it was necessary to re-average the data to obtain a sufficient number of significant figures). Table 6 shows the data and parameters which are either input parameters (e.g. liquid flow rate, 26.4 GPM) or observations to be fitted by the model (e.g. observed yield, 1.28 lb VSS/lb BOD_5 removed).

The Stensel report (1989) suggested an overall second-order reaction of primary effluent VSS with activated sludge as follows:

$$\text{rate} = -\,KPx \tag{11}$$

where x denotes activated sludge concentration and P denotes particulate BOD from primary effluent VSS. They showed that the value of K was approximately 0.001 L/mg-day for three different treatment plants.

Table 6. Model Calibration Information

Pilot Plant Data

Parameter	Value	Reference
Liquid Stage Volume	227 ft^3	1, p 3-3
Gas Stage Volume	40 ft^3	1, p 3-3
Reactor Gas Pressure	1.2" w.c.	2, p 15
Clarifier Area	50.2 ft^2	1, p 3-4
Clarifier Depth	8.7 ft^3	1, p 3-4
Net Yield (Y_n)	0.6-0.85 lb VSS/lb BOD_5	1, p 6-10
Observed Yield (Y)	1.28 lb VSS/lb BOD_5	1, p 6-9
O_2 Consumption	0.35 - 0.63 lb O_2/lb BOD_5	1, p 6-16
Average Influent Temp	19.5 °C	2, p 13
Average Influent pH	6.8	2, p 13
Average Flow	26.4 GPM	2, p 13
DO1	7.6 mg/L	2, p 14
DO2	5.2 mg/L	2, p 14
DO3	5.5 mg/L	2, p 14
DO4	5.0 mg/L	2, p 14
O_2 Uptake Rate	63 mg O_2/L-hr	2, p 14
O_2 Uptake Rate	96 mg O_2/L-hr	2, p 14
O_2 Uptake Rate	48 mg O_2/L-hr	2, p 14
O_2 Uptake Rate	41 mg O_2/L-hr	2, p 14
Recycle Rate	52%	2, p 14
O_2 Flow in	0.365 SCFM	2, p 15
O_2 Flow out	0.041 SCFM	2, p 15
O_2 Purity (feed)	97%	assumed
O_2 Purity 1	93.7%	2, p 14
O_2 Purity 2	82.8%	2, p 15
O_2 Purity 3	71.0%	2, p 15
O_2 Purity 4	65.6%	2, p 15
O_2 Utilization	92.5%	calculated
Effluent Avg pH	6.5	2, p 15
Waste Sludge	3035 GPD	2, p 15
Influent Total BOD_5	88 mg/L	2, p 17
Influent Soluble BOD_5	39 mg/L	2, p 17
Influent Total COD	217 mg/L	2, p 17
Influent Soluble COD	111 mg/L	2, p 17
Influent TSS	81 mg/L	2, p 17
Influent VSS	68 mg/L	2, p 17
MLSS	1346 mg/L	2, p 17
MLVSS	1171 mg/L	2, p 18
RAS	3577 mg/L	2, p 18
RAS (volatile)	3112 mg/L	2, p 18
SRT	0.3 - 2.0 days	2, p 3

Notes: Reference 1, p. 3-3 refers to the HPO pilot plant test report, p. 3-3; 2, p. 15 refers to the Samstag Appendix, p. 15.

The analogous reaction for this model is the reaction described by f_4 on Figure 1. This assumes that conversion of VSS to stored mass is rate limiting.

To assist in calibrating the model the Stensel data was used to identify the parameter "bstor". The Stensel data and rate relationship was equated to f_4, as follows:

$$f_4 = - KPx = \text{oxygen uptake rate} \tag{12}$$

For the purpose of this simple analysis it was assumed that 1 mg/L of VSS degraded equates to 1 mg/L of oxygen uptake. The results of this analysis for all data points suggest a value of bstor of 1.0 hr^{-1}. Two data points produced estimates of bstor of greater than 3.0 hr^{-1}. If these are removed the average is reduced to approximately 0.7 hr^{-1}. Therefore, the parameter was set to this value in the pilot plant simulations. Previous work had suggested that this parameter might be 0.5 hr, which agrees well with the Stensel results.

The only difference between this analysis and the Stensel analysis is the selection of the data. Only the first set of points were used herein, whereas Stensel used all the points after 1 hour. The value of bstor calculated using the first points produces model results which agree more closely with the pilot plant results.

The other model parameters were selectively adjusted in a trial and error fashion until a reasonable fit was obtained. Table 7 shows the fitted variables and the parameters. The model fit is very good with two exceptions: oxygen utilization and stage 1 uptake rate.

The first exception, oxygen utilization, corroborates the problems identified in the HPO pilot plant test report among the three oxygen consumption calculation procedures: gas phase mass balance, oxygen uptake rates, and COD mass balance. The report notes an approximate

Table 7. HPO Pilot Plant Simulation Results

Parameter	Measured Value	Calculated Value
MLVSS	1171 mg/L	1158 mg/L
MLSS	1346 mg/L	1331 mg/L
SRT	1.0 days	1.0 days
RAS (total)	3577 mg/L	3549 mg/L
RAS (volatile)	3112 mg/L	3055 mg/L
Soluble Substrate	5-10 mg/L	3.2 mg/L
% O2, Stage 1	93.7	89
% O2, Stage 2	82.8	79
% O2, Stage 3	71.0	73
% O2, Stage 4	66	69
Effluent pH	6.5	6.3
Observed Yield	1.28 lb VSS/lb BOD_{5r}	1.1 lb VSS/lb BOD_{5r}
Oxygen Consumption	0.35-0.63 lb O2/lb BOD_{5r}	0.66 lb O2/lb BOD_{5r}
O2 Uptake Rate, Stage 1	63 mg O2/L-hr	24 mg/L-hr
O2 Uptake Rate, Stage 2	96 mg O2/L-hr	89 mg/L-hr
O2 Uptake Rate, Stage 3	48 mg O2/L-hr	54 mg/L-hr
O2 Uptake Rate, Stage 4	41 mg O2/L-hr	40 mg/L-hr
O2 Utilization	92.5%	48%

agreement with the uptake rate and COD balance procedure, and a large difference with the gas phase mass balance.

The vent gas flow rate and stage 4 purity suggest that the oxygen utilization rate was greater than 90%. The author's previous experience with HPO plants suggests that the range of stage 4 gas purities required to obtain 90% oxygen utilization is approximately 40%. The full scale testing at the Sacramento plant showed ranges of 30 to 45% stage 4 purities when attempting to obtain 90% utilization. The HPO pilot plant averaged 65% purity, which if compared to Sacramento results would suggest approximately 50% gas utilization.

Gas purity measurement is an easy measurement to make, and can easily be performed without error. Measuring the vent gas flow rate is much more difficult. There are only 1.2 inches of water column pressure to force the vent gas through a metering device. A common error in the full scale testing program conducted by Union Carbide (based upon the author's observations and analysis at several plants) was to use a flow measuring instrument in the vent line which introduce a small but significant pressure drop. If the instrument is permanently installed in the line, the entire system pressure can increase, causing excessive leakage and structural problems. If the instrument is temporarily introduced into the vent line, an erroneously low flow reading is obtained. Very little vent gas will flow against the added head loss. At Sacramento a hot wire anemometer was used which produced very different, and much larger readings, than a propeller meter which introduced head loss.

If this experimental error existed during the HPO pilot program, it would produce the discrepancy among oxygen utilization measurement procedures. The model suggests a much lower utilization, a much larger flow rate and nearly the same exit purity. Another error which can cause the same disagreements in data is a gas leak in one or more of the pilot plant stages.

This information and analysis suggests that the COD balance and uptake rates are a more accurate measurement than the gas balance procedure. Therefore, the model was calibrated to match oxygen uptake rates, and in doing so it produces very plausible oxygen utilization rates. The purity profile (oxygen partial pressure across the four stages) also agrees very well using this procedure.

The second large difference between calibration data and calculations is the uptake rate in stage 1. The difference (-61%) is excessive; unfortunately no way has been found to provide a better fit. In the simulations developed later, it is suggested that the predicted $\alpha K_L a$ (and horsepower) be increased to compensate for this difference.

Table 8 shows the final calibrated values of the model parameters.

Table 8. Fitted Model Parameters

Parameter	Value	Description and Units*
bci	0.012	decay coefficient (hr^{-1})
BOD_5/BOD_n ratio	0.405	dimensionless
bsstor	0.015	transfer coefficient
bstor	0.50	transfer coefficient
f_{cstrm}	0.60	maximum fraction (m/m)
k_{cstor}	0.05	saturation coefficient (m/m)
K_{oex}	1.42	oxygen stoichiometric coefficient (m/m)
K_{O2sol}	1.10	oxygen stoichiometric (m/m)
K_{O2str}	1.10	oxygen stoichiometric (m/m)
K_{SO2}	2.0	oxygen saturation coefficient (mg/L)
μ_{sol}	0.006	maximum growth rate (hr^{-1})
μ_{stor}	0.75	maximum growth rate (hr^{-1})
Y_{1sol}	0.4	active mass yield (m/m)
Y_{1str}	0.4	active mass yield (m/m)
Y_2	0.15	invert mass yield (m/m)

* Mathematical definitions are provided by equations 1 through 10, on pages 4-6. A graphical interpretation is shown in Figure 1. All time units are in hours.

PLANT SIMULATION

The calibrated model was next used to simulate the full scale plant for several specific conditions. Figure 3 shows the simulation of the full scale plant for pilot plant conditions of operation and inputs (e.g. soluble BOD_5, oxygen feed, etc.). The top part of the figure shows the values of αK_La required to maintain a DO of 6.0 mg/L. The lower part of the figure shows the DO concentrations. Both are plotted as a function of time over a single day. This simulation was actually performed for 7 days in order to obtain periodic conditions, but only the last day is plotted.

The simulation shows the effectiveness of the DO controllers. The stage 4 oxygen purity is not shown, but it was also controlled at 50% ± 2% and provided a utilization of approximately 80%. Stage 1 shows excessive DO during the period of the day when the loading is low. This results because the controller was limited to a value of αK_La greater than that necessary to produce a DO of 6 mg/L. This condition is similar to a mixing limited situation. The DO and stage 4 purity controllers operate well for this simulation at low loading rate. At higher loading rates the controllers allow much greater excursions, particularly with respect to stage 4 purity.

Parameter estimates are not available for the α factors. Therefore, none have been made and the parameters graphed are αK_La's which represent the product of α factor and clean water K_La at the temperature of the simulation. This situation is created by setting the model α's to 1.0 and redefining the meaning of the K_La's printed by the model.

To calculate the horsepower required to produce adequate transfer, using the results provided herein, it is first necessary to estimate α factors. Next, it is necessary to convert form the simulation temperature to the standard temperature (20°C). Finally, a relationship between hor-

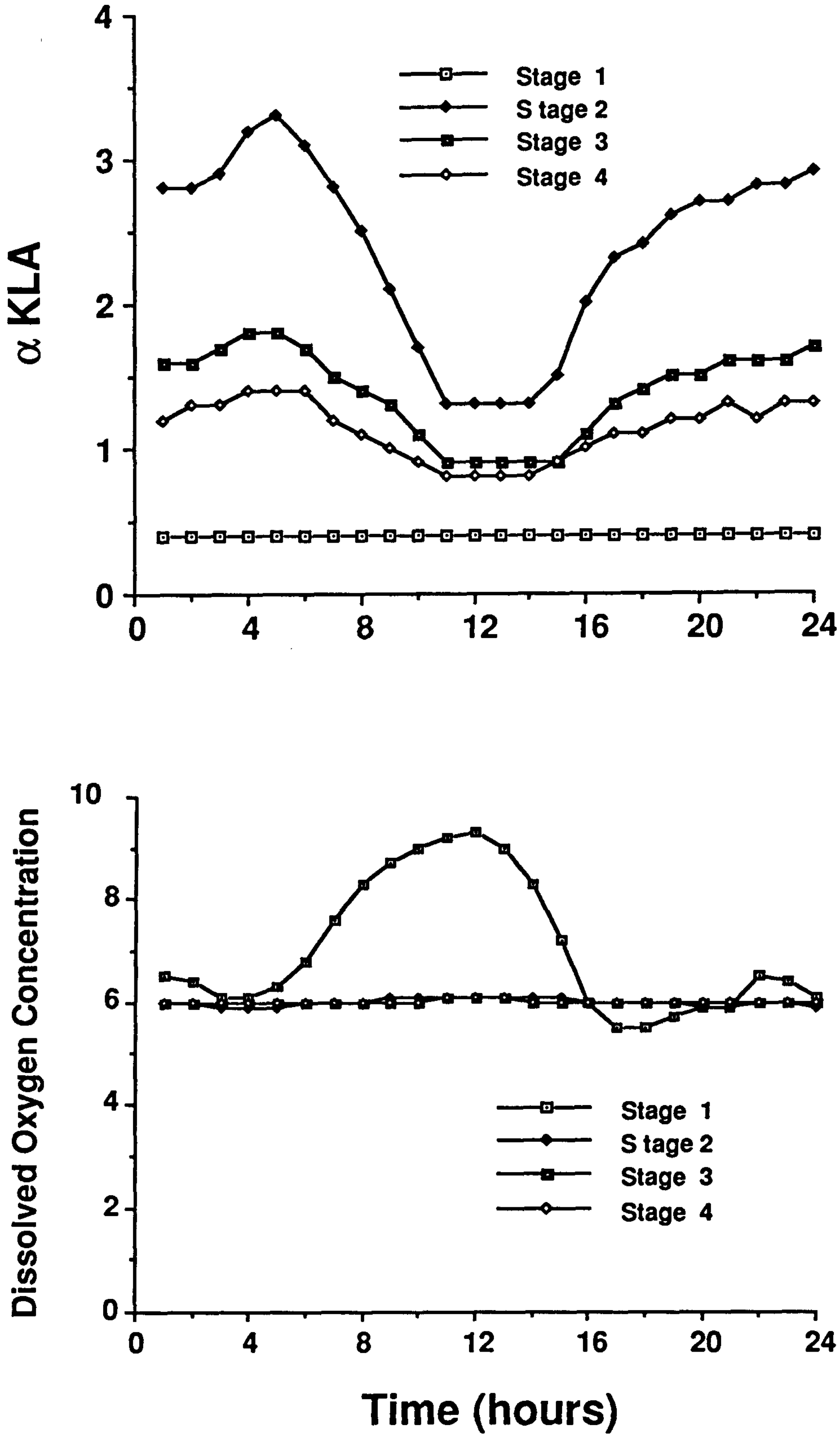

Figure 3. Typcial Simulation Results DO and αKla.

sepower and K_La is required.

Figure 3 shows that the peak K_La required to maintain 6.0 mg/L DO is approximately 3.2 hr^{-1}. If the α factor were 0.5, the clean water K_La at 15°C in 6.4 hr^{-1}. If the ASCE standard (1984) value of $\theta = 1.024$ is used the value of K_La_{20} is 7.20 hr^{-1}.

To calculate power several options are available. The techniques proposed by Butler (1989) is possible, which suggests that

$$K_La = 0.11\ (P)^{0.9} \tag{13}$$

or

$$P = (K_La/0.11)^{1.11} \tag{14}$$

For this case the power would be 113 shaft horsepower. This relationship was developed using correlations with aerator diameter and RPM. It is specific for the West Point design since the stage volumes are included.

The following tables report the maximum αK_La required to maintain a DO of 6.0 mg/L at an oxygen utilization associated with 40, 50, and 60% stage 4 oxygen purity. It was decided to control stage 4 oxygen purity as opposed to oxygen utilization. It is not possible to set both stage 4 purity and oxygen utilization. Controlling one fixes the other condition. For the parameters appropriate for the Westpoint plant, 40, 50, and 60% stage 4 oxygen purity correspond to 90, 80, and 70% oxygen utilization, respectively. The correspondence and the fact that the numbers are multiples of 10 is coincidental. At different temperatures or different parameters the correlations will be different.

Table 9 shows the maximum αK_La's for the projections made by Nicholson (1988) for the year 2005 (first page of the spreadsheet, following page 7 of the memorandum). The column TBOD to secondary treatment was used as the total load applied to the secondary system. An influent BOD concentration was calculated using the specified flow rate. The total BOD was divided between particulate and soluble according to the pilot plant findings (55% particulate, 45% soluble). All cases were simulated using the diurnal fluctuation in BOD and flow rate as indicated in the Samstag memorandum (1989). The plant was operated in the contact reaeration mode with 100% of the influent flow rate entering the second stage. It should be noted that these load projections are somewhat different than those evaluated by Lotepro and included as an Appendix in the Nicholson memorandum. The αK_La projections in Table 9 for 40, 50, and 60% stage 4 oxygen purity, which creates approximately 90, 80, and 70% % utilization, respectively.

The Nicholson memorandum assumes that constant O_2 utilization will occur for all loads. To achieve this at 6.0 mg/L DO, an αK_La of 11.0 hr^{-1} is required for stage 4 during peak demand. This compares to a required αK_La, for the same load of only 3.3 hr^{-1}, if 70% O_2 utilization occurs (60% stage 4 oxygen purity). The most economical combination of αK_La and gas utilization can be selected with the aid of the model.

Table 10 shows a second use for the model. This approach is the opposite of that shown in Table 9. In this series of simulations the projected stage horsepowers of 75, 125, 75, and 75 were used for stages 1 to 4, respectively. If a mechanical efficiency of 75% is assumed for the motor/gearbox combination, equation 13 can be used to calculate clean water K_La's. The simulations were made using PI control to keep the stage 4 oxygen purity at 40, 50, and 60%.

Table 9 Maximum and Minimum αK_La Values Required to Control Stage DOs at 6.0 mg/L at Various Oxygen Purities

40% Stage 4 Oxygen Purity

Condition	Stage 1		Stage 2		Stage 3		Stage 4	
	Max	Min	Max	Min	Max	Min	Max	Min
Average Annual Day	0.4	0.4	3.8	1.5	2.7	1.2	4.4	1.4
Maximum Month Average Day	0.7	0.4	4.4	1.6	3.1	1.2	5.5	1.5
Maximum Weak Average Day	0.7	0.4	4.0	1.5	3.3	0.9	5.2	1.0
Maximum Day	1.0	0.5	6.0	2.1	5.0	1.2	9.0	1.2
Peak Demand	1.2	0.6	7.0	2.4	5.9	1.3	11.0	1.5

50% Stage 4 Oxygen Purity

Condition	Stage 1		Stage 2		Stage 3		Stage 4	
	Max	Min	Max	Min	Max	Min	Max	Min
Average Annual Day	0.4	0.4	3.7	1.4	2.6	1.1	3.0	1.1
Maximum Month Average Day	0.8	0.4	4.0	1.4	2.8	1.1	3.6	1.2
Maximum Weak Average Day	0.9	0.4	3.9	1.4	2.8	0.8	3.3	0.9
Maximum Day	1.0	0.5	5.5	2.1	4.0	1.1	4.9	1.2
Peak Demand	1.2	0.6	6.7	2.3	4.7	1.4	5.4	1.4

60% Stage 4 Oxygen Purity

Condition	Stage 1		Stage 2		Stage 3		Stage 4	
	Max	Min	Max	Min	Max	Min	Max	Min
Average Annual Day	0.4	0.4	3.6	1.4	2.1	1.0	2.0	0.9
Maximum Month Average Day	0.7	0.4	3.8	1.4	2.3	1.0	2.3	1.1
Maximum Weak Average Day	0.7	0.4	3.4	1.4	2.1	0.8	2.1	0.7
Maximum Day	1.0	0.5	5.3	1.8	3.2	1.1	3.0	1.1
Peak Demand	1.1	0.6	6.5	2.4	3.7	1.3	3.3	1.2

Table 10 Maximum and Minimum DO Concentrations from Fixed Aerator Horsepowers

40% Stage 4 Oxygen Purity

Condition	Stage 1		Stage 2		Stage 3		Stage 4	
	Max	Min	Max	Min	Max	Min	Max	Min
Average Annual Day	29	23	18	6.2	17	5.5	14	3.3
Maximum Month Average Day	27	17	17	5.5	16	4.0	13.3	3.0
Maximum Weak Average Day	26	15	16	7.1	16	5.2	15.5	4.3
Maximum Day	23	11	13	3.2	13.3	2.2	12	1.6
Peak Demand	21	10	12	2.0	12.0	1.2	11	1.0

50% Stage 4 Oxygen Purity

Condition	Stage 1		Stage 2		Stage 3		Stage 4	
	Max	Min	Max	Min	Max	Min	Max	Min
Average Annual Day	29	24	18	7.5	18	7.0	17	7.0
Maximum Month Average Day	27	19	17	6.5	17	5.5	16	5.5
Maximum Weak Average Day	27	18	16	7.7	17	6.8	17	6.8
Maximum Day	24	14	13	3.5	14	2.5	14	2.5
Peak Demand	22	12	12	2.1	12	1.9	12	1.9

60% Stage 4 Oxygen Purity

Condition	Stage 1		Stage 2		Stage 3		Stage 4	
	Max	Min	Max	Min	Max	Min	Max	Min
Average Annual Day	31	26	19	8.0	19	8.0	19	9.0
Maximum Month Average Day	28	21	18	7.0	18	6.5	18	7.5
Maximum Weak Average Day	28	20	17	8.0	19	8.0	20	9.0
Maximum Day	25	15	14	3.8	15	3.0	16	4.0
Peak Demand	25	15	14	3.8	15	3.0	16	4.0

Additional simulations could be made using other operating conditions, such as no oxygen feed control.

CONCLUSIONS

This report has described a nonsteady-state mathematical model which can be used to estimate oxygen transfer capacity and/or requirements of a high purity oxygen activated sludge process. The effects of design variables such as reactor size, headspace volume, step feed pattern, and oxygen utilization can be explored. The model can also be used to estimate a number of other parameters, such as effluent BOD, impact of shock loads and other phenomena which may impact the plant.

The model fit the pilot plant data very well with the exception of oxygen utilization rate and uptake rate in stage 1. The oxygen utilization rate predicted by the model closely matches the oxygen uptake rate calculated in the pilot study using the mixed-liquor uptake rates or the COD balance. The model results suggest that there was a vent gas flow measuring error or gas leak from stage 4 in the pilot study.

The stage 1 uptake rate could not be calibrated to obtain a better fit. It is suggested that the predicted model's predicted K_La for stage 1 be increased by 60%. Alternatively, if the stage 1 aerator is sized for conventional operation, it will always be adequate for the contact reaeration mode.

The simulations herein were performed for a stage 4 purities of 40, 50, and 60%, which corresponds to 90, 80, and 70% oxygen utilization. For higher utilization, e.g. 90%, significantly higher αK_La's are required. For the extreme conditions of maximum day and peak demand, it may be more advisable to operate the plant at lower oxygen utilization such as 50 or 60%. This will significantly decrease the required αK_La while increasing the oxygen feed requirements. The trade-off between increased oxygen feed and increased αK_La is an economic and business decision, as well as a technical decision.

The model was used in two fashions. The first fashion used DO controllers which manipulated the $\alpha K_L a$ of a hypothetical aerator. This allows the model to calculate the required $\alpha K_L a$ to maintain a set point DO (6.0 mg/L for the cases reported herein) at a specified oxygen purity. The other method did not use DO controllers, which fixes the value of $\alpha K_L a$. In this case the model calculates the DO which results from a specific value of $\alpha K_L a$.

The model does not consider minimum required horsepower for mixing. This aspect of the design must be checked manually.

REFERENCES

1. "A Standard for the Measurement of Oxygen Transfer in Clean Water," American Society of Civil Engrs., New York, NY (1984).

2. Butler, Richard, letter to M.K. Stenstrom, July 13, 1989.

3. Clifft, R.C. and Andrews, J.F., "Predicting the Dynamics of Oxygen Utilization in the Activated Sludge Process," *J. Water Pollution Control Federation,* 53, 1219 (1981).

4. Clifft, R.C. and Andrews, J.F., "Gas-Liquid Interactions in Oxygen Activated Sludge," *J. Environ. Eng. Div. Am. Soc. Civ. Eng.,* 112, 61 (1986).

5. James, M.L., Smith, G.M. and Wolford, J.C., *Applied Numerical Methods for Digital Computation,* Third Edition, Harper and Row, San Francisco, CA (1985).

6. Lawrence, A.W. and McCarty, P.L., "Unified Basis for Biological Treatment Design and Operation," West Point Treatment Plant Secondary Treatment Facilities, High Purity Oxygen Design Test Facility Draft Report, Draft Final Report, Municipality of Metropolitan Seattle, Seattle, Washington, January 1989.

7. Linden, R.K.S., "Model for Minimizing Energy Requirements in the Pure Oxygen Activated Sludge Process," Ph.D. Dissertation, University of California, Davis (1979).

8. Mueller, J.A., et al., "Gas Transfer Kinetics of Pure Oxygen Systems," *J. Environ. Eng. Div. Am. Soc. Civ. Eng.,* 99, 264 (1973).

9. Nicholson, G., "HPO Aeration Tanks -- Design Documentation," letter and attachments, December 30, 1988.

10. Samstag, R., Minutes and Attachment to the April 16, 1989 Project meeting, April 7, 1989.

11. Stensel, H.D., "Activated Sludge Biodegradation Rates of Primary Effluent Suspended Solids," A final report to Seattle Metro, Dept. of Civil Engr., University of Washington, Seattle, Washington, November 27, 1989.

12. Stenstrom, M.K., "A Dynamic Model and Computer Compatible Control Strategies for Wastewater Treatment Plants," Ph.D. Dissertation, Clemson University, Clemson, SC (1976).

13. Stenstrom, M.K., et al., "Estimating Oxygen Transfer Capacity of a Full Scale Pure Oxygen Activated Sludge Plant," *J. Water Pollution Control Federation,* 61, No. 2, 208 (1989).

14. Vitasovic, Z. and Andrews, J.F., "Control Systems for the Activated Sludge Process, Parts I & II," Two manuscripts submitted to the *Water Pollution Research Journal of Canada,* April 1989.

APPENDIX 1 - GAS PHASE MODEL DOCUMENTATION

Estimating oxygen transfer capacity of a full-scale pure oxygen activated sludge plant

M. K. Stenstrom, W. Kido, R. F. Shanks, M. Mulkerin

ABSTRACT: A process-water, oxygen transfer compliance test was performed in November, 1983 on a 6.0-m^3/s (138-mgd) high purity oxygen activated sludge plant. The plant failed this and a subsequent process water test and the failure required the development of a procedure to determine oxygen transfer capacity of the plant. The American Society of Civil Engineer's clean water oxygen transfer standard was used in conjunction with process modeling and pilot-scale alpha factor testing. Clean water test results and a dynamic process model which predicts head-space gas purity are presented. *J. Water Pollut. Control Fed.*, **61**, 208 (1989).

KEYWORDS: oxygen transfer, aeration, oxygen, activated sludge, wastewater, modeling, simulation.

In 1973, the Sacramento Regional County Sanitation District was formed to provide wastewater collection and treatment for 17 separate agencies and industries that operated 21 separate treatment plants. To reduce costs and improve efficiency, a new regional plant was required. Planning and design for this new plant began in 1973, with construction beginning in 1976. The regional plant now treats all major sources of wastewater generated within Sacramento County.

The design engineers selected the high-purity oxygen activated sludge process for the regional plant. The selection was based on the successful process performance of a pilot plant study conducted at the city of Sacramento's main treatment plant during the last 6 months of 1973. Other reasons for selecting the process were the concerns for combined municipal and industrial wastewater treatment and odor control. Because the regional plant would have to treat a significant amount of seasonal food processing (canning) wastewater, it was believed that a high-purity oxygen system would perform better.

Plant startup began in November, 1982, and the first oxygen dissolution system compliance test was performed in November, 1983. This test was performed to verify that the specified amount of oxygen could be transferred at or below the specified power consumption rate. The compliance test was performed on process water during plant operation, as opposed to "clean water" in the conventional way. When the treatment process was designed and the specifications written, the American Society of Civil Engineers (ASCE) standard[1] for clean water testing did not exist.

The first process water test failed to demonstrate the specified performance. Two years later a second process water test showed the same result. For 2.5 years, the process, with respect to its mass transfer capability was investigated. The county and its consultants developed clean water data and a model to verify the process's oxygen transfer capability and shortcomings. The results of the model were eventually accepted by all parties. The plant model showed that the original oxygen transfer specifications, with the exception of an additional capacity requirement in the sludge reaeration mode (a form of step feed), could be met.

Plant Description

The Sacramento Regional Treatment Plant is a full secondary treatment facility providing treatment for 6.0 m^3/s (138 mgd), and includes raw and effluent pumping, primary clarification, secondary treatment with the high-purity oxygen activated sludge process, cryogenic oxygen production, disinfection, sludge thickening, and anaerobic digestion of waste sludges. The nominal design basis is shown in Table 1. The nominal design flow rate is 5.0 m^3/s (115 mgd) for dry weather, non-canning season, 6.0 m^3/s (138 mgd) for dry weather, canning season, and 10.5 m^3/s (240 mgd) for peak, wet weather conditions. The original oxygen transfer performance specifications were written using the 6.0-m^3/s flow rate, which will be used as the design flow rate throughout this manuscript, unless otherwise noted. Table 2 shows the secondary influent wastewater characteristics.

The plant has several unusual requirements. The effluent must be diverted to storage basins during periods when the Sacramento River velocity is less than 0.15 m/s (0.5 ft/sec). The stored effluent cannot be discharged directly to the Sacramento River, and must be returned to the plant influent.

An interesting aspect of this plant is its deep tanks and turbine aerators. The majority of high-purity oxygen plants use low-speed mechanical surface aerators. This plant uses turbine aerators that have a conical gas diffuser located 7.7 m (25 ft) below the liquid surface. High purity oxygen is normally released only in the first of four stages, but

Table 1—Nominal dry weather design basis for the Sacramento Regional Treatment Plant.

Process	Design basis
Primary clarification	Twelve primary clarifiers sized at 0.41 $m^3/m^2 \cdot d$ (1170 gal/sq ft/day)
Aeration basins	Eight trains of four basins (stages) in series, each measuring 14.6 m wide × 14.6 m long × 9.1 m deep (48 ft × 48 ft × 30 ft), providing a hydraulic retention time of 2.9 hours and an F:M ratio of 0.47
Oxygen production capacity	Two cryogenic oxygen plants, each producing 91 tonnes · d (100 tons/day) of 97% pure oxygen
Oxygen transfer capacity	Each train is equipped with four turbine aerators at 56 kW (75 hp), 45 kW (60 hp), 30 kW (40 hp), and 30 kW, with eight recirculation blowers totaling 1025 kW (1375 hp)
Final clarification	Sixteen 40-m (130-ft) circular tanks providing 0.23 $m^3/m^2 \cdot d$ (650 gal/sq ft/day) overflow rate.

can also be released in the other stages. Recirculation blowers are located in a central blower building. They take suction on the gas space of each stage, recirculate gas to the turbine diffusers, and are manifolded so that different blowers can be used for different stages. This unique feature gives added flexibility so that a wide range of gas recirculation rates are achieved. Normally, the gas in each stage is not mixed with other stages. To achieve the maximum mass transfer rate, high-purity oxygen is fed directly to Stage 1 recirculation blowers. It was the manufacturer's intent that the high-purity oxygen feed provide the entire gas flow to the Stage 1 blowers and turbines. The plant is designed so that the conventional and sludge reaeration modes can be used. Figure 1 is a schematic of the aeration basins.

Performance Warranty

As indicated earlier, a process water performance warranty was provided instead of a clean water specification. This was done partially because the ASCE clean water specification did not exist at the time the plant was designed, and in part because the oxygen transfer capability of a high-purity oxygen plant is strongly influenced by oxygen gas purity in each stage, which is not addressed by the ASCE standard or by clean water testing methods that existed when the plant was designed. Furthermore, the designer wanted to warrant other parts of the process, particularly the cryo plants.

The following process warranty was provided for the conventional or normal process mode and sludge reaeration modes provided that it was operating at the specified operations conditions:

- In the conventional mode, transfer 125 tonnes/d (138 tons/day) of oxygen with not less than 63.6 tonnes/d (70 tons/day) occurring in Stage 1, given 139 tonnes/d (153 tons/day) high-purity (97%) oxygen feed rate (90% oxygen utilization rate). It was further stipulated that this transfer occur at 6-mg/L average mixed liquor suspended solids (MLSS) dissolved oxygen (DO) concentration at a mixed liquor temperature of 28°C.
- Maintain an average DO in all stages of 6.0 mg/L or more, with no stage having less than 4.0 mg/L.
- Consume no more than 1600 kW (2144 hp). This total includes power for the turbine mixers and recirculation blowers, but excludes power associated with the cryo plants.
- In the sludge reaeration mode, transfer 160 tonnes/d (176 tons/day) at 177 tonnes/d (195 tons/day) of 97% purity oxygen feed rate. The temperature and DO concentrations for these requirements equaled those for the normal mode, and no maximum power was specified.
- In the event of non-compliance, the manufacturer was required to modify the system to meet the specified transfer rates. If the power consumption of the original or modified system exceeded 1600 kW, a power penalty of $3600/kW was to be assessed.

To test these warranty conditions, a full-scale process water test was planned. Oxygen transfer was estimated over a 7-day period using a steady-state material balance across the aeration basins. The material balance procedure required that the inlet and exit gas and liquid flow rates; inlet 5-day biochemical oxygen demand (BOD_5), chemical oxygen demand (COD), soluble COD, total suspended solids (TSS), volatile suspended solids (VSS), alkalinity, stage DO, and alkalinity concentrations; stage oxygen and CO_2 gas purities; and return sludge flow rate and concentration be measured at periodic time intervals, ranging from daily composites to instantaneous measurements every 4 hours. Stage gas purities were measured by collecting a sample from the gas head space of each stage; DO was measured by inserting a DO probe on a long shaft into the mixed liquor of each stage. Power was measured every 4 hours.

The COD, BOD, TSS, pH, and alkalinity data were not used in the material balance calculations for oxygen uptake rate (OUR). They were collected to ensure that the process met specified treatment efficiency, and that the influent wastewater met the design specifications shown in Table 2. This was necessary because influent wastewater characteristics can affect the gas space purity profile and oxygen transfer rates.

Table 2—Primary effluent water quality (design basis).

Parameter	Operating value
BOD_5, total	175 mg/L
BOD_5, soluble	114 mg/L
COD	335 mg/L
TSS	77 mg/L
VSS	62 mg/L
Temperature	28°C
Alkalinity (as $CaCO_3$)	160 mg/L
pH	7.1
Alpha factor	0.8
Beta factor	0.95

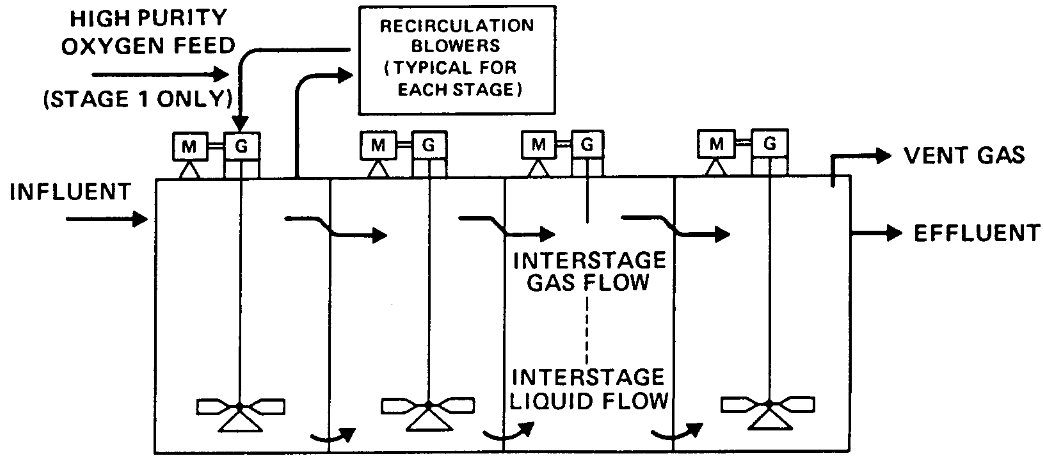

Figure 1—Plant schematic.

The data were reconciled by determining the oxygen transfer or OUR through direct measurement (input oxygen mass flowrate – the vent oxygen mass flowrate) and correcting this to the warranty conditions. The volumetric oxygen transfer coefficient, K_La, was estimated as follows:

$$K_La = \frac{OUR}{\alpha\theta^{T-20}[H\beta Y_oE_p - \text{DO}]} \qquad (1)$$

Where

K_La = volumetric mass transfer coefficient, tonnes/d;
OUR = oxygen uptake rate, tonnes/d;
α = alpha factor, dimensionless;
θ = theta factor, 1.024, dimensionless;
H = Henry's law constant, g/m^3·atm;
β = beta factor, dimensionless;
Y_o = oxygen gas pressure, atm;
E_p = effective pressure ratio, dimensionless;
DO = dissolved oxygen concentration, g/m^3; and
T = actual MLSS temperature, °C.

Alpha and beta factors were specified as 0.8 and 0.95, respectively, by the design engineer. The oxygen purity and DO concentration were measured in each stage of each train and averaged using a power-weighted ratio of each stage's mixer and blower power. These averages were calculated by multiplying each stage parameter by the total blower and turbine wire power. Products over all stages were summed and divided by the total power.

The effective pressure ratio was used to account for the hydrostatic pressure and was defined in the manufacturer's submittal as sparger mid-depth, which at 7.7 m depth is 1.37. Because the test was run for 7 consecutive days and DO and gas purity were measured at 4-hour intervals in each stage, it was necessary to evaluate Equation 1 forty-two times. The overall performance was evaluated as follows:

$$OUR = \alpha\overline{K_La}\theta^{28-20}[HY_o\beta E_p - \text{DO}] \qquad (2)$$

Where

$\overline{K_La}$ = 7-day average volumetric mass transfer coefficient.

The 7-day average OUR defined by Equation 2 was the warranty oxygen transfer rate. The warranty or specified values of α, β, E_p, DO, and Y_o were used in Equation 2. Unfortunately, no procedures were specified to determine α and β. Consequently, an error in the α or β values in Equation 1 will bias the estimate of OUR in Equation 2. This cannot be corrected. Also, there was no specified method to explain differences in the specified Y_o and the measured Y_o.

It was thought that insufficient wastewater may exist to test all eight trains at full capacity at plant startup. Therefore, a provision was made in the specification that allowed fewer than eight trains to be used for testing. For example, if there was insufficient wastewater to produce 125 tonnes/d of oxygen demand, fewer trains could be used with a linearly proportional decrease in the required oxygen transfer and allowable energy consumption.

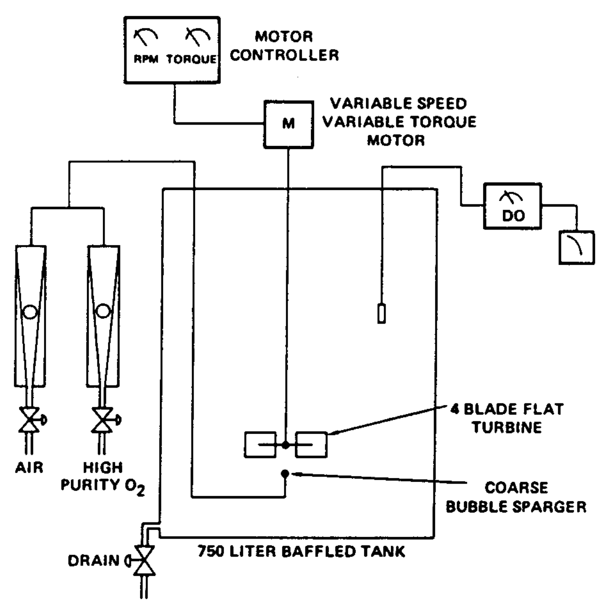

Figure 2—Alpha testing apparatus.

Table 3—Alpha factor test results.

Test series	Liquid	Date	Number of tests	K_La, min^{-1}	Alpha factor
1 S	Clean water	6/84	5	0.139 ± 0.005	
1 S	Stage 1	6/84	4	0.066 ± 0.019	0.48 ± 0.14
1 S	Stage 2	6/84	2	0.066 ± 0.008	0.48 ± 0.06
1 S	Stage 3	6/84	1	0.082	0.59
1 S	Stage 4	6/84	2	0.084 ± 0.002	0.61 ± 0.17
1 S	Primary effluent	6/84	2	0.054 ± 0.001	0.39 ± 0.008
1	Clean water	7/84	4	0.26 ± 0.007	
1	Stage 1	7/84	4	0.14 ± 0.007	0.55 ± 0.03
1	Stage 2	7/84	2	0.17 ± 0.001	0.67 ± 0.003
1	Stage 3	7/84	2	0.16 ± 0.001	0.62 ± 0.005
1	Stage 4	7/84	2	0.21 ± 0.012	0.79 ± 0.05
2	Clean water	7/84	3	0.17 ± 0.004	
2	Stage 1	7/84	2	0.0935 ± 0.003	0.54 ± 0.02
2	Stage 2	7/84	2	0.105 ± 0.001	0.60 ± 0.006
2	Stage 3	7/84	2	0.095 ± 0.001	0.55 ± 0.006
2	Stage 4	7/84	2	0.098 ± 0.002	0.56 ± 0.01

First Process Water Test

The first process water test was conducted November 2–9, 1983. Five trains were operated and an average of 42.5 tonnes/d were transferred using 889 kW. Directly scaling this transfer rate to eight trains gives 68 tonnes at 1422 kW. Oxygen utilization averaged 94.5%. The mass transfer rate was lower than expected and the manufacturer began to look for problems toward the end of the test period. This transfer was far short of the warranty conditions of 125 tonnes/d. Also, the DO in various stages did not meet the minimum measurement of 4.0 mg/L.

The manufacturer suspected that α was much less than 0.8. Consequently, a series of crude batch tests was performed in a 4-L vessel containing a fine pore stone diffuser. Primary effluent was used as the liquid for testing, as the design engineer's specification referenced an α value associated with primary effluent, as opposed to the mixed liquor. Alpha factors were then calculated by estimating K_La values from nonsteady-state reaeration of primary effluent spiked with mercuric chloride that terminated oxygen uptake. These K_La values were then divided by K_La values determined from reaeration of tap water that had been deoxygenated with nitrogen gas. A series of nine tests was performed. The average value for α was 0.35 with a minimum of 0.28 and a maximum of 0.42. Measurements for the determination of β were also taken; the average value was 0.95.

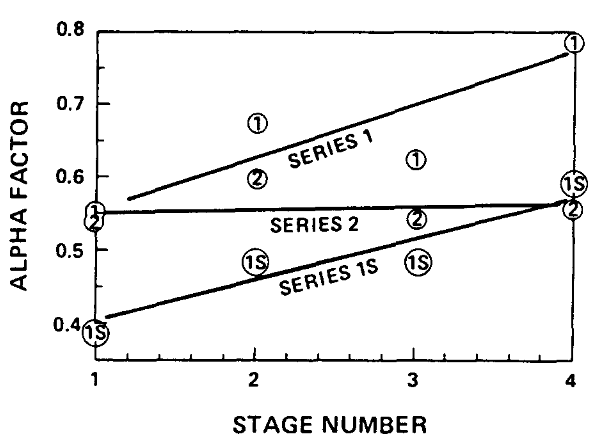

Figure 3—Alpha factor versus stage number.

The oxygen purity profile in the four stages during the process test was lower than anticipated. The manufacturer's analysis of the first process water test indicated an expected 66% power-weighted oxygen purity. The measured purity was only 52%.

When the values for α and β were incorporated into Equation 2, the warranty oxygen transfer increased to 115 tonnes/d for five trains in service, or 184 tonnes/d for eight trains, at 889 and 1422 kW, respectively. The manufacturer claimed that the wastewater and operating conditions during the test differed from those specified, and thus modified Equation 2 with an expected gas purity and α, changing the test conclusion from a 45% shortfall to a 47% excess in oxygen transfer capacity. The county and its consultants were unwilling to accept these calculation procedure modifications without documentation and verification.

Alpha factor testing. The first attempt to resolve the discrepancy in test result interpretation was to determine α. A test program was established in which primary effluent, mixed liquor, and a more appropriate apparatus were used. A realistic α for full-scale operation can only be determined using similar aeration devices. A fine pore stone in a 4-L bucket was convenient, but inappropriate, to determine α for the first process water test.

Figure 2 shows the apparatus used in this work. The 750-L aeration vessel was equipped with four baffles at 90-degree spacing. Each baffle was 10% of the tank diameter. The variable speed motor and gear box was selected to monitor rpm and torque, both of which change for different gas flow rates and water quality. A mixture of high-purity oxygen and compressed air was used. It was necessary to elevate the equilibrium DO concentration

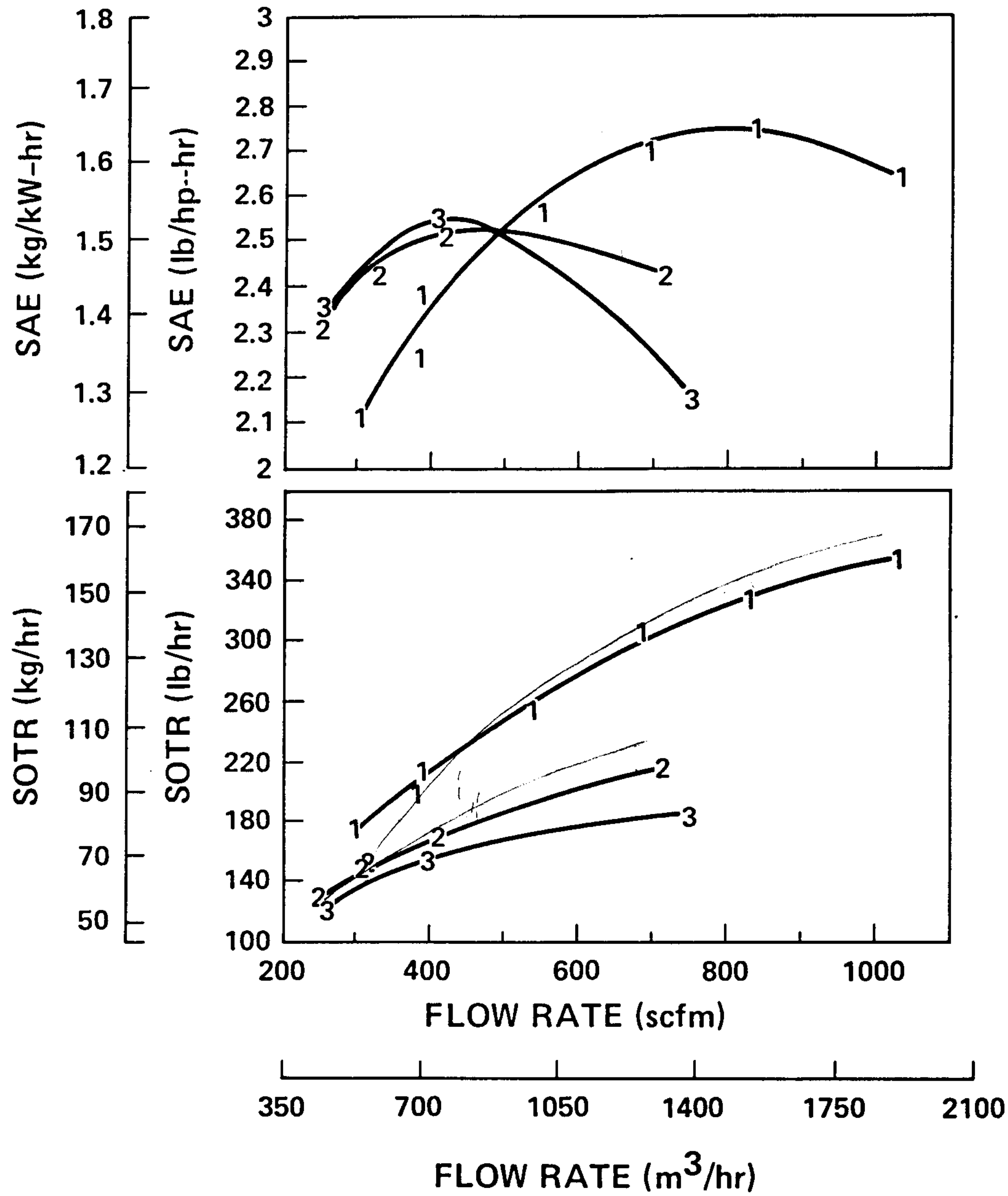

Figure 4—Clean water test results.

(C^*_∞), in the vessel in order to satisfy the oxygen uptake rate; otherwise, the DO in the test vessel would not change sufficiently to estimate K_La.

Researchers have recommended mercuric chloride or other chemicals that poison the mixed liquor to reduce its oxygen uptake rate to zero. This procedure was not used during this study, as there were concerns that the poisoning process might change the alpha factor. An alternate procedure that requires the oxygen uptake rate to be measured periodically during reaeration of a mixed-liquor sample was used.[3] Oxygen uptake rates were determined by collecting a sample from the 750-L tank, shaking if necessary to elevate DO concentration, taking a series of DO measurements, and then briefly recording the decline in DO concentration of the sample in a stirred BOD bottle.

In this way, a mathematical analysis procedure very similar to the ASCE nonsteady-state procedure can be used. To calculate K_La, the sum of squares was minimized as follows:

$$SS = \sum (DO_t - DO_t^o)^2 \qquad (3)$$

Where

DO_t^o = measured DO concentration at time t, and
DO_t = calculated DO concentration at time t.

The DO_t was calculated by integrating the following equation:

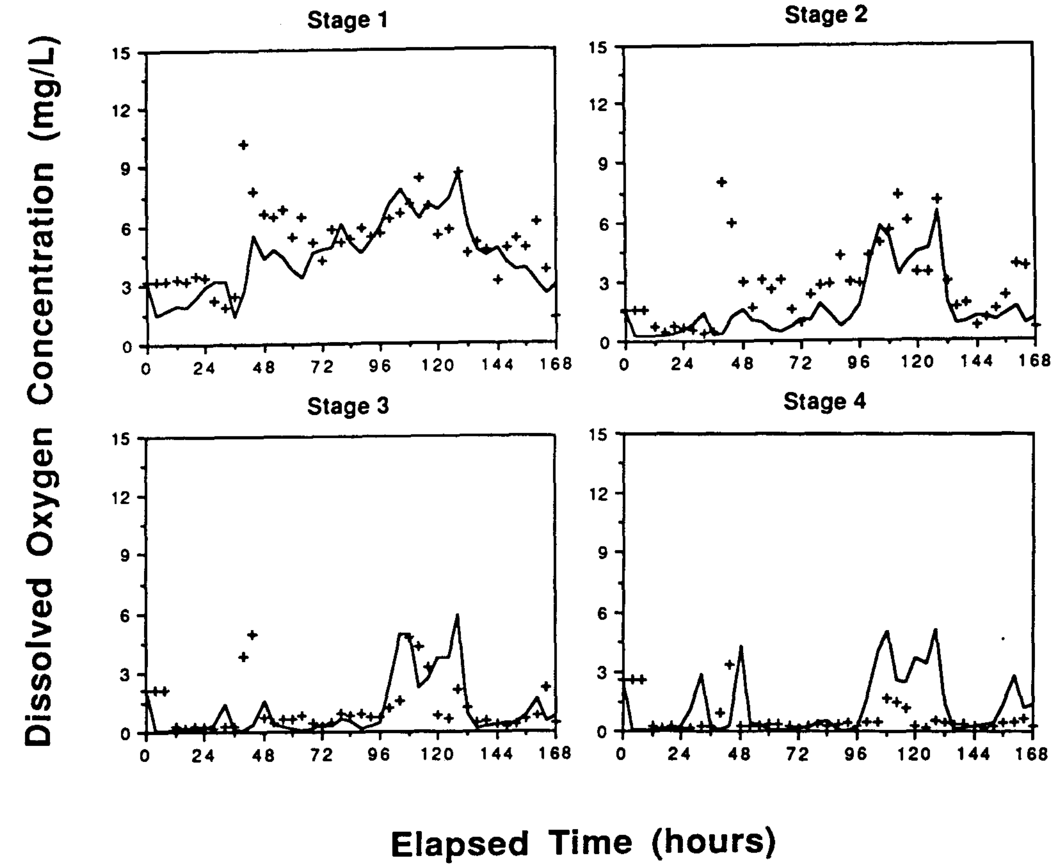

Figure 5—First process water test simulation: DO.

$$\frac{\mathrm{dDO}_t}{\mathrm{d}t} = K_L a(C^*_\infty - \mathrm{DO}_t) - r(t) \quad (4)$$

Where

C^*_∞ = equilibrium DO concentration, and
$r(t)$ = DO uptake rate as a function of t.

A procedure using the ASCE-supplied nonlinear least squares program to find the minimum in Equation 3 has been developed.[4] This technique requires that $r(t)$ in Equation 4 be adequately described by an exponential function. Another procedure, however, allows $r(t)$ to be an arbitrary function of time.[3] For this case, a second-order Lagrangian interpolation of the measured data points was used to model $r(t)$.

The apparatus in Figure 2 was placed on wheels and moved from stage to stage in order to determine α for each stage. A submersible pump, placed in each stage's sample port, was used to fill the tank. Filling time was kept minimal to keep the mixed liquor as fresh as possible, as mixed liquor in endogenous respiration generally has an elevated α.[2]

Table 3 presents results from the series of performed tests (PE = primary effluent and CW = clean water). Series 1S was performed with a 7.6-cm (3-in.) marine-type impeller at high rpm. In general, it was impossible to approximate the full-scale power density in the test tank using this impeller. A larger, 24-cm (9.5-in.) four-blade flat turbine was used later at lower rpm. This impeller, because it consumed more power, provided for conditions that were closer to those in full-scale tanks. These results are reported as Series 1.

Testing for Series 1S and 1 was performed between 9 a.m. and 3 p.m. This corresponds to the period of increasing plant load. Testing for Series 2 was performed in the early morning hours, between 5 a.m. and 10 a.m., which corresponded to the period of lower loading.

The trend for α is shown in Figure 3. Generally, α increases in the later stages. Some of the variability may be explained by loading changes, as it was impossible to perform all tests under the same plant load. During the periods when tests were performed, the approximate plant F:M ratio was 1.03 for Series 1S and 0.90 for Series 1 and 2. The corresponding mean cell retention times, calculated using sludge inventory in the aeration basins and secondary clarifiers, were 2.9 and 2.8 days, respectively. The power-weighted average α values (using the expected power consumption for the full-scale system) for Series 1S, 1, and 2 were 0.52, 0.64, and 0.56, respectively. The average α associated with primary effluent and the small impeller was 0.39 ± 0.01. This compares with primary effluent alpha values of 0.35 ± 0.07 that were determined by the manufacturer in the process water test.

The precision of the tests among replicates was quite good, but the absolute magnitude of α was much less than

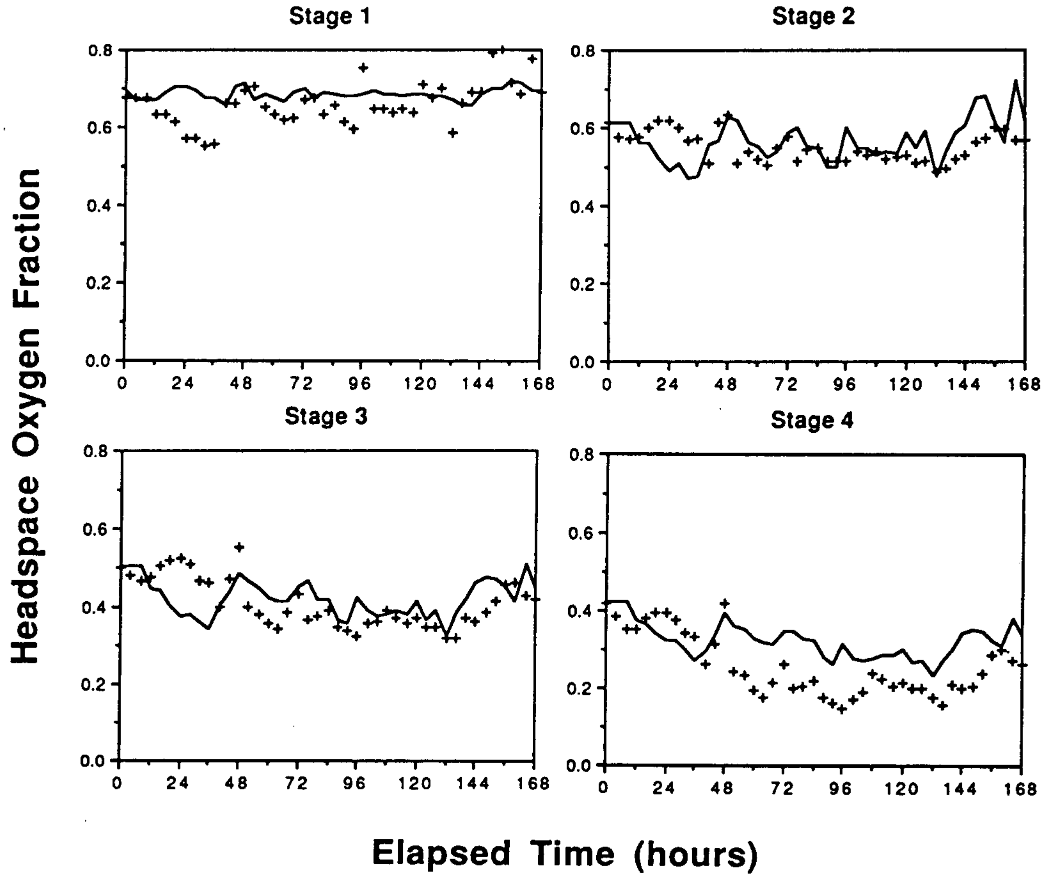

Figure 6—First process water test simulation: oxygen purity.

specified by the designer. The result did not resolve the dispute with the manufacturer, as a rather substantial power penalty and performance shortfall existed when the new α was factored in Equations 1 and 2, along with the measured gas purity of 52%. Furthermore, there was no way to show that the α-test apparatus accurately simulated the full-scale aeration system.

Clean water test. A hypothesis on the cause of the mass transfer rate deficiency was stipulated to be the specified clean water transfer efficiency of the turbines. The warranty specifications did not state the clean water transfer efficiency, but manufacturer's documents indicated a clean water transfer efficiency of 2.43 kg O_2/kW·h (4.0 lb O_2/hp/hr) in conventional mode, and 2.75 kg O_2/kW·h in sludge reaeration mode. The manufacturer used power units of brake and shaft horsepower, as opposed to wire horsepower. Clean water efficiency increased in the reaeration mode because gas recirculation rates increased and not because process conditions were different. Furthermore, an examination of documents supplied by the manufacturer revealed that the clean water transfer efficiency had been scaled up from 1.95 kg O_2/kW·h observed in their testing program in tanks 6.4 m deep to 2.43 kg O_2/kW·h in the county's 9.1-m deep tanks. Using the manufacturer's estimated blower and motor-gearbox efficiency, the standard oxygen transfer efficiency (or standard aeration efficiency, SAE), which is based upon wire horsepower, ranges from 1.83 to 2.05 kg O_2/kW·h (3 to 3.4 lb O_2/hp/hr) for the conventional and reaeration modes.

A clean water test was planned for Stages 1, 2, and 3. Stage 4 was not tested because it was identical to Stage 3. Train 8 was prepared for testing by draining and cleaning several times. This train had been previously used and was contaminated with mixed liquor. The openings between stages were blocked with plywood barriers. These barriers were designed to be opened and closed from the tank top using ropes and pulleys. During tank filling the barriers were opened to avoid damage from differences in hydrostatic pressure.

Train 8 was isolated from the high-purity oxygen system by closing the appropriate valves. Atmospheric air was supplied from a 150-hp positive displacement blower that was connected to each turbine through flexible hoses. To measure gas flow rate, a 12-m (40-ft) flow tube containing an orifice plate and a multiple-ported pitot tube was installed between the blower discharge and the turbines. Initially, there were severe problems in measuring flow rate accurately because the pitot tube's position along the flow tube influenced its flow indication. Also, the agreement with the orifice plate was poor. It seemed that there was some type of standing pressure wave in the air piping. After installing a noise silencer between the blower discharge and flow tube to function as a pulsation dampener, the problem was eliminated and the pitot tube measure agreed with the orifice plate measure to within ±1.5%.

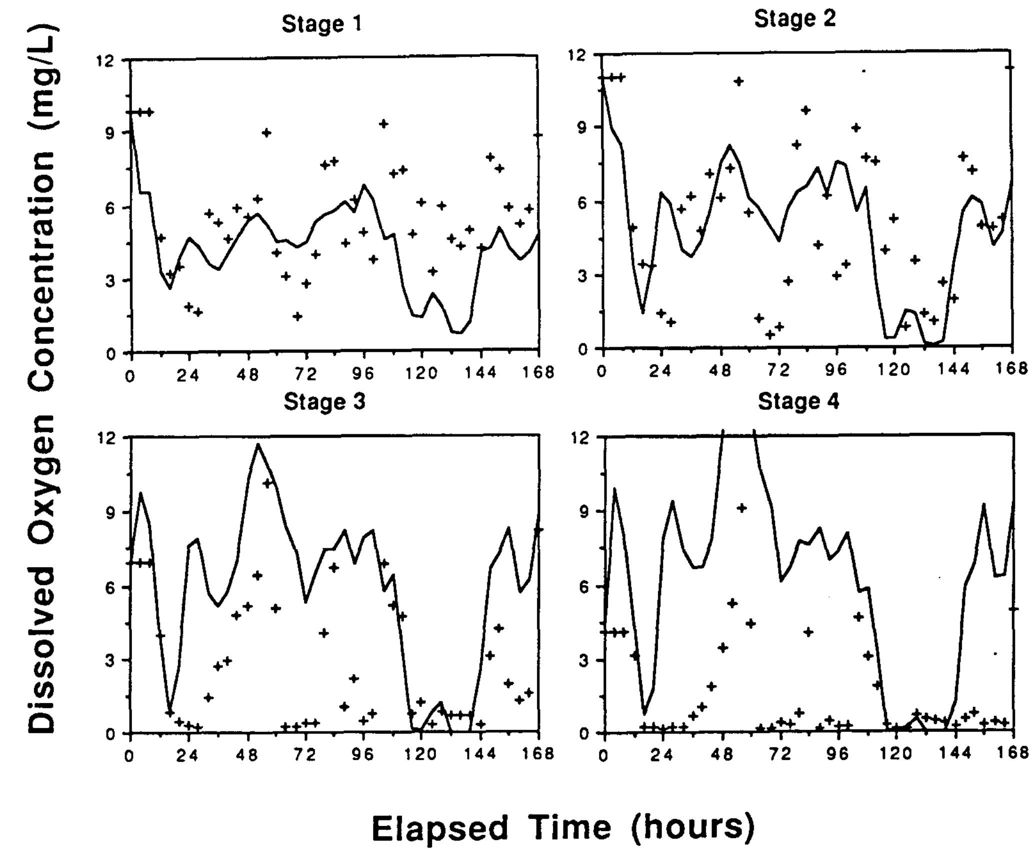

Figure 7—Second process water test simulation: DO.

There was concern that oxygen depletion from absorption to the basin's head space might occur, and that this depletion may influence transfer rates. To prevent oxygen depletion, manhole ventilators were used. The manholes access to the basins were opened and a fresh air cross-flow was established. During testing, head space oxygen purity never fell below 19%. In addition, the turbines were operated at depressed tank DO concentration without gas recirculation to determine the rate of aeration at the tank surface. No measurable change in DO was detected over a 30-minute period.

There was another concern that water quality might affect oxygen transfer performance. According to the manufacturer it was impossible to clean the tanks sufficiently; therefore, test water may not be representatively clean. To evaluate water quality during testing, du Nouy static surface tension measurements were made before, during, and after testing. The measured surface tension of tap water varied, throughout testing, less than ±1 dyne/cm. Alpha factors were also determined for basin clean water as described previously. Oxygen uptake rate was expected to be zero in the basin water and this was confirmed by direct measurement. The α factor associated with basin water was ±5% of unity, which was within the experimental error of the test procedure. It was concluded that the basin was adequately cleaned, and that test water was not contaminated.

The ASCE standard procedure was followed. Experiments were continued to $4/K_La$ units of time and data were analyzed with the nonlinear least-squares procedure. Figure 4 shows the results of the test program reported as standard oxygen transfer rate (SOTR), which is the mass of oxygen transferred per unit time at 20°C, 0 mg/L DO, $\beta = 1.0$, $\alpha = 1.0$, and at atmospheric pressure of 760 mm Hg, and as SAE, which equals SOTR/wire power input. The SAE numbers were calculated based on measured mixer power and the blower power required to produce an equivalent gas flow rate from the plant's recirculation blowers, which was measured previously as 30.9 $m^3/h \cdot kW$ (18.2 scfm/kW). The test blower power was not used because it differed in design and efficiency from the plant blowers.

Stage 3 was tested first. Three repetitions were performed at a gas flow rate of 442 m^3/h (260 scfm). The SOTR values for these three tests were 117.6, 117.8, and 118.3 kg/h, or less than 0.6% difference. The precision of the test was excellent. Based on this reproducibility, only two replicates in the other stages were performed, and the additional test was used to expand the range of gas flow rates. Six probe locations were used at different areas and depths. The probe-to-probe variability was also well within the limits of the standard.

Figure 4 shows optimum gas recirculation rates for each stage. None of the turbines met the specified transfer efficiency of 1.95 kg $O_2/kW \cdot h$ (3.2 lb O_2/hp/hr). Stage 1 was 10% short of optimum gas flow rate. Stages 2 and 3 were 17% short. For the reaeration mode, a higher SAE was anticipated by the manufacturer, Stage 1 was 20%

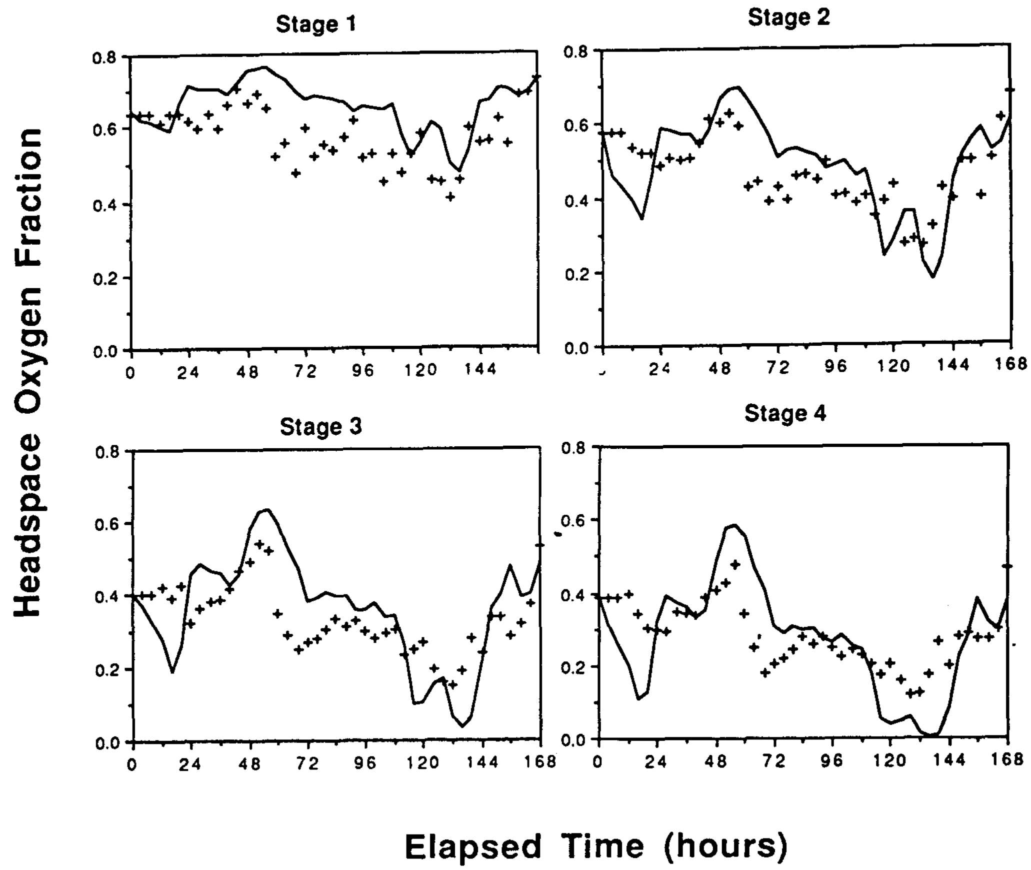

Figure 8—Second process water test simulation: oxygen purity.

short, and Stages 2 and 3 were 25% short. The effective pressure ratio ranged from 1.32 to 1.38, in close agreement with the specifications.

Because the manufacturer did not accept the clean water results as binding, a second process water test was conducted. With the information provided from the clear water testing, the optimum recirculation rates could be used in a second process water test.

Second Process Water Test

The second process water test was conducted in November, 1985. The procedures were very similar to the first test. Six trains were operated. Gas recirculation rates were 1245, 715, 715, and 470 m^3/h in Stages 1 to 4, respectively. This compared to the first process water test recirculation rates of 700, 460, 490, and 490 m^3/h. Alpha factors were determined during the test at 4-hour intervals using the 750-L apparatus. The average α factors for Stages 1 to 4 were 0.63, 0.61, 0.64, and 0.69, respectively. The average α associated with primary effluent was 0.39.

During the test, a large rainstorm occurred affecting plant operation. Also, high-purity oxygen feed rate was upset several times. The cause of those upsets was believed to have been unusual operating conditions needed for the process water test. At one point, the oxygen gas purity in the fourth stage was less than atmospheric purity, decreasing to 11% oxygen. As indicated previously, during low flow in the Sacramento River, the plant diverts effluent to holding basins, and must retreat and discharge the stored effluent during periods of increased river flow rates. During the second process water test it was necessary to divert and retreat effluent.

The manufacturer adjusted Equation 2 based on the specified α and the expected gas purity of 66% and claimed that the plant had passed the performance test. However, the specified transfer rates could not be demonstrated without relying on dubious assumptions regarding alpha factors, steady-state conditions, and gas purity. Steady-state conditions in the aeration basins were never obtained.

Process Modeling

After conducting two full-scale, 7-day process water tests with no definitive conclusions, a dynamic process model was developed to verify or disprove the design oxygen purity profile specified by the manufacturer. The process model was based in part on earlier steady-state models.[5-7] The model is similar in concept to another model.[8] The model, developed here, was written using CSMP III,[9] a simulation program designed to solve systems of linear or nonlinear ordinary differential equations. The model was developed independently of the manufacturer.

Model description. Equations 5 through 21 describe a single stage of a four-stage process. Balances must be written for species in both the liquid and gaseous phase. The model does not include activity coefficients, and is therefore restricted to low ionic strength wastewaters. For the

Table 4—Equations describing a single-stage of a four-stage process.

Species	Equations	
Gas phase		
	$\frac{dCO_2}{dt} = \frac{Q_{Go}CO_{20} - Q_G CO_2}{V_G} - K_L a_{DCD}(DCD_S - DCD_l)\frac{V_L}{V_G MW_{CO_2}}$	(5)
	$\frac{dN_2}{dt} = \frac{Q_{Go}N_{20} - Q_G N_2}{V_G} - K_L a_N(DN_S - DN)\frac{V_L}{V_G MW_{N_2}}$	(6)
	$\frac{dO_2}{dt} = \frac{Q_{Go}O_{20} - Q_G O_2}{V_G} - K_L a_{DO}(C^*_\alpha - DO)\frac{V_L}{V_G MW_{O_2}}$	(7)
	$Q_G = K_{Flow}(P_{SP} - P_T)$	
Partial pressures		
	$P_{CO_2} = CO_2 \cdot RT$	(8)
	$P_{O_2} = O_2 \cdot RT$	(9)
	$P_{N_2} = N_2 \cdot RT$	(10)
	$P_T = P_{CO_2} + P_{O_2} + P_{N_2} + P_{H_2O}$	(11)
Liquid phase		
	$\frac{dX}{dt} = \frac{Q_L}{V_L}(X_o - X) + [\mu - K_D]X$	(12)
	$\frac{dDO}{dt} = \frac{Q_L}{V_L}(DO_o - DO) + K_L a_o(C^*_\alpha - DO) - \mu X\left(\frac{1-Y}{Y}\right)Y_{O_{21}} - K_D X Y_{O_{22}}$	(13)
	$\frac{dS}{dt} = \frac{Q_L}{V_L}(S_o - S) - \frac{\mu}{Y}X$	(14)
	$\frac{dDN}{dt} = \frac{Q_L}{V_L}(DN_o - DN) + K_L a_{N_2}(DN_S - DN)$	(15)
	$\frac{dDCD}{dt} = \frac{Q_L}{V_L}(DCD_o - DCD) + K_L a_{CO_2}(DCD_S - DCD_l) + \mu X\left(\frac{1-Y}{Y}\right)Y_{CO_{21}} + K_D X Y_{CO_{22}}$	(16)
	$\mu = \frac{\hat{\mu}S}{(K_S + S)} \cdot \frac{DO}{(DO + K_{SDO})}$	(17)
	$DO_S = 5.5555 \cdot 10^4 \frac{MW_{O_2}}{H_{eO_2}} \cdot P_{O_2} \cdot \beta$	(18)
	$DCD_S = 5.5555 \cdot 10^4 \frac{MW_{CO_2}}{H_{eCO_2}} \cdot P_{CO_2} \cdot \beta$	(19)
	$DN_S = 5.5555 \cdot 10^4 \frac{MW_{N_2}}{H_{eN_2}} \cdot P_{N_2} \cdot \beta$	(20)
pH		
	$ALK = [HCO_3] + 2[CO_3^{2-}] + [OH^-] - [H^+] + [NH_3]$	(21)
	$K_W = [OH^-][H^+]$	(22)
	$K_1 = \frac{[H^+][HCO_3^-]}{[H_2CO_3]}$	(23)
	$K_2 = \frac{[H^+][CO_3^{2-}]}{[HCO_3^-]}$	(24)
	$[H^+]^2 + [H^+][ALK - NH_3] - K_W - \left[K_1 + \frac{2K_1K_2}{[H^+]}\right]H_2CO_3 = 0$	(25)

Table 4—(Continued)

Species	Equations	
	$f_{CO_2} = \frac{[H_2CO_3]}{[H_2CO_3] + [HCO_3^-] + [CO_3^{2-}]}$	(26)
	$= \frac{1}{1 + \frac{K_1}{[H^+]} + \frac{K_1 K_2}{[H^+]^2}}$	(27)
	$DCD_f = DCD f_{CO_2}$	(28)
	$f_{NH_3} = \frac{1}{1 + \frac{[H^+] K_{NH_3}}{K_W}}$	(29)
	$[NH_3] = [NH_{3T}] \cdot f_{NH_3}$	(30)

Table 5—Nomenclature for model equations.

Symbol	Definition
C_∞^*	equilibrium dissolved concentration, g/m^3
CO_2	gas phase carbon dioxide concentration, g moles/m^3
DN	dissolved nitrogen, g/m^3
DCD	dissolved carbon dioxide, including bicarbonate, g/m^3, and carbonate, g/m^3
DO	dissolved oxygen, g/m^3
K_D	decay coefficient, h^{-1}
K_La	volumetric mass transfer coefficient, h^{-1}, includes α
K_S	half saturation coefficient for substrate, g/m^3
K_{SDO}	half saturation coefficient for DO, g/m^3
K_W	ion product in water
K_1	first k_{eq} for carbon dioxide
K_2	second k_{eq} for carbon dioxide
MW	molecular weight, g
N_2	gas phase nitrogen concentration, g moles/m^3
NH_3	undisassociated ammonia concentration at pH, g/m^3
NH_{3T}	total ammonia concentration, g/m^3
O_2	gas phase oxygen concentration, g moles/m^3
Q_L	liquid flow rate per stage, m^3
Q_G	volumetric gas flow rate, m^3
S	substrate, g/m^3
V_L	stage liquid volume, m^3
V_G	stage gas volume, m^3
Y	cell yield, mass X/mass S
$Y_{O_2'}$	oxygen consumed per unit S consumed
$Y_{O_2''}$	oxygen consumed per unit X oxidized
$Y_{CO_2'}$	mass of CO_2 produced per unit mass of S converted
$Y_{CO_2''}$	mass of CO_2 produced per unit mass of X oxidized
X	cell mass concentration, g/m^3
$\hat{\mu}$	maximum specific growth rate, h^{-1}
o	as subscript, denotes influent value
S	as subscript, denotes saturation concentration at system temperature and partial pressure
f	as subscript, denotes fraction total dissolved carbon dioxide as CO_2 or H_2CO_3
T	as subscript, denotes total pressure
SP	as subscript, denotes set point value of pressure

Sacramento wastewaters (total dissolved solids approximately 500 mg/L), the effects of activity coefficients were assumed to be negligible, and comparisons to a steady-state model[7] validate this assumption. Leakage flows are not shown in this description, but are treated as sinks in the continuity terms. The leakage flow rate was set equal to the measured leakage flow rate. Leakage for the Sacramento case affected the final results by less than 1%. Equation 21 is cubic with respect to $[H^+]$. This equation was reduced to a quadratic by iteratively solving for $[H^+]$ using trial values for the $[H^+]$ in the denominator.

Material balance equations were also written on total ammonia concentration, alkalinity, and inert solids, but are not shown herein. The secondary clarifier was modeled as a zero volume clarifier. Solids thickening was not modeled, as the clarifiers during the periods of the tests were never overloaded. The details of the model are in Tables 4, 5, and 6.

Model results. Figures 5 and 6 show the modeling results and the measured data for the first process water test. The data fit the model well, except in Stage 4, where the model predicts slightly lower oxygen purity. The fit is exceptionally good given that BOD_5 data were determined from analysis of samples that were collected at 24-hour intervals. Undoubtedly, model results would have been better if BOD_5 data were associated with samples collected every 4 hours.

Figures 7 and 8 show the second process water test results. The fit is still good but not as good as in the first

Table 6—Parameter values.

Parameter	Value	Parameter	Value
$\hat{\mu}$	0.20 h^{-1}	$Y_{CO_2'}$	1.37
Y	0.40	$Y_{CO_2''}$	1.95
$Y_{O_2'}$	1.42	K_D	0.004 h^{-1}
$Y_{O_2''}$	1.42	β	0.99
$Y_{NH_3'}$	0.039	$K_La_{CO_2}$	0.836 $K_La_{O_2}$
$Y_{NH_3''}$	0.1239	$K_La_{N_2}$	0.943 $K_La_{O_2}$
K_{SDO}	0.5 g/m^3	θ	1.024
K_s	50 g/m^3		

Table 7—Comparisons of model results and average of measured results.

	DO concentration, mg/L					Oxygen fraction, percent				
	Stage 1	Stage 2	Stage 3	Stage 4	Average	Stage 1	Stage 2	Stage 3	Stage 4	Average
Test 1										
Data	5.1	2.8	1.2	0.6	2.4	66.0	56.6	42.1	32.3	49.3
Model	4.6	1.7	1.1	1.2	2.1	68.6	55.1	41.0	25.8	47.6
Difference	0.5	1.1	0.1	−0.6	0.3	−2.6	1.5	1.1	6.5	1.7
Test 2										
Data	5.5	5.2	2.9	1.6	4.3	58.0	46.9	33.4	28.4	41.6
Model	4.3	4.9	5.8	6.2	5.3	65.8	48.9	36.0	27.5	44.5
Difference	1.2	0.3	−2.9	−4.6	−1.0	−7.8	−2.0	−2.6	0.9	−2.9

test. The second process water test was subjected to two major upsets in high purity oxygen feedrate and large changes in influent concentration and flowrate caused by the rainstorm. In addition, the BOD_5 data from the fifth day of testing were unavailable.

To fit the process data for both tests, a single consistent set of biological parameters was used (Table 6). Only the α factors and process inputs were changed. To improve fit, α factors were adjusted. Initially, the empirically determined α factors were used. The power-weighted α for Test 1 was 0.51. The α factor estimates for Stages 1 to 4 were 0.60, 0.55, 0.40, and 0.40, respectively. In the second test the α factors were reduced to 0.50, 0.50, 0.40, and 0.40 for Stages 1 to 4, respectively, which provided a power-weighted α factor of 0.46.

The α factors estimated in the dynamic modeling compare favorably with those determined experimentally: 0.51 and 0.46 as compared to 0.54 and 0.63 for Test Series 1 and 2, power weighted to match the process tests. Stage 4 contributed more to differences in model and test values. The process test results did not show the increasing trend the pilot scale tests did. This was probably because the power was constant in all pilot tests, and was selected to approximate the average power use in all four stages. In the process tests, the actual power utilization in Stage 1 was 2.1 times that used in Stage 4. Alpha factors can be a function of power density.[9]

Table 7 summarizes average model predictions and averaged experimental results for the two 7-day tests. Arithmetic averages were used for all four stages. Elsewhere, the power weighted average is used. Arithmetic average is useful in comparing goodness of fit. Generally, the agreement is quite good, within 1.7 percentage points for Test 1 and 2.9 percentage points for Test 2, for overall gas purity. The average DO was within 0.3 and 1.0 mg/L for Tests 1 and 2, respectively.

Performance Simulation

The model was used to simulate the warranty conditions. Water quality parameters were set to those shown in Table 2 and α and β factors were set at 0.8 and 0.95, respectively. The results using the manufacturer's suggested gas recirculation rates are shown in Table 8, for both the normal and reaeration modes. Using these recirculation rates, the process can transfer 125 tonnes/d but exceeds the specified power by 240 kW. For the reaeration mode the process also fails, because of insufficient DO in Stage 2.

By adjusting the turbine recirculation rates it is possible to meet the specifications. Table 9 shows that the process just meets the energy requirement while transferring 125 tonnes/d. For the reaeration mode, using all available blowers, the process very nearly meets the specification. The DO concentration in Stage 2 is nearly zero at the warranty condition. To meet the specification, it is necessary to increase the K_La in Stage 2 by increasing turbine horsepower. It was estimated that an 8% increase in turbine horsepower would provide sufficient K_La to meet the specifications.

Conclusions

Compliance testing, which took almost 3.5 years to complete for the Sacramento Regional Treatment Plant, is lengthy and expensive. The greatest difficulty in deter-

Table 8—Model results using manufacturer's gas recirculation rates.

Mode	Flow, m^3/s	Gas recirculation rates, m^3/h	O_2 uptake, tonnes/d	O_2[a] purity, percent	kW	DO,[a] mg/L
Normal	6.0	1275, 764, 713, 713	125	64	1840	5.71
Reaeration	4.0	660, 725, 725, 660	110	61	1656	7.88[b]
Reaeration	6.0	660, 725, 725, 660	156	60	1656	2.37[b]

[a] Power weighted, averaged over all stages.
[b] Insufficient DO, Stage 2.

Table 9—Model results using near-optimal gas recirculation rates.

Mode	Flow, m^3/s	Gas recirculation rates, m^3/h	O_2 uptake, tonnes/d	O_2[a] purity, percent	kW	DO,[a] mg/L
Normal	6.0	1200, 544, 357, 410	125	67	1589	6.0
Reaeration	4.0	1020, 1020, 1020, 663	107	61	1901	12.8
Reaeration	4.3	1020, 1020, 1020, 663	118	61	1901	11.1
Reaeration	4.8	1020, 1020, 1020, 663	130	61	1902	9.5
Reaeration	5.2	1020, 1020, 1020, 663	141	62	1903	7.93
Reaeration	6.0	1020, 1020, 1020, 663	160	62	1904	5.41[b]

[a] Power weighted, averaged over all stages.
[b] Insufficient DO, stage 2.

mining performance compliance is specifying process water and determining α factors. The intent in using the process water test was to warrant several parts of the process beyond the aeration system; however, the shortcoming in the turbine aeration system was obscured by the overwhelming difficulty of measuring ancillary variables, such as α factors.

Alpha factors should have been based on mixed liquor as opposed to the primary effluent. The greatest difficulty in conducting the process water test, particularly the second one, was that the plant was operating very near its maximum transfer rate and efficiency. The shortfall in turbine SOTR and SAE consumed the designer's safety factor; thus, the plant had to operate at optimum conditions to meet the warranty. Optimum conditions are generally unattainable for 7 consecutive days in a large plant like the regional plant.

It is recommended that future performance warranties for high-purity oxygen processes include an ASCE standard clean water test for the aeration system. The mass transfer characteristics of the aeration devices can be accurately established using this procedure. In this work, the ASCE procedure showed replication among tests of ±0.6%.

Planning for the clean water test required approximately 3 months and the test was completed within 1 week. It was possible to clean tanks previously used in the activated sludge process. Less planning would have been required if the ASCE test had been performed before startup, when clean tanks were available.

To ensure other aspects of high-purity oxygen process performance, separate warranties should be written in addition to the ASCE clean water test procedure.

Acknowledgments

Credits. The clean water test was conducted by AERTEC of North Andover, Mass. Larry Effinger performed the alpha factor testing, Ron Linden reviewed the modeling results, and Ed Edmission assisted with the clean water testing.

Authors. Michael K. Stenstrom is a professor in the Civil Engineering Department at UCLA. Wendell Kido is the superintendent of the Sacramento Regional Wastewater Treatment Plant (SRWTP). Robert Shanks is the project engineer for the SRWTP. Mike Mulkerin is the operations engineer for the SRWTP. Correspondence should be addressed to Michael K. Stenstrom, University of California, Civil Engineering Dept., 4173 Engineering I, Los Angeles, CA 90024-1600.

References

1. "A Standard for the Measurement of Oxygen Transfer in Clean Water." Am. Soc. Civil Eng., New York, N. Y. (1984).
2. Stenstrom, M. K., and Gilbert, R. G., "Effects of Alpha Beta and Theta Factors Upon the Design, Specification and Operation of Aeration Systems." *Water Res.* (G.B.), **15**, 643 (1981).
3. Hwang, H. J., and Stenstrom, M. K., "Evaluation of fine-bubble alpha factors in near full-scale equipment." *J. Water Pollut. Control Fed.*, **57**, 1142 (1985).
4. Doyle, M. L., *et al.*, "Pilot plant determination of oxygen transfer in fine bubble aeration." *J. Water Pollut. Control Fed.*, **55**, 1435 (1983).
5. Mueller, J. A., *et al.*, "Gas Transfer Kinetics of Pure Oxygen Systems." *J. Environ. Eng. Div. Am. Soc. Civ. Eng.*, **99**, 264 (1973).
6. McWhirter, J. K., and Vahldieck, N. P., "Oxygenation Systems Mass Transfer Design Considerations." In "The Activated Sludge Process." J. R. McWhirter (Ed.), Chemical Rubber Company Press, Inc., Cleveland, Ohio, **1**, 235 (1970).
7. Linden, R. K. S., "Model for Minimizing Energy Requirements in the Pure Oxygen Activated Sludge Process." PhD. dissertation, Univ. of California, Davis (1979).
8. Cliff, R. C., and Andrews, J. F., "Gas-Liquid Interactions in Oxygen Activated Sludge." *J. Environ. Eng. Div. Am. Soc. Civ. Eng.*, **112**, 61 (1986).
9. Stenstrom, M. K., and Hwang, H. J., "The Effect of Surfactants on Industrial Aeration Systems." *Proc. Indust. Waste Conf.*, Purdue Univ., West Lafayette, Ind., 902 (1979).
10. Weast R. C., "Handbook of Chemistry and Physics." Chemical Rubber Company, Cleveland, Ohio (1971).

APPENDIX 2 - CODE LISTING

```
      function afgen(ax,n,x,arr)
c*********************************************************************
c.. this function generates an arbitrary function defined by pairs of
c   data points contained in the array arr, with the number of points=n.
c   note that the function checks for proper data entry on the first
c   call, and checks to see if x is in the range defined data contained
c   in by the arr array. linear interpolation is used.
c*********************************************************************
      dimension arr(2,200),ax(5)
      common /com/time,fintim,prdel,outdel,delt,delt2,delmax,delmin,
     1method,keep,last,key(15)
c.. check for initial entry
      if(key(13)) 10,10,30
10    if(n-1) 11,11,12
11    write(6,1000) n
1000  format(//,' less than two data points were supplied for an afgen',
     1' function',//,' execution terminating')
      stop 31
12    ax(4)=1
c.. check to see if the data was entered correctly in ascending order
      do 13 i=2,n
13    if(arr(1,i).le.arr(1,(i-1))) goto 14
      goto 15
14    k=i-1
      write(6,1010) i,arr(1,i),k,arr(1,k)
1010  format(//,' the independent variable for an afgen function has ',
     1'not been',/,' entered in ascending order',/,' the',i3,'th point='
     2,2x,e17.6,2x,'while the',i3,'th point=',2x,e17.6,/,' execution ter
     3minating')
      stop 32
15    ax(1)=0.
      if(x.lt.arr(1,1)) ax(1)=1
      if(x.gt.arr(1,n)) ax(1)=-1.
      if(ifix(ax(1))) 16,17,16
16    write(6,1020) x,arr(1,1),arr(1,n)
1020  format(' the initial entry to an afgen function is out of range',
     1/,' the value of the independent variable is',e17.6,' while the',
     2/,' minimum value of the function is',e17.6, ' and the maximum',
     3/,' value of the function is',e17.6)
      if(ax(4)) 82,17,92
17    i=1
18    if(arr(1,i).ge.x) goto 20
      i=i+1
      goto 18
20    if(i.eq.1) goto 70
      i=i-1
      if(arr(1,i).lt.x) goto 70
      goto 20
c.. normal entry for afgen
30    if(x.lt.arr(1,1).or.x.gt.arr(1,n)) goto 80
      i=ifix(ax(2))
40    if(arr(1,i).ge.x) goto 50
      i=i+1
      goto 40
50    i=i-1
60    if(arr(1,i).lt.x) goto 70
      goto 50
70    i=i+1
      ax(2)=i
      afgen=arr(2,(i-1))+(x-arr(1,(i-1)))*(arr(2,i)-arr(2,(i-1)))/
     1(arr(1,i)-arr(1,(i-1)))
      ax(4)=1.
      goto 100
80    if(x.lt.arr(1,n)) goto 90
```

```
      if(ax(4)) 87,82,82
82    write(6,1030) time,x,arr(1,n)
1030  format(' independent variable for afgen function above range at',
     1' time=',e12.6,/,' independent variable=',e12.6,' maximum for this
     2 afgen function=',e12.6)
87    afgen=arr(2,n)
      ax(4)=-1
      ax(2)=n
      goto 100
90    if(ax(4)) 97,92,92
92    write(6,1040) time,x,arr(1,1)
1040  format(' independent variable for afgen function below range at',
     1' time=',e12.6,/,' independent variable=',e12.6,' minimum for this
     2 afgen function=',e12.6)
97    ax(2)=1
      ax(4)=-1
      afgen=arr(2,1)
100   return
      end
      subroutine antgrl(acx,xdot,x,m,n,iz)
c***********************************************************************
c.. this function performs the integration using first,second,and fourth
c.. order correct methods
c***********************************************************************
      dimension acx(5,iz:n),xdot(iz:n),x(iz:n)
      common /com/time,fintim,prdel,outdel,delt,delt2,delmax,delmin,
     lmethod,keep,last,key(15)
      common /coml/abserr,relerr,ptime,pltime,icount,iprint,ipoint
c.. check to see if this is the first call to the integrator. If
c   so, insert the initial condition into the storage array.
      if(key(13).le.0) then
        do 100 i=1,m
        acx(5,i)=x(i)
100     acx(1,i)=x(i)
       endif
c.. this section for first order integration (euler)
c***********************************************************************
      if(method.eq.1) then
        do 110 i=1,m
        x(i)=acx(1,i)+delt*xdot(i)
110     acx(1,i)=x(i)

c.. this section or second order integration  (modified euler)
c***********************************************************************
      elseif(method.eq.2) then
        if(keep.eq.1) then
c.. first half step for modified euler
          do 220 i=1,m
          x(i)=acx(1,i)+delt*xdot(i)
          acx(2,i)=acx(1,i)
          acx(3,i)=xdot(i)
220       acx(1,i)=x(i)
c.. second half step fort modified euler
        else
          do 240 i=1,m
          x(i)=(xdot(i)+acx(3,i))/2.*delt+acx(2,i)
240       acx(1,i)=x(i)
        endif
c.. this section for fourth order integration
c***********************************************************************
      elseif (method.eq.3) then
        if(keep.eq.1) then
310       do 320 i=1,m
          x(i)=acx(1,i)+delt2*xdot(i)
```

```
320       acx(2,i)=xdot(i)
        elseif(keep.eq.2) then
330       do 340 i=1,m
          x(i)=acx(1,i)+delt2*xdot(i)
340       acx(3,i)=xdot(i)
        elseif(keep.eq.3) then
350       do 360 i=1,m
          x(i)=acx(1,i)+xdot(i)*delt
360       acx(4,i)=xdot(i)
        else
370       do 380 i=1,m
      x(i)=delt/6.*(acx(2,i)+2.*acx(3,i)+2.*acx(4,i)+xdot(i))+acx(1,i)
380       acx(1,i)=x(i)
        endif
c*********************************************************************
c.. variable step rks integration
c.. check to see if the time step is being reduced and time is being
c   backed-up
      elseif (method.eq.4) then
c.. check to see if its the first time step
        if(key(1).eq.0) then
          do 400 i=1,m
400       acx(1,i)=acx(5,i)
        endif
        if(keep.eq.1) then
c.. successful integration step. begin the next first pass of the next
c   time step
          do 410 i=1,m
          acx(1,i)=acx(5,i)
          x(i)=acx(1,i)+delt2*xdot(i)
410       acx(2,i)=xdot(i)
        elseif(keep.eq.2) then
c.. second pass
          do 420 i=1,m
          x(i)=acx(1,i)+delt2*xdot(i)
420       acx(3,i)=xdot(i)
        elseif(keep.eq.3) then
c.. third pass
          do 430 i=1,m
          x(i)=acx(1,i)+xdot(i)*delt
430       acx(4,i)=xdot(i)
c.. the first, second, and third steps are identical to rks fixed-step
c
c.. fourth pass.  check to see if the error has been exceeded or if
c   doubling is possible
        elseif (keep.eq.4) then
          do 440 i=1,m
          x(i)=delt/6.*(acx(2,i)+2.*acx(3,i)+2.*acx(4,i)+
     1    xdot(i))+acx(1,i)
440       acx(5,i)=x(i)
c.. calculate an error predictor using simpson's rule
          do 450 i=1,m
          sint=delt/6.*(acx(2,i)+4.*acx(3,i)+xdot(i))+acx(1,i)
          switch=abs(x(i)-sint)/(abserr+relerr*x(i))
c.. these keys are summed in order to detect errors for all
c   integrations.
          if(switch.gt.0.5) key(9)=key(9)-1
450       if(switch.gt.1.) key(7)=key(7)-1
        endif
      endif
      return
      end
      subroutine centra
c.. this subroute controls the integration.  it is called at the end of
```

```
c    the dynamic section.
c***********************************************************************
      common /com/time,fintim,prdel,outdel,delt,delt2,delmax,delmin,
     1method,keep,last,key(15)
      common /com1/abserr,relerr,ptime,pltime,icount,iprint,ipoint
      if(method.eq.1) then
c.. first-order method
c***********************************************************************
         time=time+delt
         last=1
         keep=1
         key(1)=1
      elseif(method.eq.2) then
c.. second-order method (modified euler)
c***********************************************************************
         if(keep.eq.1) then
c.. first half step for modified euler
            time=time+delt2
            key(1)=1
            last=0
            keep=2
         elseif(keep.eq.2) then
c.. second half step for modified euler
            time=time+delt2
            key(1)=0
            keep=1
            last=1
         endif
      elseif(method.eq.3) then
c.. fourth-order integration (runge-kutta)
c***********************************************************************
         if(keep.eq.1) then
c.. first quarter step for runge-kutta
310         time=time+delt2
            key(1)=1
            last=0
            keep=2
         elseif(keep.eq.2) then
c.. second quarter step
320         keep=3
            last=0
         elseif(keep.eq.3) then
c.. third quarter step
330         keep=4
            last=0
            time=time+delt2
         elseif(keep.eq.4) then
c.. final quarter step
340         keep=1
            last=1
            key(1)=0
         endif
      elseif(method.eq.4) then
c.. variable step fourth-order rks
c***********************************************************************
         if(keep.eq.1) then
c.. first quarter step for runge-kutta
            time=time+delt2
            key(1)=1
            last=0
            keep=2
         elseif(keep.eq.2) then
c.. second quarter step
            keep=3
```

```
            last=0
         elseif(keep.eq.3) then
c.. third quarter step
            keep=4
            last=0
            time=time+delt2
         elseif(keep.eq.4) then
c.. final quarter step
c.. final step for rks variable step.  check to see if delt should be
            key(1)=0
            if(key(7).lt.0) then
c.. reduce the time step and backup the value of time
              time=time-delt
              delt=0.5*delt
              delt2=0.5*delt
              if(delt.lt.delmin)  then
c.. terminate the run due to the inability to find a reasonable delt
      write(6,1100) time,delt,delmin
1100  format(//,' *******execution terminating at time=',e17.6,'*******'
     1,/,' delt (',e12.6,') is less than delmin (',e12.6,')')
                 stop 36
              else
                 key(1)=1
                 keep=1
                 last=0
                 key(8)=key(10)
              endif
            else
              if(key(8).gt.0.or.key(9).lt.0) then
c.. error is ok but too great to allow doubling or doubling not
c   allowed.
                 key(8)=key(8)-1
                 last=1
                 keep=1
              else
c.. allow doubling if delt was not changed on the previous key(8)/4
c   time steps.
                 delt2=delt
                 delt=delt*2.
                 key(8)=key(10)
                 last=1
                 keep=1
              endif
            endif
         endif
      endif
      key(2)=0
      key(3)=0
      key(7)=1
      key(9)=0
      key(13)=1
      return
      end
      function svs(c)
c.. this funciton calculates the settling velocity as a function of
c   the suspended solids concentration.  (Metro Data)
      svs=3.8344*exp(-c*5.62e-04)
      return
      end
      function step(p)
c**********************************************************************
c.. this function simulates a unit step at time=p
c**********************************************************************
      common /com/time,fintim,prdel,outdel,delt,delt2,delmax,delmin,
```

```
     1method,keep,last,key(15)
      step=0.
      if((time-p).ge.0.) step=1.
      return
      end
      subroutine start
c***********************************************************************
c.. this subroutine writes out the initial program control
c.. statements  and sets up the constants.
c***********************************************************************
c.. Modified 3/18/88 to read abserr and relerr, added if-then's
c        in printing sequence
      common /com/time,fintim,prdel,outdel,delt,delt2,delmax,delmin,
     1method,keep,last,key(15)
      common /com1/abserr,relerr,ptime,pltime,icount,iprint,ipoint
      data ndbl/8/
c.. set all the keys to zero. Some routines require this, others
c   reset the appropriate key.
      do 10 i=1,15
10    key(i)=0
c.. round off the values of delt, prdel, and outdel to insure that
c   they are even multiples of fintim
      dn=fintim/delt
      n=dn+0.5
      delt=fintim/n
      delt2=0.5*delt
c.. round off outdel and delt
      dn=outdel/delt
      n=dn+0.5
      outdel=n*delt
      dn=prdel/delt
      n=dn+0.5
      prdel=n*delt
c.. initialize other counters and variables.
      iprint=1
      icount=1
      ipoint=12
      pltime=outdel
      ptime=prdel
      keep=1
c.. ndbl is the required number of successful integration  steps
c   before doubling is allowed in the variable step method(s).
      key(10)=ndbl*4
      key(8)=key(10)
      write(6,1000) fintim,delt,prdel,outdel
1000  format('1  MKS''s CSMP Program Version 1.5  February 1990',
     1///,80('*'),/,' Timer Variables',
     2t25,'Finish Time',t50,f8.2,/,1x,t25,'Delt',t49,
     3e17.6,/,1x,t25,'Print Interval',t51,f8.3,/,1x,t25,'Plot Interval',
     4t51,f8.3)
      if(method.eq.4) then
c.. calcualte the maximum and minimum values of delmin and delmax
        delmin=delt*1.e-05
        delmax=fintim*1.e-02
        if(delmax.gt.prdel) delmax=prdel
        if(delmax.gt.outdel) delmax=outdel
c.. read in the values of abserr and delmax
        read(8,1001,end=20) abserr
        read(8,1001,end=30) relerr
1001    format(f10.0)
        goto 40
20      abserr=0.01
30      relerr=0.01
40      write(6,1005) delmin,delmax,abserr,relerr
```

```
1005  format(1x,t25,'delmin',t49,e17.6,/,1x,t25,'delmax',t49,e17.6,
     1/,1x,t25,'abserr',t49,e17.6,/,1x,t25,'relerr',t49,e17.6)
      endif
      if(method.eq.1) then
        write(6,1010)
1010  format(/,' intergation method selected is simple euler (first'
     1,'-order-correct)')
      else if (method.eq.2) then
        write(6,1020)
1020  format(////,' Integration method selected is modified euler',
     1' (second-order correct)')
      else if (method.eq.3) then
        write(6,1030)
1030  format(////,' Integation method selected is runge-kutta',
     1' (fourth-order correct)')
      else if (method.eq.4) then
        write(6,1040)
1040  format(////,' Integration method selected is runge-kutta variable'
     1,' step',/,' (fourth-order correct)')
      endif
100   write(6,1100)
1100  format(/,' Note that the values of delt, prdel, and outdel ',
     2'have been rounded',/,' to make delt an even divisor of fintim,'
     3,' and prdel and outdel even',/,' multiples of delt',////)
      key(1)=0
      keep=1
      return
      end
      function pulse(s,p1,p2)
c***********************************************************************
c.. this functin simulates a pulse of length p2, triggered when
c   p1  becomes greater than zero
c***********************************************************************
      common /com/time,fintim,prdel,outdel,delt,delt2,delmax,delmin,
     1method,keep,last,key(15)
c..   this function simulates a pulse function of length p2 triggered
c     by p1>0.0.  s is a storage variable
      pulse=0.
      if(key(1)) 10,10,31
10    if(p1) 20,20,30
20    s=0.
      goto 100
30    pulse=1.
      s=p2
      goto 100
31    if(method-1) 35,40,35
35    if(last-1) 36,40,36
36    if(s-time) 100,70,70
40    if(s-time) 50,70,70
50    if(p1) 100,100,60
60    s=time+p2
70    pulse=1
100   return
      end
      function fmin(array,n)
c.. this function finds the minumum value of array elements
      dimension array(1)
      fmin=array(1)
      do 10 i=2,n
      if(array(i).lt.fmin) then
        fmin=array(i)
      endif
10    continue
      return
```

```
      end
      function fmax(array,n)
c.. this function defines the maximum value of array elements
      dimension array(1)
      fmax=array(1)
      do 10 i=2,n
      if(array(i).gt.fmax) then
        fmax=array(i)
      endif
10    continue
      return
      end
      real function limit(lo,hi,x)
      real lo*4
      if(lo.gt.hi) then
        write(6,1000) lo,hi
1000  format(' execution terminating in function limit due to',
     1' improper specification of limits.',/,' lower limit (',e17.6,
     2') is greater than the upper limit (',e17.6,')')
        stop 51
      endif
      if(x.le.lo) then
        limit=lo
      elseif(x.gt.hi) then
        limit=hi
      else
        limit=x
      endif
      return
      end
      function constr(a,b)
      constr=a
      if(a.lt.b) constr=0.
      return
      end
      function fcnsw(p0,p1,p2,p3)
c.. function switch function
      if(p0)10,20,30
10    fcnsw=p1
      return
20    fcnsw=p2
      return
30    fcnsw=p3
      return
      end
      function fco2(ph,ck1,ck2)
c.. this function calculates the fraction of the total carbon which
c   is in the h2co3 form.
      real*4 dk1,dk2,dph
c.. convert to double precision for the calculation
      dk1=ck1
      dk2=ck2
      dph=ph
      fco2=1.d+0/((dph*dk1*dk2/dph+dk1)/dph+1.d0)
      return
      end
      subroutine finsh(x,y)
      real*4 impuls,insw,ior,modint,nand,nlfgen,nor,not
      common /com/time,fintim,prdel,outdel,delt,delt2,delmax,delmin,
     1method,keep,last,key(15)
      common /com1/abserr,relerr,ptime,pltime,icount,iprint,ipoint
c.. check to see if it the end part of the integration step.
      if(keep.eq.1.and.last.eq.1) then
c.. check to see if Y > X
```

```
        if(y.ge.x) then
c.. set key(11) to a negative number and write a message
          key(11)=-1
          write(6,1000) time,x,y
1000  format(//,' Simulation terminating due to finish condition at time
     1=',e17.6,/,' for x=',e17.6,' is greater than or equal to y=',
     2e17.6)
        endif
      endif
      return
      end
      real function impuls(s,p1,p2)
c***********************************************************************
c.. this function simulates an impulse train starting at time=p1 and
c   repeating at every p2 units of time
c***********************************************************************
      common /com/time,fintim,prdel,outdel,delt,delt2,delmax,delmin,
     1method,keep,last,key(15)
      impuls=0.
      if(key(1)) 10,10,15
10    if(p1.le.0.25*delt) impuls=1.
      goto 100
15    if(method-1) 40,40,20
20    if(last-1) 100,40,100
40    if(time.lt.p1) goto 100
      if(p2) 50,50,60
50    impuls=1.
      goto 100
60    n=(time-p1)/p2+0.5
      if(abs(time-p1-float(n)*p2).le.(0.25*delt)) impuls=1.
100   s=impuls
      return
      end
      function insw(p0,p1,p2)
c.. input switch function
      implicit real*4 (a-h,o-z)
      real insw*4
      insw=p1
      if(p0.ge.0.0)insw=p2
      return
      end
      subroutine intgrl(ax,xdot,x)
c***********************************************************************
c.. this function performs the integration using first,second,and fourth
c.. order correct methods
c***********************************************************************
      dimension ax(5)
      common /com/time,fintim,prdel,outdel,delt,delt2,delmax,delmin,
     1method,keep,last,key(15)
      common /com1/abserr,relerr,ptime,pltime,icount,iprint,ipoint
c.. check and see if this is the first call to the integrator.
c   If so, insert the initial condition into the storage array.
      if(key(13).le.0)  then
        ax(1)=x
        ax(5)=x
      endif
      if(method.eq.1) then
c.. this section for first order integration (euler)
c***********************************************************************
        x=ax(1)+delt*xdot
        ax(1)=x
      elseif(method.eq.2) then
c.. this section or second order integration  (modified euler)
c***********************************************************************
```

```
         if(keep.eq.1) then
c.. first half step for modified euler
           x=ax(1)+delt*xdot
           ax(2)=ax(1)
           ax(3)=xdot
           ax(1)=x
         else
c.. second half step fort modified euler
           x=(xdot+ax(3))/2.*delt+ax(2)
           ax(1)=x
         endif
       elseif(method.eq.3) then
c.. this section for fourth order integration
c***********************************************************************
         if(keep.eq.1) then
           x=ax(1)+delt2*xdot
           ax(2)=xdot
         elseif(keep.eq.2) then
           x=ax(1)+delt2*xdot
           ax(3)=xdot
         elseif(keep.eq.3) then
           x=ax(1)+xdot*delt
           ax(4)=xdot
         elseif(keep.eq.4) then
           x=delt/6.*(ax(2)+2.*ax(3)+2.*ax(4)+xdot)+ax(1)
           ax(1)=x
         endif
c***********************************************************************
c.. variable step rks integration
c.. check to see if the time step is being reduced and time is being
c   backed-up
       elseif(method.eq.4) then
c.. check to see if delt has been decreased. If not save the
c   variable step output (ax(5)).
         if(key(1).eq.0) ax(1)=ax(5)
c.. successful integration step. begin the next first pass of the next
c   time step
         if(keep.eq.1) then
           x=ax(1)+delt2*xdot
           ax(2)=xdot
         elseif(keep.eq.2) then
           x=ax(1)+delt2*xdot
           ax(3)=xdot
         elseif(keep.eq.3) then
           x=ax(1)+xdot*delt
           ax(4)=xdot
         elseif(keep.eq.4) then
c.. fourth step.  check to see if the error has been exceeded or if
c   doubling is possible
           x=delt/6.*(ax(2)+2.*ax(3)+2.*ax(4)+xdot)+ax(1)
           ax(5)=x
c.. calculate an error predictor using simpson's rule.
           sint=delt/6.*(ax(2)+4.*ax(3)+xdot)+ax(1)
           switch=abs(x-sint)/(abserr+relerr*x)
c.. these keys are summed in order to detect errors for all
c   integrations.
           if(switch.gt.0.5) key(9)=key(9)-1
           if(switch.gt.1.) key(7)=key(7)-1
         endif
     . endif
       return
       end
       function ior(a,b)
c.. inclusive or function
```

```
      real*4 ior
      ior=1.0
      if(a.le.0.0.and.b.le.0.0)ior=0.0
      return
      end
c***********************************************************************
c***********************************************************************
c.. main simulation program
c***********************************************************************
c***********************************************************************
c.. the following dimension statement is for the secondary clarifier
      dimension vs(10),tflux(10),cdot(10),c(10),setflx(10),ac(5,10)
c.. the following dimension statement is for the integrators.
c   function. They define the storage that each intergrator
c   needs.  The storage is used to save the initial conditions
c   and the intermediate values.
      dimension ds(0:5),      s(0:5),         as(5,0:5),
     1dcstor(0:5),            cstor(0:5),     acstor(5,0:5),
     2dcact(0:5),             cact(0:5),      acact(5,0:5),
     3dci(0:5),               ci(0:5),        aci(5,0:5),
     4dcnv(0:5),              cnv(0:5),       acnv(5,0:5),
     5dcx(0:5),               cx(0:5),        acx(5,0:5)
      dimension dcnh3(0:5),  cnh3(0:5),      acnh3(5,0:5),
     1ddn2(0:5),              dn2(0:5),       adn2(5,0:5),
     2ddco2(0:5),             dco2(0:5),      adco2(5,0:5),
     3ddo2(0:5),              do2(0:5),       ado2(5,0:5),
     4co2dot(0:5),            co2(0:5),       aco2(5,0:5),
     5o2dot(0:5),             o2(0:5),        ao2(5,0:5),
     6as4po2(5),              atpl(5)
c   control system dimensions
      dimension drsetdo(4),   rsetdo(4),      arsetdo(5,4),
     1dosp(4)
      real*4 n2dot(0:5),        n2(0:5),        an2(5,0:5),
     1klalim(4,2)
c.. other dimensions
      dimension alpha(4),effd(4),q(4),qr(4),qt(0:4),tpres(0:4),
     1vi(4),vig(4),vlf(4),vgf(4),alkmol(4),sdco2(4),sdo2(4),sdn2(4),
     2pco2(4),pn2(4),po2(4),qgt(0:5),qg(4),co2kla(4),akla(4),hi(4),
     3ph(4),qleak(4),o2uptr(4),fstep(4),aqfun(5),qfun(2,200),
     4asbfun(5),sbfun(2,200),apbfun(5),pbfun(2,200)
      real*4 kla(4),klai(4),kcstor,kflow,koex,ko2sol,ko2str,kso2,
     1lkparm,mlss,mlvss,n2mw,n2kla(4),mod,kflowl,kflowm
c.. the following real, dimension, and common are for the internal
c   workings of the program and should not be changed they
c   communicate the value of time and other keys for progr control.
      real*4  impuls,limit
c.. coment out the next line to avoid warning messages
c       real*4  impuls,insw,ior,limit,modint,nand,nor,not
      common /com/time,fintim,prdel,outdel,delt,delt2,delmax,delmin,
     1method,keep,last,key(15)
c.. specifiy the maximum number of elements in a the clarifier
c   and the nubmer of aeration stages in series. They must be
c   integer constants.
      max=10
      nstage=4
      nstagl=nstage+1
c.. set the number of allowable paris of data for afgen. It must
c   be an integer constant, and match arrays, qfun.....
      maxaf=200
c***********************************************************************
c***********************************************************************
c.. initial section of the model
c***********************************************************************
c***********************************************************************
```

```
c.. read the integration and print control parameters.
c       delt = integration inverval
c       method = integration method
c              1 = euler;
c              2 = modified euler;
c              3 = fourth-order  runge-kutta;
c              4 = fourth-order runge-kutta, variable step.
c       prdel = print interval;
c       outdel = plot interval;
c       fintim = finish time;
c       keep = counter for integration step (automatically set).
c       key(i) = program control counters (see documentation)
c
c.. Open the file and leave it open for later use in the
c   centra routine for variable step information.
      open(unit=8,file='timers',status='old')
      rewind 8
      read(8,*) method
      read(8,*) delt
      read(8,*) prdel
      read(8,*) outdel
      read(8,*) fintim
c.. intialize the counters which control printing, plotting and
c.. program management. never change the statements in the following
c.. section
      ptime=prdel
      pltime=0.
      iplot=0
      iprint=1
      time=0.
      last=1
c*********************************************************************
c.. specify the model parameters
c*********************************************************************
      open(unit=9,file='params',status='old')
      rewind 9
c.. add ko2sol, ko2str
      read(9,1000) alpha(1),alpha(2),alpha(3),alpha(4),bci,beta,
     1b5tobu,bsstor,bstor,fcstrm,kcstor,klai(1),klai(2),klai(3),
     2klai(4),klalim(1,1),klalim(2,1),klalim(3,1),klalim(4,1),
     3klalim(1,2),klalim(2,2),klalim(3,2),klalim(4,2),
     4dosp(1),dosp(2),dosp(3),dosp(4),pgain,rgain,s4po2sp,pgano2,
     5rgano2,koex,ko2sol,ko2str,kso2,usol,ustor,y1co21,y1co22,
     6y1sol,y1stor,y2,ynh31,areaft,adepth,anelem,ameth,
     7vgf(1),vgf(2),vgf(3),vgf(4),vlf(1),vlf(2),vlf(3),vlf(4)
1000  format(f10.0)
      close (9)
      nelem=anelem+0.5
      meth=ameth+0.5
      if(nelem.gt.max) then
        write(6,*) 'The number of clarifier final elements has been'
        write(6,*) 'specified greater than the maximum possible.  The'
        write(6,*) 'maximum possible is being used (',max,')'
        nelem=max
      endif
      nelem1=nelem-1
c*********************************************************************
c.. specify all the initial conditions here
c*********************************************************************
      open(unit=10,file='inits',status='old')
      rewind (10)
c.. the initial conditions are stored in the first position of
c   the storage array
      read(10,*) (s(ij),     ij=0,nstage)
```

```
      read(10,*) (cstor(ij), ij=0,nstage)
      read(10,*) (cact(ij),  ij=0,nstage)
      read(10,*) (ci(ij),    ij=0,nstage)
      read(10,*) (cnv(ij),   ij=0,nstage)
      read(10,*) (cx(ij),    ij=0,nstage)
      read(10,*) (cnh3(ij),  ij=0,nstage)
      read(10,*) (do2(ij),   ij=0,nstage)
      read(10,*) (pco2(ij),  ij=1,nstage)
      read(10,*) (po2(ij),   ij=1,nstage)
      read(10,*) (ph(ij),    ij=1,nstage)
      read(10,*) (c(ij),     ij=1,nelem)
      close(10)
c***********************************************************************
c.. set the initial conditions for controllers
      do 10 i=1,nstage
10    rsetdo(i)=0.
      s4po2=0.
      s1tp1=0.
c.. specify the model inputs here.
c***********************************************************************
      open(unit=11,file='inputs',status='old')
      rewind (11)
      read(11,1020) sbodin,cacto,cio,cnh3o,cnvo,cstoro,
     1cxo5,calkao,do2o,lkparm,vbasin,pho,qmgd,fstep(1),fstep(2),
     2fstep(3),fstep(4),qrrat,srtsp,temp,tgasm,fpuro2,finput,
     3floamp,sbodam,pbodam
1020  format(f10.0)
      qmgdt=qmgd
      floamp=floamp/100.
      sbodam=sbodam/100.
      pbodam=pbodam/100.
      input=finput+0.5
      if(input.eq.3) then
c.. read the diurnal flow rate data
        open(unit=13,file='diurnal',status='old')
        rewind 13
c.. first read the number of data points
        read(13,*,end=20000) nqfun
          if(nqfun.gt.maxaf) then
            write(6,*) ' Number of flow input datapairs exceeds ',maxaf
            write(6,*) ' Number specified =',nqfun
            write(6,*) ' Execution stoping'
            stop 1
          else
            do 11 ij=1,nqfun
11          read(13,1030) qfun(1,ij),qfun(2,ij)
            read(13,*) npbfun
            do 12 ij=1,npbfun
12          read(13,1030) pbfun(1,ij),pbfun(2,ij)
            read(13,*) nsbfun
            do 13 ij=1,nsbfun
13          read(13,1030) sbfun(1,ij),sbfun(2,ij)
1030      format(2f10.0)
          endif
      endif
      close(11)
c.. convert the number of parallel basins to integer
      nbasin=vbasin+0.5
      nbasn1=nbasin+1
c.. convert the units from English to Metric.  Set the flows all to
c   zero to initialize.  They maybe controlled later.
      hclar=adepth*0.3048
c.. convert the flow rate from mgd to m3/hr
      qm3hr=qmgd*3785./(24.*nbasin)
```

```
c.. calculate the recycle flow rate. assume that it all goes to stage 1
      qr(1)=qm3hr*qrrat
c.. set the other recycle rates to zero.
      do 32 i=2,nstage
32    qr(i)=0.
c.. set the zeroth influent flow rate to zero
      qt(0)=0.
      qgt(0)=0.
c.. calculate all the other flow rates. this is necessary for calculatin
c   the initial conditions. these calcs are repeated in the dynamic sect
      do 35 ij=1,nstage
c.. convert fstep from a percent to a fraction
      fstep(ij)=fstep(ij)/100.
      vi(ij)=1./(vlf(ij)*0.02831685)
      vig(ij)=1./(vgf(ij)*0.02831685)
      q(ij)=qm3hr*fstep(ij)
      qt(ij)=q(ij)+qr(ij)+qt(ij-1)
      qg(ij)=0.
      qgt(i)=0.
35    continue
c.. calculate the total reactor volume
      vit=1./((vlf(1)+vlf(2)+vlf(3)+vlf(4))*0.02831685)
c.. open this file for the plot dataset. autocad reads this file.
      open(unit=12,file='output.dat')
c.. open this file for the additional plotting dataset. It is provided
c   in order to write and save things as required for special purposes.
c   The writes are added at the end of the pr subroutine.
c***********************************************************************
c.. specify the physical constants that never change
c***********************************************************************
c.. Theta factor for aeration
      theta=1.024
c.. Molecular Weights
      co2mw=44.009
      n2mw=28.013
      o2mw=31.998
c.. ratios of n2 and co2 klas to o2
      fklan2=0.943
      fklaco=0.836
c.. gas flow constant (linear weir).  Set the initial guess
c   proportional to the liquid flow rate.
      kflow=1. + 10.* qmgd
c.. Ideal gas constant
      r=8.2056e-05
      rt=r*(temp+273.15)
c.. The following relations are empirical fits of handbook data
c   for various physical/chemical constants. This avoids "table
c   look up"
c.. pkW of water
      pkw=14.943-4.2467e-02*temp+1.8234e-04*temp**2
      ckw=10**(-pkw)
c.. pk NH3
      pknh3=pkw - 10.059-3.1956e-02*temp
      cknh3=10**(-pknh3)
c.. pk1 and pk2 of H2CO3
      pk1=6.5793-1.3525e-02*temp+1.8126e-04*temp**2
      pk2=10.629-1.5054e-02*temp+1.2074e-04*temp**2
      ck1=10**(-pk1)
      ck2=10**(-pk2)
c.. vapor press H2O (atms)
      vph2o=(5.0538-2.1092e-02*temp+3.0783e-02*temp**2)/760.
c.. Henry's Law for O2, N2, and CO2. Include beta and
c     convert units.
      heo2=(2.5001+8.453e-02*temp-3.0576e-04*temp**2)/(55555.*
```

```
     1o2mw*1.e-04*beta)
      hen2=(5.2726+0.14661*temp-4.5931e-04*temp**2)/(55555.*
     1n2mw*1.e-04*beta)
      heco2=(0.72206+2.9690e-02*temp+2.6693e-04*temp**2)/(55555.*
     1co2mw*1.e-03*beta)
c.. convert the influent alkalinity from mg/L  CACO3 to mol
      alkao=calkao/50.e+03
c.. calculate stage-wise initial conditions and other inputs
      tpres(0)=1.002
      co2(0)=0.
      n2(0)=0.
      o2(0)=0.
      do 40 i=1,nstage
c.. total gas pressure in each stage, atms.  These are based upon
c   UCC's design recommendations.
      tpres(i)=1.002-(i-1)*0.0005
c..   leak gas flow rates
      qleak(i)=0.
c.. set the effective depths for aeration (=1.0 for surface).
      effd(i)=1.0
c.. hydrogen ion concentration
      hi(i)=10**(-ph(i))
      alkmol(i)=alkao
c.. initial concentrations for dissolved co2 and n2--assume
c   equilibrium.
      pn2(i)=tpres(i)-po2(i)-pco2(i)-vph2o
      n2(i)=pn2(i)/rt
      o2(i)=po2(i)/rt
      co2(i)=pco2(i)/rt
      dco2(i)=pco2(i)/(heco2*fco2(hi(i),ck1,ck2))
      dn2(i)=pn2(i)/hen2
c.. calculate the kla's for n2 and co2 as a function of the o2
c   klas.  copy this to the dynamic section if kla's change
      akla(i)=kla(i)*alpha(i)*theta**(temp-20.)
      n2kla(i)=alpha(i)*kla(i)*fklan2*theta**(temp-20.)
40    co2kla(i)=alpha(i)*kla(i)*fklaco*theta**(temp-20.)
c***********************************************************************
c.. perform all initial calculations here
c***********************************************************************
c.. aeration basin
      y2k=(1.-y2)*koex
      y1strp=1./y1stor
      y1solp=1./y1sol
      y1pst=ko2str*(1.-y1stor)/y1stor
      y1psol=ko2sol*(1.-y1sol)/y1sol
c.. empirical temperature correction factor. Copy to dynamic if
c   temperature changes during a simulation.
      tfac=theta**(temp-20.)
c.. secondary clarifier calcs.  Use inverse for faster floating point.
      ameter=1./(areaft*0.092903/nbasin)
      dxi=1./(hclar/nelem)
      avol=1./(ameter*dxi)
c.. convert the influent bod's from 5 to ultimate
      soavg=sbodin/b5tobu
      so=sbodin/b5tobu
      cxo=cxo5/b5tobu
      cxoavg=cxo5/b5tobu
c.. calculate the influent H ion concentration
      hio=1./(10**pho)
c.. influent co2.  calculate it from the influent pH and alkalinity
      dcdo=(alkao-(cnh3o/14.e+03)/(1.+hio*cknh3/ckw))*co2mw*1.e+03
c.. influent dissolved n2.  assume equilibrium with air.
      dn2o=0.791/hen2
c.. assume zero dissolved oxygen concentration.
```

```
      do2o=0.
c.. gas influent calcs
      tgash=tgasm*2000./(24.*nbasin)
      qgo=tgash*rt*454./(o2mw*(tpres(1)-vph2o)) +
     1tgash*(1.-fpuro2)/fpuro2*rt*454./(n2mw*(tpres(1)-vph2o))
      o2o=(tpres(1)-vph2o)*fpuro2/rt
      n2o=(tpres(1)-vph2o)*(1.-fpuro2)/rt
      co2o=0.
      qg(1)=qgo
c.. calculate approximate contoller gains for the pressure
c   controls.  Assume a 0.001 inch water pressure will create a flow
c   of qgo. Give 50% to the P gain and 50% to the I gain
      wgainp=0.5*qgo/0.001
      wgaini=0.1*qgo/0.001
      qgolim=3.*qgo
c.. set the maximum value of the weir coefficient. Use different
c   values for the pilot plant and the full scale plant.
      if(qmgd.le.1.) then
        kflowm=900.
      else
        kflowm=350000.
      endif
c.. calculate the initial mlvss, mlss, and initial recycle
c   concentration from initial conditions
      mlvss=cstor(nstage)+cact(nstage)+ci(nstage)+cx(nstage)
      mlss=mlvss+cnv(nstage)
c.. calculate the initial sludge wasting rate.
c   average the mlss concentrations across the stages. Use stage volumes
c   since concentrations and volumes will not be the same for each stag
      smlss=0.
      do 50 i=1,nstage
50    smlss=smlss+(ci(i)+cstor(i)+cx(i)+cact(i)+cnv(i))/vi(i)
      smass=0
      if(meth.gt.1) then
        do  51 i=1,nelem
51      smass=smass+c(i)
        smass=smass*avol+smlss
      else
        smass=smlss
      endif
c.. calculate the sludge wasting flow rate, qw
      qw=smlss/(c(nelem)*srtsp*24.)
c.. set the program control parameters in start
      call start
c**********************************************************************
c**********************************************************************
c.. dynamic section
c**********************************************************************
c**********************************************************************
c.. loop point.  statement 100 must always be the first
c.  statment in the dynamic section.
100   continue
c**********************************************************************
c.. input section. the time varying inputs are generated here.
c**********************************************************************
c.. adjust the input gas flow rates to control utilization
      tgasml=tgasm*(1.+pgano2*ds4po2+rgano2*s4po2)
      tgash=tgasml*2000./(24.*nbasin)
      qgo=tgash*rt*454./(o2mw*(tpres(1)-vph2o)) +
     1tgash*(1.-fpuro2)/fpuro2*rt*454./(n2mw*(tpres(1)-vph2o))
      qg(1)=qgo
c.. adjust the last stage pressure by modulating the weir coefficient
      kflowl=limit(0.,kflowm,(kflow+wgainp*dtpl+wgaini*sltpl))
c.. skip this section if it is not the last integration step.
```

```
c   skip if the inputs are constants (input=1)
      if(keep.eq.1.and.input.gt.1) then
        if(input.eq.2) then
c.. sine wave input of 24 hours.
c.. calcuate the varying inputs to each stage.
          asine=sin(time*0.130899)
          qfac=1.+floamp*asine
c.. each stage can have flow input
          do 110 i=1,nstage
110       q(i)=qm3hr*qfac*fstep(i)
c.. calculate the total flow in mgd for convenience later
          qmgdt=qmgd*qfac
c.. ratio recycle flow to stage 1 only
          qr(1)=qm3hr*(1.+floamp*asine)*qrrat
c.. soluble and particulate bods
          so=soavg*(1.+sbodam*asine)
          cxo=cxoavg*(1.+pbodam*asine)
        else if(input.eq.3) then
c.. diurnal input simulating Metro's flow rates and BOD.
c.. calcuate a time variable that is periodic over 24 hours.
          timep=mod(time,24.)
c.. soluble and particulate bods
          so=soavg*afgen(asbfun,nsbfun,timep,sbfun)
          cxo=cxoavg*afgen(apbfun,npbfun,timep,pbfun)
c.. flow calculation
          qfac=afgen(aqfun,nqfun,timep,qfun)
c.. calculate the total flow in mgd for convenience later
          qmgdt=qmgd*qfac
c.. ratio recycle flow to stage 1 only
          qr(1)=qm3hr*qfac*qrrat
c.. each stage can have flow input
          do 120 i=1,nstage
120       q(i)=qm3hr*qfac*fstep(i)
        else
          write(6,*) 'Incorrect input specification. input=',input
          stop 2
        endif
      endif
c***********************************************************************
c.. clarifier section of the activated sludge plant
c***********************************************************************
c.. aeration basin soluble species are assumed to be unaffected by
c.. the clarifier
      s(nstag1)=s(nstage)
      do2(nstag1)=do2(nstage)
      dco2(nstag1)=dco2(nstage)
      dn2(nstag1)=dn2o
      cnh3(nstag1)=cnh3(nstage)
c.. calculate the overflow rate in m/day
      ovel=(qt(4)-qw-qr(1))*0.09072*ameter
c.. calculate the effluent TSS or set it constant
      cover=16.
c.. calculate the mixed-liquor volatile and total suspended solids
c   concentration (stage 4).
      mlvss=cstor(nstage)+cact(nstage)+ci(nstage)+cx(nstage)
      mlss=mlvss+cnv(nstage)
      fluxin=(qt(nstage)*mlss-(qt(nstage)-qr(1)-qw)*cover)*ameter
c.. call the secondary clarifier subroutine.
      call settle(c,cdot,setflx,tflux,vs,dxi,fac1,fluxin,qr(1),qw,
     1ameter,tfac,mlss,nelem,max)
c.. calculate the recycle concentration.  The nstage+1 location
c   of each concentration array is used for the recycle concentration
      cact(nstag1)=cact(nstage)*fac1
      ci(nstag1)=ci(nstage)*fac1
```

```
      cnv(nstag1)=cnv(nstage)*fac1
      cstor(nstag1)=cstor(nstage)*fac1
      cx(nstag1)=cx(nstage)*fac1
c***********************************************************************
c.. aeration basin part of the activated sludge plant
c***********************************************************************
c.. oxygen feed control--based upon stage 4 purity
c   calculate the error
      ds4po2=s4po2sp-po2(4)
c.. total pressure in the first stage
      tp1=vph2o+pco2(1)+pn2(1)+po2(1)
c.. calculate the error
      dtp1=tp1-tpres(1)
      do 200 i=1,nstage
      j=i-1
c.. stage controller calculations
c   calculate the difference between do and the setpoint
      drsetdo(i)=dosp(i)-do2(i)
c   ratio up the klas for control. Set the gains to zero for no control.
      kla(i)=klai(i)*(1.+pgain*drsetdo(i)+rgain*rsetdo(i))
c   check to see that the limits are not exceeded.
      if(kla(i).lt.klalim(i,1)) kla(i)=klalim(i,1)
      if(kla(i).gt.klalim(i,2)) kla(i)=klalim(i,2)
c.. now calcuate the actual kla's correcting for temp and for the
c   other species.
      akla(i)=kla(i)*alpha(i)*tfac
      n2kla(i)=alpha(i)*kla(i)*fklan2*tfac
      co2kla(i)=alpha(i)*kla(i)*fklaco*tfac
c.. calculate factors used in more than one material balance.
c.. carbonaceous model
      fcstor=cstor(i)/(cact(i)+cstor(i)+ci(i))
      fcact=fcstor/(fcstor+1.)
      fx=cx(i)/(cx(i)+cact(i))
      fo2=limit(0.,40.,do2(i))/(do2(i)+kso2)
c     fo2=do2(i)/(do2(i)+kso2)
      f1=bsstor*cact(i)*s(i)*(fcstrm-fcstor)
      f2=usol*cact(i)*s(i)*fo2
      f3=ustor*cact(i)*fcact*fo2
      f4=bstor*cact(i)*fx/(fx+kcstor)
      f5=bci*cact(i)*fo2
c.. partial pressures for co2, o2, and n2
      pco2(i)=co2(i)*rt
      po2(i)=o2(i)*rt
      pn2(i)=n2(i)*rt
c.. saturation concentrations for co2, o2, and n2
c   multiply by an effective depth to account for subsurface
c   aeration
      sdco2(i)=effd(i)*pco2(i)/heco2
      sdo2(i)=effd(i)*po2(i)/heo2
      sdn2(i)=effd(i)*pn2(i)/hen2
c.. stripping rates for co2, o2 and n2
      fcdco2=fco2(hi(i),ck1,ck2)
      strpco=co2kla(i)*(sdco2(i)-dco2(i)*fcdco2)
      strpo2=akla(i)*(sdo2(i)-do2(i))
      strpn2=n2kla(i)*(sdn2(i)-dn2(i))
c.. calculate the exit gas flow rate. Limit backflow
      qgtemp=kflow1*(vph2o+pco2(i)+pn2(i)+po2(i)-tpres(i))
      qgt(i)= limit(0.,qgolim,qgtemp)
c.. Alkalinity
      alkmol(i)=alkao+(cnh3(i)-cnh3o)/14.e+03
c.. uncomment the following writes for debugging
c      write(6,4001) i,alkmol(i),dco2(i),cnh3(i),ck1,ck2,ckw,cknh3
c4001  format(' reactor no=',i4,/,
c     1        ' alkmol    =',e17.6,/,
```

```
c     2        ' dco2       =',e17.6,/,
c     3        ' cnh3       =',e17.6,/,
c     4        ' ck1        =',e17.6,/,
c     5        ' ck2        =',e17.6,/,
c     6        ' ckw        =',e17.6,/,
c     7        ' cknh3      =',e17.6,/)
c.. hydrogen ion concentration, as a function of alk
      call phcal(hi,alkmol,dco2,cnh3,ck1,ck2,ckw,cknh3,i)
c.. calcuate the ph concentration if hi is positive; otherwise set it
c   to an artifically low number. This will allow the model to continue
c   to run for debbugging purposes, which is the only time one should
c   have a negative hi concentration.  We may want to change this later.
      if(hi(i).gt.0.) then
         ph(i)=-alog10(hi(i))
      else
         ph(i)=5.1d+00
         hi(i)=7.9d-06
      endif
c.. add the liquid flows
      qt(i)=q(i)+qr(i)+qt(j)
c.. Soluble substrate balance
      ds(i)=vi(i)*(q(i)*so+qt(j)*s(j)+qr(i)*s(nstag1)-
     1qt(i)*s(i)) - f1 -f2*y1solp
c.. Stored Mass balance
      dcstor(i)=vi(i)*(q(i)*cstoro+qt(j)*cstor(j)+qr(i)*s(nstag1)-
     1qt(i)*cstor(i)) + f1 - f3*y1strp + f4
c.. Active Mass balance
      dcact(i)=vi(i)*(q(i)*cacto+qt(j)*cact(j)+qr(i)*cact(nstag1)-
     1qt(i)*cact(i)) + f2 + f3 - f5
c.. Inert Mass balance
      dci(i)=vi(i)*(q(i)*cio+qt(j)*ci(j)+qr(i)*ci(nstag1)-
     1qt(i)*ci(i)) + f5*y2
c.. Non-volatile Mass balance
      dcnv(i)=vi(i)*(q(i)*cnvo+qt(j)*cnv(j)+qr(i)*cnv(nstag1)-
     1qt(i)*cnv(i))
c.. Stored Substrate Balance
      dcx(i)=vi(i)*(q(i)*cxo+qt(j)*cx(j)+qr(i)*cx(nstag1)-
     1qt(i)*cx(i))-f4
c.. Ammonia Balance (no nitrification allowed)
      dcnh3(i)=vi(i)*(q(i)*cnh3o+qt(j)*cnh3(j)+qr(i)*cnh3(nstag1)-
     1qt(i)*cnh3(i)) + (f5*(1.-y2) - f2 - f3)*ynh31
c.. Dissolved oxygen balance
      o2uptr(i)= f2*y1psol + f3*y1pst + f5*y2k
      ddo2(i)=vi(i)*(q(i)*do2o+qt(j)*do2(j)+qr(i)*do2(nstag1)-
     1qt(i)*do2(i)) + strpo2 -o2uptr(i)
c.. Dissolved nitrogen
      ddn2(i)=vi(i)*(q(i)*dn2o+qt(j)*dn2(j)+qr(i)*dn2o-
     1qt(i)*dn2(i)) + strpn2
c.. Dissolved CO2
      ddco2(i)=vi(i)*(q(i)*dcdo+qt(j)*dco2(j)+qr(i)*dco2(nstag1)-
     1qt(i)*dco2(i))+ strpco + o2uptr(i)*1.375
c.. Gas phase balances
c   CO2 Gas Phase
      co2dot(i)=vig(i)*(qgt(j)*co2(j)+qg(i)*co2o-(qgt(i)+qleak(i))
     1*co2(i))-strpco*vig(i)/(vi(i)*co2mw)
c.. Nitrogen Gas Phase
      n2dot(i)=vig(i)*(qgt(j)*n2(j)+qg(i)*n2o-(qgt(i)+qleak(i))
     1*n2(i))-strpn2*vig(i)/(vi(i)*n2mw)
c.. Oxygen Gas Phase
      o2dot(i)=vig(i)*(qgt(j)*o2(j)+qg(i)*o2o-(qgt(i)+qleak(i))
     1*o2(i)) -strpo2*vig(i)/(vi(i)*o2mw)
200   continue
      i=4
c.. calculate sludge age and fm ratio
```

```
c   average the mlss concentrations across the stages.  Use the each
c   stage volume since they may not be the same.
       smlss=0.
       do 210 i=1,nstage
210    smlss=smlss+(ci(i)+cstor(i)+cx(i)+cact(i)+cnv(i))/vi(i)
       smass=0
       if(meth.gt.1) then
         do 220 i=1,nelem
220      smass=smass+c(i)
         smass=smass*avol+smlss
       else
         smass=smlss
       endif
c.. calculate the total sludge waste mass rate required to maintain SRT.
       wmass=smass/(srtsp*24.)
c.. subtract off the effluent TSS mass wasted
       emass=(qt(nstage)-qr(1)-qw)*cover
       qw=(wmass-emass)/c(nelem)
c.. calculate the o2 utilization
c.. o2 input
       o2in=qg(1)*fpuro2
       o2out=qgt(4)*po2(4)
       o2util=(o2in-o2out)/o2in
       trig1=-0.5+impuls(s1,0.0,6.0)
       plse=pulse(s2,trig1,3.0)
c*********************************************************************
c.. integrator statement section of the model. place all intgrl's here
c*********************************************************************
c.. secondary clarifier
       call antgrl(ac,cdot,c,10,10,1)
c.. substrate, stored, active, biologically inert, non-volatile, and particulate
c    masses.
       call antgrl(as,ds,s,4,5,0)
       call antgrl(acstor,dcstor,cstor,4,5,0)
       call antgrl(acact,dcact,cact,4,5,0)
       call antgrl(aci,dci,ci,4,5,0)
       call antgrl(acnv,dcnv,cnv,4,5,0)
       call antgrl(acx,dcx,cx,4,5,0)
c.. ammonia, dissolved n2, dissolved oxygen and dissolved co2
       call antgrl(acnh3,dcnh3,cnh3,4,5,0)
       call antgrl(adn2,ddn2,dn2,4,5,0)
       call antgrl(ado2,ddo2,do2,4,5,0)
       call antgrl(adco2,ddco2,dco2,4,5,0)
c.. gas phase nitrogen, oxygen and co2
       call antgrl(an2,n2dot,n2,4,5,0)
       call antgrl(aco2,co2dot,co2,4,5,0)
       call antgrl(ao2,o2dot,o2,4,5,0)
c.. integral part of PI controlers for DO
       call antgrl(arsetdo,drsetdo,rsetdo,4,4,1)
c.. integral part of PI controller for O2 utilization
       call intgrl(as4po2,ds4po2,s4po2)
c.. integral part of PI controller for last stage pressure
       call intgrl(atp1,dtp1,sltp1)
c.. subroutine centra controls the integration and must always be placed
c   at the end of the dynamic section,  after all integrations.
       call centra
       if(last.ne.1) goto 100
c*********************************************************************
c*********************************************************************
c.. this part of the dynamic section is reserved for printing, plotting,
c   and performing other calculations which are required only at the end
c   of an integration step.  this section is skipped unless 'last=1'
c*********************************************************************
c*********************************************************************
```

```
500   last=1
c.. check to see if its time to print
      if(keep.eq.1) then
       if((ptime-time).le.delt2.or.key(11).lt.0) then
         iprint=iprint+1
         ptime=iprint*prdel
         key(2)=1
         ip=1
      call pr(akla,alkmol,alpha,b5tobu,c,cact,ci,
     1cnh3,cnv,cover,cstor,cx,cxo,do2,emass,kla,koex,max,nbasin,nelem,
     2nstage,o2uptr,o2util,pco2,ph,plse,pn2,po2,q,qg,qgo,qgt,qmgd,
     3qmgdt,qr,qt,qw,s,smass,so,srtsp,tp1,tpres,trig1,vi,vig,vph2o,ip)
         endif
c.. check to see if its time to save the variables for plotting
       if((pltime-time).le.delt2.or.key(11).lt.0) then
         iplot=iplot+1
         pltime=iplot*outdel
         ip=2
      call pr(akla,alkmol,alpha,b5tobu,c,cact,ci,
     1cnh3,cnv,cover,cstor,cx,cxo,do2,emass,kla,koex,max,nbasin,nelem,
     2nstage,o2uptr,o2util,pco2,ph,plse,pn2,po2,q,qg,qgo,qgt,qmgd,
     3qmgdt,qr,qt,qw,s,smlss,so,srtsp,tp1,tpres,trig1,vi,vig,vph2o,ip)
         endif
       endif
c.. check to see if time is greater than fintim
c   and that simulation ends on a printout.
      if((fintim-time).le.delt2.and.key(2).eq.1) then
         go to 10000
      endif
c.. this statement ends the dynamic section due to completion of
c   a finish condition.
      if(key(11).lt.0) goto 10000
      goto 100
c.. print out the final values after a finish condition.
      ip=1
      call pr(akla,alkmol,alpha,b5tobu,c,cact,ci,
     1cnh3,cnv,cover,cstor,cx,cxo,do2,emass,kla,koex,max,nbasin,nelem,
     2nstage,o2uptr,o2util,pco2,ph,plse,pn2,po2,q,qg,qgo,qgt,qmgd,
     3qmgdt,qr,qt,qw,s,smlss,so,srtsp,tp1,tpres,trig1,vi,vig,vph2o,ip)
      ip=2
      call pr(akla,alkmol,alpha,b5tobu,c,cact,ci,
     1cnh3,cnv,cover,cstor,cx,cxo,do2,emass,kla,koex,max,nbasin,nelem,
     2nstage,o2uptr,o2util,pco2,ph,plse,pn2,po2,q,qg,qgo,qgt,qmgd,
     3qmgdt,qr,qt,qw,s,smlss,so,srtsp,tp1,tpres,trig1,vi,vig,vph2o,ip)
c.. terminal section of the model
c***********************************************************************
c.. reserver numbers in excess of 100000 for the terminal section
c***********************************************************************
10000 continue
      open(unit=16,file='ninits',status='unknown')
      rewind (16)
c.. write out the final values of the integration variables to allow
c   easy restarting from the last conditions.
      write(16,1201) (s(ij),     ij=0,nstage)
1201  format(5f9.1,'     Soluble Substrate')
      write(16,1202) (cstor(ij), ij=0,nstage)
1202  format(5f9.1,'     Stored Mass')
      write(16,1203) (cact(ij),  ij=0,nstage)
1203  format(5f9.1,'     Active Mass')
      write(16,1204) (ci(ij),    ij=0,nstage)
1204  format(5f9.1,'     Biologically Inert Mass')
      write(16,1205) (cnv(ij),   ij=0,nstage)
1205  format(5f9.1,'     Non-volatile Mass')
      write(16,1206) (cx(ij),    ij=0,nstage)
```

```
1206  format(5f9.1,'    Stored Substrate')
      write(16,1207) (cnh3(ij),  ij=0,nstage)
1207  format(5f9.1,'    Ammonia Concentration')
      write(16,1208) (do2(ij),   ij=0,nstage)
1208  format(5f9.1,'    Dissolved Oxygen')
      write(16,1209) (pco2(ij),  ij=1,nstage)
1209  format(4f9.2,9x,'    Carbon Dioxide')
      write(16,1210) (po2(ij),   ij=1,nstage)
1210  format(4f9.2,9x,'    Oxygen Purity')
      write(16,1211) (ph(ij),    ij=1,nstage)
1211  format(4f9.2,9x,'    Basin pHs ')
      write(16,1212) (c(ij),     ij=1,nelem)
1212  format(10f9.1,'    Clarifier Solids')
      close(12)
      close(16)
      stop
20000 write(6,*) 'The program was unable to open the input file'
      write(6,*) '(diurnal) which contains the input data pairs'
      write(6,*) 'for BOD and FLOW'
      stop 10
      end
      function outsw(p0,p1,p2,p3)
c.. output switch
      if(p0.lt.0.0)go to 50
      p3=p1
      p2=0.
      go to 90
50    p2=p1
      p3=0.
90    outsw=p2
100   return
      end
      subroutine phcal(hi,alk,co2,cnh3,ck1,ck2,ckw,cknh3,ir)
c.. this function calculates the ph of a dilute solution in a closed
c   biox reactor.  the calculation is implicit
      real*8 dk1,dk2,dco2,dph,dnh3,dalk,b,c,zgess,dknh3,
     1dkw
      dimension cnh3(0:5),co2(0:5),alk(4),hi(4)
      common /com/time,fintim,prdel,outdel,delt,delt2,delmax,delmin,
     1method,keep,last,key(15)
c.. set the first guess equal to previous pH
      dph=hi(ir)
c.. set the guess to a plausibile pH
      zgess=1.e-06
c.. convert the single precision args to double precision
      dk1=ck1
      dk2=ck2
      dkw=ckw
c.. also convert co2 and nh3 to molar concentration.
      dco2=co2(ir)/44009.d+00
      dalk=alk(ir)
      dknh3=cknh3
      dnh3=cnh3(ir)/14.d+03
c.. quadratic coefficients
      b=dalk-dnh3/(1.d+00+dph*dknh3/dkw)
      iter=0
10    c=-dkw- (dk1 + 2.d+00 * dk1*dk2/zgess)*dco2
c.. calc the ph
      dph= (-b+dsqrt(b**2 - 4.d+00*c))/2.d+00
      if(dabs(dph-zgess).gt.1.d-12) goto 20
c.. normal convergence
        hi(ir)=dph
        return
20    if(iter.gt.10 ) goto 30
```

```
        zgess=dph
        b=dalk-dnh3/(1.+dph*dknh3/dkw)
        iter=iter+1
        goto 10
c.. no convergence
30      write(6,1000) zgess,dph,iter
1000    format(' non convergence in ph calculation.',/,
     1  1x,' execution stopping',/,' final guess for ph =',d17.6,
     2  /,' final calc for ph =',d17.6,/,' iteration number=',i5)
        hi(ir)=dph
        write(6,*) 'alka=',alk,' co2=',co2
      stop 49
      end
      subroutine pr(akla,alkmol,alpha,b5tobu,c,cact,ci,
     1cnh3,cnv,cover,cstor,cx,cxo,do2,emass,kla,koex,max,nbasin,
     2nelem,nstage,o2uptr,o2util,pco2,ph,plse,pn2,po2,q,qg,qgo,qgt,
     3qmgd,qmgdt,qr,qt,qw,s,smlss,so,srtsp,tp1,tpres,trig1,vi,vig,
     4vph2o,iplace)
c..
c.  This subroutine prints and prints output to a file for
c   plotting later. It is created as a subroutine solely to
c   reduce the size of the main program. ip=1 for printing
c   and ip .ne. 1 for printing to a file (for plotting later).
c..
c   This routine will generate warning messages since not
c   all of the variables passed into the routine are used
c   or printed at the present time.
c
c.. integrator dimensions
      dimension s(0:5),cstor(0:5),cact(0:5),ci(0:5),cnv(0:5),
     1cx(0:5),cnh3(0:5),do2(0:5)
      real*4 kla(4),koex,mlss,mlvss
c.. other dimensions
      dimension alpha(4),q(4),qr(4),qt(0:4),tpres(0:4),vi(4),vig(4),
     1alkmol(4),pco2(4),pn2(4),po2(4),qgt(0:5),qg(4),akla(4),ph(4),
     2o2uptr(4),c(max)
c.. the following real, dimension, and common are for the internal
c   workings of the program and should not be changed they
c   communicate the value of time and other keys for progr control.
      common /com/time,fintim,prdel,outdel,delt,delt2,delmax,delmin,
     1method,keep,last,key(15)
      common /com1/abserr,relerr,ptime,pltime,icount,iprint,ipoint
c..
c  check to see if this call is for printing or plotting.
      if (iplace.eq.1) then
c.. printing.
         nr=6
      else
         nr=12
      endif
         write(nr,1000)
1000  format(
     1' Stage  Sub   Cact    Cx   Cstor Cinert   Cnh3    Cnv    DO',
     1'       pH   Alka')
         do 10 i=1,nstage
10       write(nr,1010) i,s(i),cact(i),cx(i),cstor(i),ci(i),cnh3(i),
     1   cnv(i),do2(i),ph(i),alkmol(i)
1010  format(2x,i2,1x,f6.1,1x,f6.0,1x,f6.0,1x,f6.1,1x,f6.1,2x,f5.1,1x,
     1        f6.1,3x,f4.1,4x,f3.1,1x,f7.5)
         write(nr,1020)
1020  format(//,
     1' Stage  ppO2  ppCO   ppN2   O2up    Kla  MLVSS  MLSS')
         o2mass=0.
         do 20 i=1,nstage
```

```
c.. print the o2uptake in units of lb/day if the flow is less than 1 mgd
c   or tons per day if it's more than 1 mgd.  This is necessary to
c   conform to the existing output formats so that we don't have to
c   change the plotter program.
          if(qmgd.gt.1.) then
             o2ton=nbasin*o2uptr(i)/(vi(i)*37833.3)
             o2mass=o2mass+o2ton*2000.
          else
             o2ton=nbasin*o2uptr(i)/(vi(i)*18.92)
             o2mass=o2mass+o2ton
          endif
          mlvss=cact(i)+ci(i)+cstor(i)+cx(i)
          mlss=mlvss+cnv(i)
          pvol=mlvss/mlss
          fm=24.*(q(1)+q(2)+q(3)+q(4))*(cxo+so)*b5tobu/(pvol*smlss)
20        write(nr,1030) i,po2(i),pco2(i),pn2(i),o2ton,kla(i),mlvss,
     1    mlss
1030      format(2x,i2,1x,f6.2,2x,f5.2,2x,f5.2,1x,f6.1,2x,f5.1,1x,
     1           f6.0,1x,f6.0)
          write(nr,1040)
1040      format(/,' Secondary Clarifier Output',/)
          do 30 k=1,nelem
30        write(nr,1050) k,c(k)
1050      format(1x,'c(',i2,')=',e17.6,' mg/l')
          write(nr,1060) trigl,plse,time,delt
1060      format(/,' trigl=',e12.6,3x,' pulse=',e12.6,' time=',f8.2,
     1    ' delt=',e12.6)
          write(nr,1070)
1070      format(/)
c.. calculate the terms necessary for calibration eg. lbo2/lb bod etc.
c   Skip if plotting since the plotter program currently has no provisions
c   for these outputs.
        if(iplace.eq.1) then
c.. bod5 removed per day
          bod5r=qmgdt*8.34*((cxo+so)-(s(4)+cover*pvol*koex))*b5tobu
c.. o2 consumed per unit of bod5 removed
          o2ratio=o2mass/bod5r
c.. waste sludge mass (lb/day)
          wastx=nbasin*pvol*(qw*c(nelem)+emass)*24./454.
c.. waste sludge mass per unit of bod5 removed
          wastxr=wastx/bod5r
c.. check srt
          srtc=smlss/((qw*c(nelem)+emass)*24.)
          qgscfm=qgo*0.5885
          write(nr,1080) qmgdt,so,qgscfm,cxo,srtc,fm
1080      format(' Flow (MGD)',t28,f9.2,t40,'Influent Sol BODU',
     1    t65,f9.0,/,' O2 Gas Flow (SCFM)',t28,f9.2,t40,
     2    'Influent Part.  BODU',t65,f9.0,/,' SRT (days)',t28,f9.1,
     3    t40,'F:M ratio (1/day)',t65,f9.2)
           write(nr,1090) bod5r,o2mass,wastx,o2ratio,wastxr,o2util
1090       format(' BOD5 Removed (#/day)',t28,f9.0,t40,'O2 ',
     1     'Consumed (#/day)',t65,f9.0,/,' Waste VSS (#/day)',
     2     t28,f9.0,t40,'O2 Cnsd/BOD5 Rmvd(#/#) ',t65,
     3     f9.2,/,' Waste VSS/BOD5 Rmvd(#/#)',t28,f9.2,t40,
     4     'O2 Utilization (frac)',t65,f9.2)
          write(nr,1100)
1100      format(/)
        endif
        return
        end
      function qntzr(q,e)
      qout=ifix(0.5+abs(e)/q)
      qntzr=q*qout
      if(e.lt.0.) qntzr=-qntzr
```

```
      return
      end
      subroutine settle(c,cdot,setflx,tflux,vs,dxi,facl,fluxin,qr,
     1qw,ameter,tclar,mlss,nelem,max)
c.. this subroutine simulates the settler using the
c   Bryant/Stenstrom/etc 1 d model
      dimension c(max),cdot(max),vs(max),tflux(max),setflx(max)
      real*4 mlss
      neleml=nelem-1
      do 10 i=1,nelem
      vs(i)=svs(c(i))*tclar
10    setflx(i)=vs(i)*c(i)
c.. calculate the total fluxes and derivates in each segment of the
c   settler
c.. calculate the underflow velocity in m/hr
      u=(qr+qw)*ameter
c.. first section
      tflux(1)=u*c(1)+aminl(setflx(1),setflx(2))
      cdot(1)=(fluxin-tflux(1))*dxi
c.. middle sections
      do 20 i=2,neleml
      tflux(i)=u*c(i)+aminl(setflx(i),setflx(i+1))
20    cdot(i)=(tflux(i-1)-tflux(i))*dxi
c.. bottom element
      cdot(nelem)=(tflux(nelem-1)-u*c(nelem))*dxi
c.. calculate the thickening factor
      facl=c(nelem)/mlss
      return
      end
      function sine(pl,p2,p3)
c*****************************************************************
c.. this function simulates a sine wave when time is greater than
c   pl with p2 randian/time and lag of p3 radians
c*****************************************************************
      common /com/time,fintim,prdel,outdel,delt,delt2,delmax,delmin,
     1method,keep,last,key(15)
      sine=0.
      if(time.ge.pl) then
         sine=sin(p2*(time-pl)+p3)
      endif
      return
      end
```

APPENDIX 3 - INPUT FILES FOR THE HPO PILOT PLANT

```
13
0.        1.1       time= 12 midnight ; this file has format 2f10.0
2.        1.02      time= 2 AM        ; the time should appear in the
4.        0.90      time= 4 AM        ; first 10 columns with a decimal
6.        0.78      time= 6 AM        ; point. In the second 10 columns
8.        0.71      time= 8 AM        ; the normalized flow or bod should
10.       0.83      time=10 AM        ; appear with a decimal point. For
12.        1.00     time=12 Noon      ; example, "2.        1.1" means that
14.        1.1      time= 2 PM        ; the flow or bod was 110% of the mean
16.        1.12     time= 4 PM        ; at 2 AM.  The number of points is
18.        1.13     time= 6 PM        ; specified by an integer that must
20.        1.15     time= 8 PM        ; appear on the first line of the file
22.        1.10     time=10 PM        ; and on the first line preceeding the
24.        1.1      time=12 midnight  ; bod data.  This integer tells the
14                                    ; the program how many data pairs to
0.        1.00                        ; to read.  The time can be entered in
1.5       1.06                        ; any arbitrary spacing. It must be
3.5       1.11                        ; in ascending order.  The time spacing
5.5       0.90                        ; for flow and bod do not have to match.
7.5       1.06                        ; blank lines are not permitted.
9.5       0.94
11.5      1.06           Particulate BOD
13.5      1.18
15.5      1.02
17.5      0.94
19.5      0.92
21.5      0.85
23.5      0.94
24.       0.97
14
0.        1.20           Soluble     BOD
1.5       1.23
3.5       1.58
5.5       1.43
7.5       1.14
9.5       0.45
11.5      0.39
13.5      0.39
15.5      0.94
17.5      1.04
19.5      1.17
21.5      1.06
23.5      1.17
24.       1.185
```

0.	0.3	25.4	10.1	4.0	Soluble Substrate
0.	30.3	49.9	73.4	84.8	Stored Mass
0.	1083.5	371.3	376.9	380.7	Active Mass
0.	1935.1	678.5	678.5	678.6	Biologically Inert Mass
0.	493.8	173.3	173.3	173.3	Non-volatile Mass
0.	6.0	54.6	30.7	14.1	Stored Substrate
0.	45.6	46.7	46.0	45.5	Ammonia Concentration
0.	6.0	6.0	6.0	6.0	Dissolved Oxygen
0.06	0.07	0.08	0.09		Carbon Dioxide
0.89	0.79	0.73	0.69		Oxygen Purity
6.20	6.26	6.22	6.20		Basin pHs
839.3	839.3	839.2	1256.2	3795.1	

39.	Soluble BOD5 (influent, mg/L)
0.	Conc. Influent active mass (mg/L)
50.0	Conc. Influent biologically inert mass (mg/L)
49.2	Conc. Influent ammonia (mg/L)
13.0	Conc. Influent non-volatile solids (mg/L)
0.	Conc. Influent stored mass (mg/L)
49.0	Conc. Influent particulate BOD5 (mg/L)
200.	Conc. Influent alkalinity (as CaCO3, mg/L)
0.5	DO (influent, mg/L)
0.00	Leak parameter
1.	Number of Basins
6.8	pH (influent)
0.03802	Flow rate Q (mgd)
0.	Percent flow to Stage 1 (contact/reaeration)
100.	Percent flow to Stage 2 (contact/reaeration)
0.	Percent flow to Stage 3 (contact/reaeration)
0.	Percent flow to Stage 4 (contact/reaeration)
0.50	Recycle Rate (fraction of input flow rate)
1.0	SRT (set point, days)
19.5	Temperature (deg C)
.02115	Oxygen feed in tons per day
.97	Fractional oxygen purity
1.	Input type (1=constant, 2= sinusoidal, 3= actual Randall's data)
20.	Percent sinusoidal variation in flow input (input type = 2)
20.	Percent sinusoidal variation in Particluate BOD input (input type=2)
20.	Percent sinusoidal variation in Soluble BOD input (input type = 2)

```
1.0         alpha1  (ratio of process to clean water kla's for stage 1)
1.0         alpha2  (ratio of process to clean water kla's for stage 2)
1.0         alpha3  (ratio of process to clean water kla's for stage 3)
1.0         alpha4  (ratio of process to clean water kla's for stage 4)
0.012       bci (active mass decay coefficient)
0.99        beta (ratio of process to clean water c sats)
0.405       BODU to BOD5 ratio
0.015       bsstor(specific rate for conversion of sol sub to stored mass)
0.500       bstor (specific rate for conversion of part sub to stored mass)
0.60        fcstorm  (maximum fraction that can be stored mass)
0.05        kcstor (stored substrate fraction, dimensionless)
1.          kla1 (kla for stage 1, 1/hour) **************************
1.          kla2 (kla for stage 2, 1/hour)
1.          kla3 (kla for stage 3, 1/hour)
1.          kla4 (kla for stage 4, 1/hour)
0.5         Lower limit for kla1  **************************
0.5         Lower limit for kla2 (upper and lower limts on kla in 1/hour)
0.5         Lower limit for kla3
0.5         Lower limit for kla4
10.         Upper limit for kla1  **************************
8.0         Upper limit for kla2
6.          Upper limit for kla3
6.          Upper limit for kla4
6.0         DO set point for Stage 1 ******************************
6.0         DO set point for Stage 2 (mg/L)
6.0         DO set point for Stage 3
6.0         DO set point for Stage 4
2.0         Proportional gain for DO control (set to zero for no control)
0.2         Reset (integral) gain for DO control (set to zero for no control)
0.65        O2 purity in stage 4 setpoint (mole fraction)
0.0         Proportional gain for stage 4 purity control (1.0)
0.0         Rest (integral) gain for stage 4 purity control (1.0)
1.42        koex (o2 uptake from endogenous respiration)
1.10        ko2sol (o2 uptake from soluble substrate synthesis)
1.10        ko2str (o2 uptake from stored substrate synthesis)
2.0         kso2  (do half saturation coefficient, mg/L)
0.006       usol  (maximum growth rate on soluble substrate 1/hr )
0.75        ustor (maximum growth rate on stored substrate 1/hr )
1.2         y1co21 (co2 pro'd per unit soluble substrate metabolized)
1.2         y1co22 (co2 pro'd per unit particulate substrate metabolized)
0.4         y1sol  (active mass yield from soluble substrate)
0.4         y1stor (active mass yield from stored substrate)
0.15        y2     (biologically inert mass yield from active mass decay)
0.1239      ynh31  (ammonia consumed by active mass)
50.2        Clarifier area (ft^2)
8.7         Clarifier depth (ft)
5.          Number of elements in the clarifier (10 max)
1.          SRT/FM definitiion (2 uses clarifier sludge mass, 1 ignores it)
10.         VG1 (gas, ft^3)
10.         VG2 (gas, ft^3)
10.         VG3 (gas, ft^3)
10.         VG4 (gas, ft^3)
66.8        VL1 (liquid, ft^3)
66.8        VL2 (liquid, ft^3)
66.8        VL3 (liquid, ft^3)
66.8        VL4 (liquid, ft^3)
```

```
4         method (1 = Euler, 2 = Modified Euler, 3 = RKS, 4 = RKS variable)
0.0010    delt (integration time step)
5.00      prdel (print interval)
125.      outdel (plot interval)
125.      fintim (length of simulation)
0.001     abserr (absolute integration error, RKS variable only)
0.001     relerr (relative integration error, RKS variable only)
```

APPENDIX 4 - INPUT FILES FOR THE WESTPOINT PLANT

```
13
0.         1.1        time= 12 midnight ; this file has format 2f10.0
2.         1.02       time= 2 AM        ; the time should appear in the
4.         0.90       time= 4 AM        ; first 10 columns with a decimal
6.         0.78       time= 6 AM        ; point. In the second 10 columns
8.         0.71       time= 8 AM        ; the normalized flow or bod should
10.        0.83       time=10 AM        ; appear with a decimal point. For
12.         1.00      time=12 Noon      ; example, "2.        1.1" means that
14.         1.1       time= 2 PM        ; the flow or bod was 110% of the mean
16.         1.12      time= 4 PM        ; at 2 AM.  The number of points is
18.         1.13      time= 6 PM        ; specified by an integer that must
20.         1.15      time= 8 PM        ; appear on the first line of the file
22.         1.10      time=10 PM        ; and on the first line preceeding the
24.         1.1       time=12 midnight  ; bod data.  This integer tells the
14                                      ; the program how many data pairs to
0.         1.00                         ; to read.  The time can be entered in
1.5        1.06                         ; any arbitrary spacing. It must be
3.5        1.11                         ; in ascending order.  The time spacing
5.5        0.90                         ; for flow and bod do not have to match.
7.5        1.06                         ; blank lines are not permitted.
9.5        0.94
11.5       1.06              Particulate BOD
13.5       1.18
15.5       1.02
17.5       0.94
19.5       0.92
21.5       0.85
23.5       0.94
24.0       0.97
14
0.         1.20              Soluble     BOD
1.5        1.23
3.5        1.58
5.5        1.43
7.5        1.14
9.5        0.45
11.5       0.39
13.5       0.39
15.5       0.94
17.5       1.04
19.5       1.17
21.5       1.06
23.5       1.17
24.0       1.19
```

.0	.4	30.1	12.3	5.0						Soluble Substrate
.0	18.8	46.6	68.6	76.3						Stored Mass
.0	743.6	259.8	266.7	271.4						Active Mass
.0	1164.9	421.9	422.2	422.3						Biologically Inert Mass
.0	296.0	107.4	107.4	107.4						Non-volatile Mass
.0	2.5	46.0	22.7	8.2						Stored Substrate
.0	45.0	46.3	45.4	44.8						Ammonia Concentration
.0	5.9	5.9	6.0	6.0						Dissolved Oxygen
.05	.07	.08	.08							Carbon Dioxide
.90	.81	.76	.72							Oxygen Purity
6.21	6.27	6.23	6.20							Basin pHs
565.6	567.6	563.2	565.5	560.0	561.1	556.7	557.8	554.4	2437.0	Clarifier Solids

43.	Soluble BOD5 (influent, mg/L)
0.	Conc. Influent active mass (mg/L)
50.0	Conc. Influent biologically inert mass (mg/L)
49.2	Conc. Influent ammonia (mg/L)
13.0	Conc. Influent non-volatile solids (mg/L)
0.	Conc. Influent stored mass (mg/L)
53.0	Conc. Influent particulate BOD5 (mg/L)
200.	Conc. Influent alkalinity (as CaCO3, mg/L)
0.	DO (influent, mg/L)
0.00	Leak parameter
6.	Number of Basins
6.8	pH (influent)
143.3	Flow rate Q (mgd)
0.	Percent flow to Stage 1 (step feed)
100.	Percent flow to Stage 2 (step feed)
0.	Percent flow to Stage 3 (step feed)
0.	Percent flow to Stage 4 (step feed)
0.50	Recycle Rate (fraction of input flow rate)
1.0	SRT (set point, days)
15.0	Temperature (deg C)
69.1	Oxygen feed in tons per day
.97	Fractional oxygen purity
3.	Input type (1=constant, 2= sinusoidal, 3= actual Randall's data)
20.	Percent sinusoidal variation in flow input (input type = 2)
20.	Percent sinusoidal variation in Particluate BOD input (input type=2)
20.	Percent sinusoidal variation in Soluble BOD input (input type = 2)

```
1.0        alpha1  (ratio of process to clean water kla's for stage 1)
1.0        alpha2  (ratio of process to clean water kla's for stage 2)
1.0        alpha3  (ratio of process to clean water kla's for stage 3)
1.0        alpha4  (ratio of process to clean water kla's for stage 4)
0.012      bci (active mass decay coefficient)
0.99       beta (ratio of process to clean water c sats)
0.405      BODU to BOD5 ratio
0.015      bsstor(specific rate for conversion of sol sub to stored mass)
0.500      bstor (specific rate for conversion of part sub to stored mass)
0.60       fcstorm  (maximum fraction that can be stored mass)
0.05       kcstor (stored substrate fraction)
1.         kla1 (kla for stage 1, 1/hour) **************************
5.         kla2 (kla for stage 2, 1/hour)
5.         kla3 (kla for stage 3, 1/hour)
3.         kla4 (kla for stage 4, 1/hour)
0.5        Lower limit for kla1  **************************
0.5        Lower limit for kla2 (upper and lower limits on kla in 1/hour)
0.5        Lower limit for kla3
0.5        Lower limit for kla4
10.        Upper limit for kla1  **************************
8.0        Upper limit for kla2
6.         Upper limit for kla3
6.         Upper limit for kla4
6.0        DO set point for Stage 1 ****************************
6.0        DO set point for Stage 2
6.0        DO set point for Stage 3
6.0        DO set point for Stage 4
2.0        Proportional gain for DO control (set to zero for no control)
0.2        Reset (integral) gain for DO control (set to zero for no control)
0.65       O2 purity in stage 4 setpoint
0.0        Proportional gain for stage 4 purity control (1.0)
0.0        Rest (integral) gain for stage 4 purity control (1.0)
1.42       koex (o2 uptake from endogenous respiration)
1.10       ko2sol (o2 uptake from soluble substrate synthesis)
1.10       ko2str (o2 uptake from stored substrate synthesis)
2.0        kso2  (do half saturation coefficient, mg/L)
0.006      usol  (maximum growth rate on soluble substrate, 1/hr)
0.75       ustor (maximum growth rate on stored substrate, 1/hr)
1.2        y1co21 (co2 pro'd per unit soluble substrate metabolized)
1.2        y1co22 (co2 pro'd per unit particulate substrate metabolized)
.4         y1sol  (active mass yield from soluble substrate)
.4         y1stor (active mass yield from stored substrate)
0.15       y2     (biologically inert mass yield from active mass decay)
0.1239     ynh31  (ammonia consumed by active mass)
238000.    Clarifier area (ft^2)
16.        Clarifier depth (ft)
10.        Number of layers in the claifier
1.         SRT/FM definitiion (2 uses clarifier sludge mass, 1 ignores it)
12544.    VG1 (gas, ft^3)
12544.    VG2 (gas, ft^3)
12544.    VG3 (gas, ft^3)
12544.    VG4 (gas, ft^3)
78400.    VL1 (liquid, ft^3)
78400.    VL2 (liquid, ft^3)
78400.    VL3 (liquid, ft^3)
78400.    VL4 (liquid, ft^3)
```

```
4         method
0.0010    delt
1.        prdel
75.       outdel
96.0      fintim
0.001     abserr
0.001     relerr
```

APPENDIX 5 - SHORT INSTRUCTIONS FOR SETTING UP AND RUNNING THE MODEL

The model comes on a single master disk and is contained in the file MAIN.EXE. Copy this file to a working disk which can be either a floppy disk or a hard disk. The input files are contained in two directories on the master disk. One is set up for the pilot plant and the other is set up for the full scale plant. They are contained in directories "pilot" and "bigplant", respectively. Decide which you wish to simulate and copy the entire contents of the directory to the working disk. There are five files that you should copy: INITS, PARAMS, TIMERS, INPUTS, and DIURNAL.

After copying these files remove the master disk. Print the five input files (MAIN.EXE cannot be printed). Inspect the inputs and decide which ones you want to change. After deciding upon the modifications, edit each file accordingly. It is important that you do not delete lines or change the location of the numbers. Changes will cause the program to fail when reading the input files. It is also important to make sure that your editor saves the files in ASCII format. You may use a word processor to make these modifications, but you must specify the output as ASCII. Also you must not use tabs in the input files.

After performing these modifications you can now run the model. This can be done in two ways. The model can be run "raw" without the Autocad graphics by simply typing "MAIN." The plot output will be written into the file "OUTPUT.DAT." The plot output file will not include the mass balance terms below the clarifier solids outputs.

The model can be run using the Autocad graphics if they are available. Copy the programs written by the Metro staff to the directory containing the MAIN program and the input files. Make sure that the directory containing the Autocad program is in the DOS path. Type

SPLOT and wait. After the program loads you will see an Autocad menu. You may edit the input files, except for diurnal, using EDT or your editor renamed EDT. After performing the editing you must now run the program. The Autocad menu and screen will disappear and the MAIN program will run as it did in the raw case. It may take as long as 15 to 25 minutes for this phase of the program to finish. After it finishes the Autocad menu will return. Observe the instruction in the lower part of the screen to plot the model outputs.

The Autocad menu will allow you to re-edit the input files and rerun the model as many times as you desire.

The model must be run with 10 layers in the clarifier if it is to be used with the Autocad program.