EFFICIENT SOLUTION OF MULTIPLE-MODEL, MULTIPLE-ALGORITHM PROBLEMS IN UNDERGRADUATE AND GRADUATE EDUCATION

Mordechai Shacham, Ben-Gurion University, Beer-Sheva, Israel

Michael B. Cutlip, University of Connecticut, Storrs, CT 06269, USA

Michael Elly, Intel Corp., Qiryat Gat, Israel

Introduction

Mathematical software packages, such as Excel®, Maple™, Mathcad, MATLAB®, Mathematica® and Polymath, are currently utilized for numerical problem solving in engineering education (Cutlip et al., 1998, Shacham and Cutlip, 1999). From the numerical solution perspective, it is convenient to characterize the various problems as Single Model-Single Algorithm (SMSA) problems and complex problems with some combination of Multiple Models and Multiple Algorithms. Typical examples of SMSA type problems include the following:

1. Steady state operation of a tubular reactor where the model consists of a system of ordinary differential equations and explicit algebraic equations. One numerical integration algorithm (such as the 4th order Runge-Kutta) can be used to solve this model.
2. Calculation of the bubble-point temperature for a non-ideal liquid mixture where the model includes a system of implicit and explicit algebraic equations. A non-linear equation solver algorithm (such as the Newton-Raphson technique) can be used to solve this problem .
3. Fitting the Wagner equation to vapor pressure data using a linear regression algorithm.

The application of the mathematical software packages for solving SMSA problems has essentially replaced all other solution techniques, as can be seen in many recent textbooks (see, for example, Fogler, 2005). However, for complex problem types, the solution process is often more involved. The types of models included in the "complex" category are:

1. Multiple Model Single Algorithm (MMSA) Problem.

A typical example of such a problem is the cyclic operation of a semi-batch bioreactor (Cutlip and Shacham, 2007). The three modes of operation of the bioreactor (initialization, processing, and harvesting) are represented by different models comprising of ordinary differential equations and explicit algebraic equations. All models can be solved by one numerical integration algorithm (such as the 4th order Runge-Kutta).

2. Single Model Multiple Algorithm (SMMA) Problem.

An example of such a problem is the problem of parameter estimation in dynamic systems. In this case there is a model comprising of ordinary differential equations and explicit algebraic equations, with parameters that should be fitted to experimental data using nonlinear regression techniques. One option is to solve this system by integrating the differential equations with specified parameter values in an internal loop, and minimizing the sum of squares of the difference between the calculated and the experimental values using an optimization algorithm, in the outer loop .

Additional examples for SMMA problems include: the solution of two point boundary value problems where the integration of the model is carried out in the inside loop and a nonlinear equation solver algorithm adjusts the boundary values in an outer loop, and the solution of differential-algebraic systems of equations where the same algorithms are used but in an opposite hierarchy.

3. Multiple Model Multiple Algorithm (MMMA) problem.

A typical example of such a problem is the optimization of the semi-batch bioreactor, described earlier, with respect with some of its operational parameters. An additional example is the modeling of an exothermic batch reactor, where the two stages of operation (heating and cooling) require different models and different integration algorithms (stiff and non-stiff).

The solution of such complex problems can be rather cumbersome and time consuming even if mathematical software packages are used, as manual transfer of data from one model to another and consecutive manual reruns may be frequently required. However, combining the use of several software packages of various levels of complexity, flexibility and user friendliness, can considerably reduce the time and effort required for solving complex models. Following this premise, the models representing the various stages of the problems are coded and tested using a software package (for example, POLYMATH, a product of Polymath Software, http://www.polymath-software.com) that requires very little technical coding effort. After testing each of the modules separately, they are combined into one program using a programming language, or a mathematical software package that supports programming (for example, MATLAB, a trademark of The Math Works, Inc. http://www.mathworks.com). To minimize the probability of introducing errors into the model equations, an option within POLYMATH for automatic conversion of the programs representing the various modules can be converted to MATLAB functions. MATLAB can then be programmed to handle the consecutive and repetitive calls to the various models, applying the appropriate solution algorithms, and assigning the hierarchy of the computations .

In this paper, the solution of a problem involving the biokinetic modeling of a chemostat with imperfect mixing will be carried out to demonstrate the solution of a SMSA problem, the generation of parametric runs of an SMSA problem, and the solution of a SMMA problem.

Modeling a chemostat with Polymath and solution for one set of parameter values

The chemostat problem is presented by Cutlip and Shacham (2008). The complete details of the problem are provided in Appendix 1. The chemostat may be considered to be modeled as a reactor with a completely-mixed volume V_1 that interacts with another completely-mixed but stagnant volume V_2. This is shown in Figure 13–13 of Appendix 1. Volume V_2 represents the poorly mixed regions within a production fermenter. The microbial system to be modeled involves substrate S going to product P only under the action of cells X. The separate balances on the substrate, cells, and product in each reactor utilize Monod kinetics and a cell death rate constant given by k_d. The four balance equations are shown in Appendix 1. The assignment involves solution of the model equations for one set of parameter values, plotting the substrate (S), cell (X), and product (P) concentrations versus the dilution rate: $D = F_1/V_1$ (where F_1 is the feed flow rate to the well mixed volume: dm^3/hr, and V_1 is the well mixed part of the reactor volume: dm^3). Additionally, the cell production rate, DX_1, and product production rate, DP_1, should be plotted as a function of the dilution rate. Finally the dilution rate which will maximize the production rate for the cells and the dilution rate which will maximize the production rate for the product are to be determined.

The Polymath model for the chemostat, given in Table 1, includes four implicit nonlinear algebraic equations which are obtained from the balance equations. The Polymath model (including the "comments" which start with the # sign) provide complete documentation of the equations, the values of the constants and the initial estimates used for the four unknowns: S_1, S_2, X_1, and X_2. Statements 1 through 4 of Table 1 present the implicit equations for obtaining the substrate concentration in the well mixed volume (S_1) and in the stagnant volume (S_2), and the cell concentration in the well mixed volume (X_1) and in the stagnant volume (X_2). Explicit variables and constants are described in statements 5-18. Initial estimates for the unknowns in the nonlinear equations are provided in lines 19 to 22.

The results for the case where F_1 = 0.17 L/hr are given In Table 2. For this case, the dilution rate D = 0.085 hr^{-1}; the cell production rate DX_1 = 0.0119717 g/hr and the product production rate DP_1 = 0.0066612 g/hr.

Exporting the chemostat model to MATLAB for parametric runs

Parametric runs can be carried out with POLYMATH by manually changing the parameter values. However, carrying out such runs by programming within MATLAB can be more efficient. The MATLAB function representing the operation of the chemostat is generated automatically by Polymath (Table 3). Note that MATLAB requires input of the variable values into the function in a single array (**x**, in this case), and return of the function values in a single array (**fx**, lines 20-23 in Table 3). The variable values are put back into variables with the same names as used in the Polymath model (lines 2-5) to make the MATLAB code more meaningful. POLYMATH orders the equations sequentially as required by MATLAB and converts any needed intrinsic functions and logical expressions.

In order to carry out parametric runs while changing the feed flow rate F_1 this variable has to be added as an input parameter to the *NLEfun* function (Table 3) and a main program has to be prepared which changes the value of F_1, solves the system of nonlinear equations, collects the pertinent data, and plots the results of the parametric runs. Part of this main program is shown in Table 4. The value of F_1, is changed starting at F_1 =0.1 up to F_1 = 0.25 with steps of 0.01. The MATLAB library function *fsolve* is used to solve the system of algebraic equations in line 7 of Table 4. The variable values needed for preparing the various plots are calculated and stored in lines 8 through 10.

A plot of S_1, X_1, and P_1 as function of the dilution rate is shown in Figure 1 and the cell and product production rates are plotted in Figure 2. Observe that there are clear maximum points for the two production rates in the vicinity of D = 0.08 hr^{-1}. Determination of the exact points of the maximum involves the solution of a SMMA problem.

Maximization of the production rates – solving an SMMA problem

The two optimization problems can be defined as minimization problems (as only minimum searching library functions available in MATLAB) as follows:

$$\min_{F_1} - DX_1 \quad \text{and} \quad \min_{F_1} - DP_1 \quad \text{where} \quad D = F_1 / V_1 . \tag{1}$$

The calculation of D, X_1 and P_1 associated with a particular value of F_1 involves the solution of a system of nonlinear algebraic equations (NLEs) while a minimization algorithm is required in order to find the values of F_1 that satisfy Eq.1. Thus this is a single model (the chemostat) and multiple algorithm (one for solution of NLEs and one for minimization).

The MATLAB library function *fminbnd* for single value minimization can be used for finding the minimum of the functions in 1. In order to carry out the minimization two new functions should be prepared. The first one (shown in Table 5) obtains F_1 as input, uses the *fsolve* library function to solve the chemostat model, and returns – DX_1 to the calling function. The second function does the same except that it returns the value of – DP_1. Two calls to the library function *fminbnd* identify the highest production rate for cells DX_1 = 0.01207 g/hr at dilution rate of D = 0.079932 hr^{-1} and the highest production rate for product DP_1 = 0.0066643 g/hr at dilution rate of D = 0.083729 hr^{-1}.

Conclusions

This work demonstrates that the effort involved in solving complex problems which involve multiple models and or multiple algorithms can be considerably reduced by using a combination of software packages. This is achieved by utilization of the most effective software tools for the various tasks. Use of a "user-friendly" mathematical software package such as POLYMATH for the preparation of the individual models reduces the technical effort involved in coding and minimizes the potential errors that may enter the final completed mathematical model. Conversion of the verified separated modules into a more versatile program within a package such as MATLAB requires essentially the addition of a main executive program that controls the interaction with the user, the sequencing and integration of the results of the various modules, and the reporting of the results.

In addition to reducing the time and effort required for building the process models, the proposed approach yields modular programs that are well documented and easily modified. It also ensures that the models presented in the documentation are almost identical to the models used in the program.

The POLYMATH and MATLAB programs used in this study are available at the site: ftp://ftp.bgu.ac.il/shacham/chemostat/.

References

1. Cutlip, M., Hwalek, J.J., Nuttall, H.E., Shacham, M., Brule, J., Widman, J., Han, T., Finlayson, B., Rosen, E. M. and Taylor, R., “A Collection of Ten Numerical Problems in Chemical Engineering Solved by Various Mathematical Software Packages.” *Comput. Appl. Eng. Educ*., 6(3), 169-180(1998).
2. Cutlip, M. B. and Shacham, M. *Problem Solving In Chemical and Biochemical Engineering with Polymath, Excel and MATLAB*. Prentice-Hall, Upper Saddle River, New-Jersey, 2008.
3. Fogler, H. S., *Elements of Chemical Reaction Engineering*, 4th Ed, Prentice-Hall, Upper Saddle River, New-Jersey, 2005.
4. Shacham, M. and M.B. Cutlip, “A Comparison of Six Numerical Software Packages for Educational Use in the Chemical Engineering Curriculum”, Computers in Education Journal, IX(3), 9-15 (1999).

Table 1. Polymath model for the chemostat with imperfect mixing

No.	Equation # Comment
1	f(S1) = F1*S0+F2*S2-(1/Yxs)*(mum*S1/(Ks+S1))*X1*V1-F1*S1-F2*S1 # Substrate balance on the well mixed volume
2	f(S2) = F2*S1-(1/Yxs)*(mum*S2/(Ks+S2))*X2*V2-F2*S2 # Substrate balance on the stagnant volume
3	f(X1)=F2*X2+(mum*S1/(Ks+S1)-kd)*X1*V1-F1*X1-F2*X2 # Cell balance on the well mixed volume
4	f(X2)=F2*X1+(mum*S2/(Ks+S2)-kd)*X2*V2-F2*X2 # Cell balance on the stagnant volume
5	P1=Yps*(S0-S1) # Production (g/dm^3)
6	D=F1/(V1+V2) # Dilution rate (1/hr)
7	S0 = 0.6 # Feed substrate concentration (g/dm^3)
8	kd = 0.002
9	Yxs=0.4 # Yield coefficient (g cells/g substrate)
10	Yps=.2 # Yield coefficeint (g product/g substrate)

11	Ks = 0.2 # Monod constant (g substrate/L)
12	mum = 0.2 #Maximal specific growth rate (1/hr)
13	V1=1.7 # Well mixed voume (dm^3)
14	V2=0.3 # Stagnant volume (dm^3)
15	F1=.17 # Feed flow rate to the well mixed volume (dm^3/hr)
16	F2=0.2*F1 # Feed flow rate to the stagnant volume (dm^3/hr)
17	PR_DX1=D*X1 # Cell production rate (g/hr)
18	PR_DP1=D*P1 # Product production rate (g/hr)
19	S1(0) = 0
20	S2(0)=0
21	X1(0)=0.2
22	X2(0)=0.4

Table 2. Chemostat results for F_1 = 0.17 L/hr

Variable	Value	f(x)	Initial Guess
S_1 (g/L)	0.208163	4.32E-13	0
S_2 (g/L)	0.044739	6.83E-14	0
X_1 (g/L)	0.140844	-1.73E-13	0.2
X_2 (g/L)	0.202638	-2.74E-14	0.4
D (1/hr)	0.085		
F_1 (L/hr)	0.17		
F_2 (L/hr)	0.034		
PR_DP1 (g/hr)	0.006661		
PR_DX1 (g/hr)	0.011972		

Table 3. MATLAB function (model) for the chemostat with imperfect mixing

No.	Equation % Comment
1	function fx = MNLEfun(x);
2	S1 = x(1);
3	S2 = x(2);
4	X1 = x(3);
5	X2 = x(4);
6	Yps = .2; %Yield coefficeint (g product/g substrate)
7	F1 = .17; %Feed flow rate to the well mixed volume (dm^3/hr)
8	S0 = .6; %Feed substrate concentration (g/dm^3)
9	kd = .002;
10	Yxs = .4; %Yield coefficient (g cells/g substrate)
11	P1 = Yps * (S0 - S1); %Production (g/dm^3)
12	Ks = .2; %Monod constant (g substrate/L)
13	mum = .2; %Maximal specific growth rate (1/hr)
14	V1 = 1.7; %Well mixed volume (dm^3)
15	V2 = .3; %Stagnant volume (dm^3)
16	D = F1 / (V1 + V2); %Dilution rate (1/hr)
17	F2 = .2 * F1; %Feed flow rate to the stagnant volume (dm^3/hr)
18	PR_DX1 = D * X1; %Cell production rate (g/hr)
19	PR_DP1 = D * P1; %Product production rate (g/hr)
20	fx(1,1) = F1 * S0 + F2 * S2 - (1 / Yxs * mum * S1 / (Ks + S1) * X1 * V1) - (F1 * S1) - (F2 * S1);
21	fx(2,1) = F2 * S1 - (1 / Yxs * mum * S2 / (Ks + S2) * X2 * V2) - (F2 * S2);

22	fx(3,1) = F2 * X2 + (mum * S1 / (Ks + S1) - kd) * X1 * V1 - (F1 * X1) - (F2 * X2);
23	fx(4,1) = F2 * X1 + (mum * S2 / (Ks + S2) - kd) * X2 * V2 - (F2 * X2);

Table 4. Part of the MATLAB "main program" for parametric studies with the chemostat

No.	Equation % Comment
1	options = optimset('Diagnostics',['off'],'TolFun',[1e-9],'TolX',[1e-9]);
2	Yps = 0.2; S0 = 0.6; kd = 0.002; Yxs = 0.4; Ks = 0.2;
3	mum = 0.2; V1 = 1.7; V2 = 0.3;
4	F1=0.1;
5	xguess = [0 0 0.2 0.4]; % initial guess vector
6	for k=1:16
7	xsolv=fsolve(@MNLEfun,xguess,options,F1);
8	S1(k)=xsolv(1); S2(k)=xsolv(2); X1(k)=xsolv(3); X2(k)=xsolv(4);
9	F1list(k)=F1; D(k) = F1 / (V1 + V2); P1(k)= Yps * (S0 - S1(k));
10	PR_DX1(k) = D(k) * X1(k); PR_DP1(k) = D(k) * P1(k);
11	F1=F1+0.01;
12	end

Table 5. A function for calculating the cell production rate for a single value of F_1

No.	Equation % Comment
1	function PR_DX=ProdRateCell(F1)
2	V1 = 1.7;
3	V2 = 0.3;
4	xguess = [0 0 0.2 0.4]; % initial guess vector
5	options = optimset('Diagnostics',['off'],'TolFun',[1e-9],'TolX',[1e-9]);
6	xsolv=fsolve(@MNLEfun,xguess,options,F1);
7	X1=xsolv(3);

8	D = F1 / (V1 + V2);
	PR_DX = -D* X1;

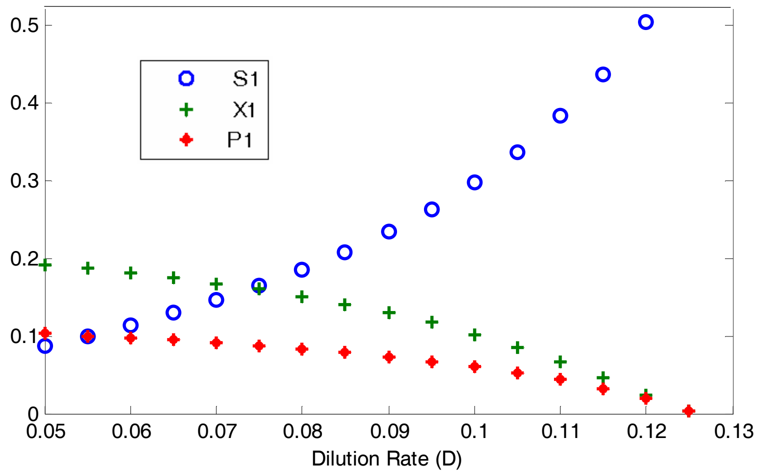

Figure 1. Plot of S_1, X_1 and P_1 as function of dilution rate

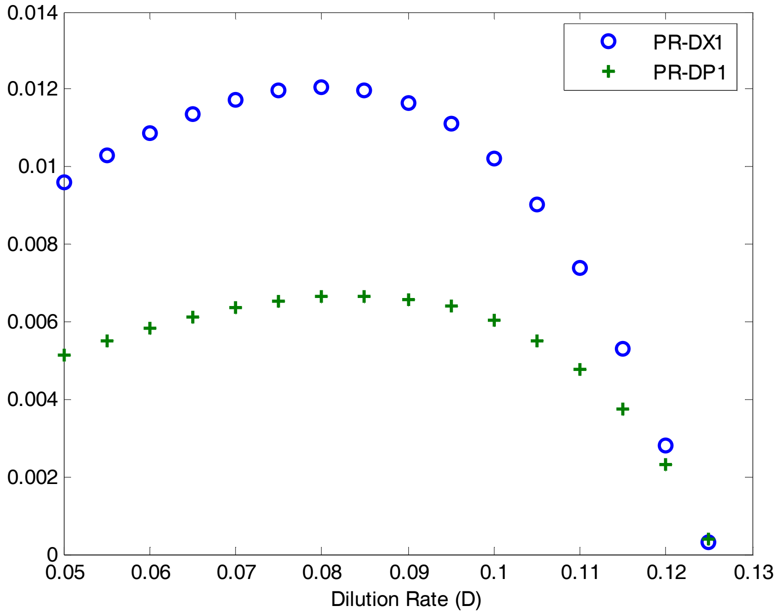

Figure 2. Cell production rate (PR-DX1) and product production rate (PR-DP1) as function of dilution rate

Appendix 1 - Problem Statement for Biokinetic Modeling Imperfect Mixing in a Chemostat

A chemostat is usually considered to be a completely mixed reactor; however, this is not always the case. Consider the situation where the chemostat may be considered to be modeled as a reactor with a completely-mixed volume V_1 that interacts with another completely-mixed but stagnant volume V_2 as shown in Figure A-1. Volume V_2 may be considered to model the poorly mixed regions within a production fermenter.

The microbial system to be modeled involves substrate S going to product P only under the action of cells X. The following separate balances on the substrate, cells, and product in each reactor utilize Monod kinetics and a cell death rate constant given by k_d.

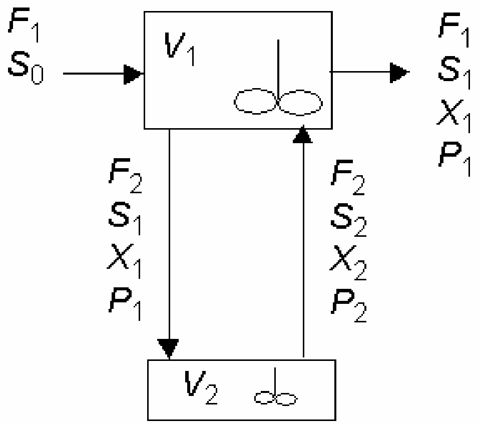

Figure A-1 Chemostat Reactor Model

Steady-State Substrate Balance on Volume V_1

$$F_1 S_0 + F_2 S_2 + \frac{1}{Y_{X/S}} \left(\frac{\mu_m S_1}{K_S + S_1} \right) X_1 V_1 \;=\; F_1 S_1 + F_2 S_1 \qquad \textbf{(A-1)}$$

Steady-State Substrate Balance on Volume V_2

$$F_2S_1 + \frac{1}{Y_{X/S}}\left(\frac{\mu_m S_2}{K_S + S_2}\right)X_2V_2 = F_2S_2 \tag{A-2}$$

Steady-State Cell Balance on Volume V_1

$$F_2X_2 + \left(\frac{\mu_m S_1}{K_S + S_1} - k_d\right)X_1V_1 = F_1X_1 + F_2X_2 \tag{A-3}$$

Steady-State Cell Balance on Volume V_2

$$F_2X_1 + \left(\frac{\mu_m S_2}{K_S + S_2} - k_d\right)X_2V_2 = F_2X_2 \tag{A-4}$$

Overall Steady-State Material Balance for Product

$$P_1 = Y_{P/S}(S_0 - S_1) \tag{A-5}$$

Microbial growth has been studied in a continuous culture and the following parameters were obtained: μ_m = 0.2 h^{-1}, K_S=0.2 g/dm^3, k_d = 0.002 h^{-1}, $Y_{X/S}$ = 0.4 g cells/g substrate, and $Y_{P/S}$ = 0.2 g product/g substrate. Tracer studies have indicated that the incomplete mixing can be described by a well-mixed volume V_1 = 1.7 dm^3 and a stagnant volume of V_2 = 0.3 dm^3 with a constant flow rate relationship of F_2 = 0.2 F_1 in dm^3/h. Chemostat operation is such that X_0 = 0 and S_0 = 0.6 g/dm^3, and the endogenous metabolism can be neglected.

(a) Create a single graph of *S*, *X*, and *P* versus the dilution rate defined by $D = F_1/V_1$.

(b) Plot the cell production rate, the product D^*X_1, and the product production rate, the product of D^*P_1, as a function of the dilution rate between 0.05 and 0.130 h^{-1}.

(c) Estimate the dilution rate that will maximize the production rate, DX_1, for the cells and the dilution rate that will maximize the production rate, DP_1, for the product.