A Simultaneous-Modular Approach to Process Flowsheeting and Optimization

Part III: Performance on Optimization Problems

The performance of the simultaneous-modular approach on four process optimization problems is studied, and numerical experiments using different computational strategies are also performed. The simultaneous-modular approach is found to be very effective on process optimization problems, and the strategies used in our implementation SIMMOD are found to be more efficient than those used in previous implementations of the simultaneous-modular approach for process optimization.

HERN-SHANN CHEN
and M. A. STADTHERR
Chemical Engineering Department
University of Illinois
Urbana, IL 61801

SCOPE

This is the third in a series of papers in which we present a comprehensive study and critical evaluation of the simultaneous-modular approach for process flowsheeting and optimization. In Part I of this series, we discussed the theory underlying the simultaneous-modular approach, and described our implementation of SIMMOD. In Part II, we used SIMMOD to study the performance of the simultaneous-modular approach on several process simulation problems, and presented the results of numerical experiments comparing different computational strategies.

In this paper we use SIMMOD to evaluate the performance of the simultaneous-modular approach and the computational strategies for implementing it on process optimization problems. These strategies have been discussed in more detail previously (Part I of this series). The specific questions to be addressed here are:

1. In a recent study, Jirapongphan (1980) showed that if the stream connection equations are converged before performing optimization calculations, the simultaneous-modular approach may converge very slowly. We examine this anomaly in more detail.

2. The line search technique used in the Han-Powell method to force convergence from poor initial guesses may sometimes cause the algorithm to converge very slowly (Powell, 1980). It is interesting to see whether this kind of phenomenon occurs in chemical process optimization problems and whether the basic watchdog technique (Chamberlain et al., 1979) is effective in alleviating the problem.

3. In most process optimization problems, many choices of design variables are possible. It is interesting to see if the efficiency of the Han-Powell method is greatly affected by the choice of design variables.

4. It is well known that scaling has a very strong effect on the performance of most optimization algorithms. The results presented by Chen and Stadtherr (1983) on some standard mathematical test problems also indicate that scaling has significant effects on the performance of the Han-Powell method. Thus, we would like to study here the effectiveness of the simple scaling scheme used in SIMMOD (Part I).

5. The use of approximate physical property models to generate the flowsheet-level Jacobian has been shown to be effective on simulation problems (Part II). The Jacobian evaluation time is significantly reduced and there is virtually no effect on the convergence of the simultaneous-modular approach. Since a Jacobian and a gradient evaluation are required at every iteration of the Han-Powell method, the potential saving of CPU time is even greater for process optimization problems. However, in the Han-Powell method this derivative information will be used to update a Hessian approximation. This is significantly different than the process simulation case, in which the Jacobian is updated in NEQLU using function values. Because of this, it is almost certain that the superlinear rate of convergence of the Han-Powell method will be lost. However, if the Jacobian and gradient approximations are good enough, a reasonable rate of convergence can still be expected.

Four benchmark problems are used in this numerical study. These problems have all been studied by others so that some comparisons can be made. Problems 1 and 2 are the two- and three-flash-unit systems studied by Jirapongphan (1980) using a somewhat different formulation of the simultaneous-modular approach. Problem 3 is a gasoline polymerization process

Hern-Shann Chen is now at Union Carbide Corporation, South Charleston, WV 25303
Correspondence concerning this paper should be addressed to M. A. Stadtherr.

studied by Friedman and Pinder (1972) and Gaines and Gaddy (1976), using the sequential-modular approach. Problem 4 is an ammonia synthesis process studied recently by Parker and Hughes (1981) using a quadratic approximation programming (QAP) approach, and by Biegler and Hughes (1981) and Jirapongphan (1980) using the simultaneous-modular approach.

CONCLUSIONS AND SIGNIFICANCE

Results of studies performed on four benchmark problems show that the simultaneous-modular approach is very effective for process optimization, especially when using our implementation SIMMOD. In fact, in terms of overall computational efficiency, the strategies used in SIMMOD are found to provide roughly an order of magnitude improvement over other versions of the simultaneous-modular approach.

Numerical experiments using SIMMOD show that: 1. A number of sequential-modular iterations should be performed in order to initialize the problem. 2. The basic watchdog technique is very effective. 3. A good scaling procedure can improve the efficiency and reliability of the optimization algorithm. 4. In some situations the choice of design variables can have a significant effect on overall efficiency. 5. The use of simple thermodynamic models to approximate derivatives appears to be attractive.

Other studies have indicated the potential of the simultaneous-modular approach for solving process simulation and optimization problems. We believe that in the techniques used by SIMMOD we have realized a considerable portion of this potential.

INTRODUCTORY REMARKS

Throughout this study the following options are used in SIMMOD unless specified otherwise:

1. All thermodynamic properties are calculated using the Peng-Robinson (1976) equation of state with all binary parameters set equal to zero.
2. Tear sets are generated automatically by the method described in Part I.
3. Before starting the simultaneous-modular iterations, we set all tear stream variables equal to zero, ignore free variables and design specifications, and perform a number of sequential-modular iterations to initialize tear stream variables. The exact number of sequential iterations used is given below.
4. The Jacobian and gradient are generated by full-block perturbation, as described in Part I.
5. Scaling is performed by setting the initial Hessian approximation to be a nonidentity diagonal matrix as described in Part I.
6. The relative convergence tolerance for the module calculations is 10^{-6}, the perturbation factor in the Jacobian and gradient evaluations is 10^{-3}, and the relative convergence tolerance for the flowsheet-level optimization problem is 10^{-3}. The convergence test used for the optimization problem is considered in more detail below.
7. All runs using SIMMOD were done on a CDC Cyber 175.

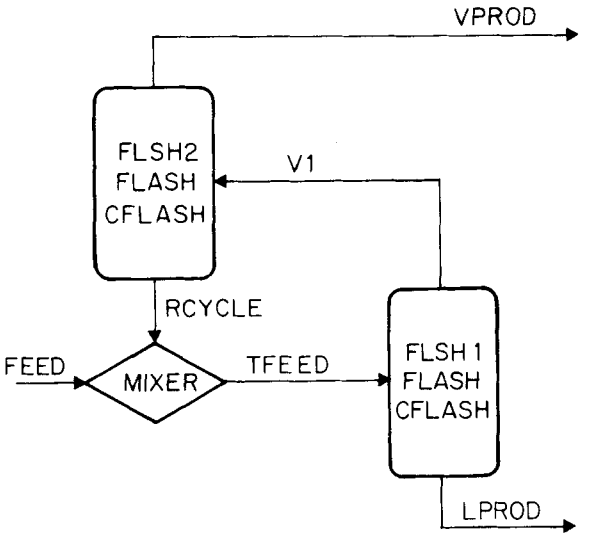

Figure 1. Block diagram for two-flash-unit process (Problem 1). In each block the user-assigned name appears first, followed by the SIMMOD module name, and then the SIMMOD cost module name if appropriate. User-assigned stream names are also shown.

PROBLEM 1: OPTIMIZATION OF A TWO-FLASH-UNIT SYSTEM

This is a hypothetical process studied by Jirapongphan (1980).

TABLE 1. SPECIFICATIONS FOR THE TWO FLASH-UNIT PROBLEM

Maximize: product value − utility cost		
Product values		
VPROD: 15.17 \$/kgmol *n*-Butane		
LPROD: 12.79 \$/kgmol *n*-Hexane		
Utility costs		
Cooling: 0.00018964 \$/MJ cooling		
Heating: 0.0026636 \$/MJ heating		
Design variables (lower bound, starting value, upper bound)		
Flash pressure in FLSH1. (0.7095, 0.7095, 0.8936) MPa		
Flash temperature in FLSH1. (299.8, 385.9, 432.0) K		
Flash temperature in FLSHT. (299.8, 372.0, 432.0) K		
Stream FEED:	Temperature	121.1 C
	Pressure	1.72 MPa
	Component flow rates:	
	n-Butane	0.0378 kgmol/s
	n-Pentane	0.0504 kgmol/s
	n-Hexane	0.0378 kgmol/s
MIXER	Mixer	
FLSH1	Isothermal flash	
	Flash temperature is a design variable	
	Flash pressure is a design variable	
FLSH2	Isothermal flash	
	Flash temperature is a design variable	
	Pressure drop = 0.0	

TABLE 2. SAMPLE INPUT FILE FOR TWO FLASH-UNIT PROBLEM

```
TITLE "TWO FLASH UNITS PROCESS OPTIMIZATION"
RUN-OPTIONS: SAVE-RESULTS, JACOPT = FULL;
C
C ***** OBJECTIVE FUNCTION *****
C
MAX 1.0 * SALE – 1.0 * UTILITY
COMPONENTS: N-BUTANE, N-PENTANE, N-HEXANE
PROPERTY: PENG-ROBINSON
C
C     THIS IS THE MODULE DESCRIPTION SECTION
C
MODEL MIXER BY MIXER: IN = RCYCLE, FEED; OUT = TFEED;
MODEL FLSH1 BY FLASH: IN = TFEED; OUT = V1, LPROD;
C
C ***** 3 = FLASH T, 5 = FLASH P *****
C
  PARA = 1.0, 0.0, 385.9, 0.0, 0.7095;
  FREE = 3(0.0, 299.8, 432.0), 5(0.0, 0.7095, 0.8936)
  CPAR = 12*0.0, 0.0026636, 0.00018964
MODEL FLSH2 BY FLASH: IN = V1; OUT = VPROD, RCYCLE:
  PARA = 1.0, 0.0, 372.0, 1.0, 0.0;
  FREE = 3(0.0, 299.8, 432.0)
  CPAR = 12*0.0, 0.0026636, 0.00018964
C
C     THIS IS THE STREAM DESCRIPTION SECTION
C
C
C ***** VALUE FOR FEED IS SPECIFIED BY VALUE = *****
STREAM FEED: TYPE = INPUT
  T = 250 F, P = 250 PSIA, VALUE = 0.0 "$/KGMOLE"
  CFLOW = 300, 400, 300 "LBMOLE/HR"
C ***** VALUE OF VPROD AND LPROD ARE SPECIFIED IN UVALUE *****
STREAM VPROD: TYPE = OUTPUT
STREAM LPROD: TYPE = OUTPUT
C
C     OTHER INFORMATION
C
CONV-CONTROL: DERROR = 1.0E-4, SEQUENTIAL = 3, SIMULTANEOUS = 100,
              ACCM = 1.0E-10;
PRINT-CONTROL: FREQUENCY=100;
END
```

TABLE 3. PERFORMANCE OF SIMMOD ON TWO FLASH-UNIT PROBLEM I

	Run Number					
	1	2	3	4	5	6
No. Seq. Iter. to Initialize	1	1	1	3	3	3
Line Search Obj.	Penalty	Watch	Watch	Penalty	Watch	Watch
Scaling	No	No	Yes	No	No	Yes
No. of Simul. Iterations	63	25	12	30	25	11
Total No. Line Searches	151	45	16	70	30	14
Initialization Time, s	0.015	0.013	0.012	0.038	0.039	0.041
Function Eval. Time, s	2.644	0.759	0.266	1.154	0.492	0.233
Derivative Eval. Time, s	6.452	2.493	1.192	2.999	1.499	1.107
SQPHP Overhead Time, s	1.552	0.600	0.287	0.709	0.369	0.266
Equiv. No. of Seq. Iter.	609	229	106	297	147	100
Final Obj. Fun. Value in M$/yr	23.149	23.150	23.150	23.140	23.145	23.150

Tear set {RCYCLE} was chosen by SIMMOD.
Design variables are flash temperatures and pressure.

It can be regarded as a distillation column with only two ideal stages but with a heat load at each stage. The block diagram of this problem is shown in Figure 1 and specifications for this problem are given in Table 1. The objective function used is the gross profit of the process, (product value) − (utility cost). The flash temperatures of FLSH1 and FLSH2 and the common pressure of FLSH1 and FLSH2 are chosen as design variables.

As discussed in more detail previously (Chen and Stadtherr, 1985a), Jirapongphan uses what we refer to as formulation I, in which all connecting streams are torn. For this problem, using Jirapongphan's approach requires the solution of a nonlinear programming (NLP) problem involving 26 variables and 23 equality constraints. Since Jirapongphan follows an infeasible path approach, each line search in the Han-Powell method for solving the NLP problem requires only one sequential-modular iteration. As noted above, the quadratic/linear approximation (Q/LAP) method of Beigler and Hughes (1981) is in one sense a sort of sequential-modular approach. On the other hand it is also quite similar in principle to the method of Jirapongphan, differing primarily in the fact that Q/LAP follows a feasible path approach, using the equality contraints to eliminate most of the variables before entering the NLP routine. Thus, for this problem, Q/LAP would require the solution of an NLP problem with only three variables and no equality constraints. Because a feasible path is followed however, each line search now requires a complete simulation. SIMMOD, along with the IPOSEQ, RFV, and CFV methods of Biegler and Hughes (1982a, 1983) can be classified as using what we call formulation III, SIMMOD and IPOSEQ following an infeasible path, and RFV and CFV a feasible path. Using SIMMOD for this problem, the NLP problem that must be solved has 8 variables and 5 equality constraints, and each line search requires only one sequential-modular iteration.

In Jirapongphan's study, the tolerance for the flash calculations and the perturbation factor are set at 10^{-10} and 10^{-5} respectively. For the purpose of comparison, we also adopt these values for problems 1 and 2. The relative convergence tolerance used on the flowsheet level for these two optimization problems is 10^{-4}, i.e., the objective function is optimized to about four significant digits.

A sample input file for this problem is shown in Table 2. Note that this input file is essentially the same as that for a process simulation problem. The only differences are: the objective function is specified through the MAX/MIN statement; product or feed stream values are specified through the VALUE statement or through a user supplied function subprogram UVALUE; cost parameters are specified through the CPAR statements; inequality specifications are allowed in the SPEC statements; and the number of free variables can be different from the number of design specifications.

The performance of the simultaneous-modular approach on this problem with different combinations of options is shown in Table 3. Some of the entries in this table can be interpreted as follows:

1. In the usual implementation of the Han-Powell method, a single penalty function is used as the objective in the line search. As mentioned above, and discussed in more detail shortly, on some problems this approach may lead to a very slow rate of convergence. Many techniques (Mayne, 1980; Fletcher, 1982; Yamashita, 1982; Schittkowski, 1981; Chamberlain et al., 1979, 1982) have been proposed for overcoming this problem. In SQPHP (Chen and Stadtherr, 1983), the NLP routine used in SIMMOD, we use the basic watchdog technique of Chamberlain et al. (1979). In this method the primary line search objective is still the penalty function, but a secondary line search function, namely the Lagrangian, is added and used if necessary to maintain a good convergence rate. A generalized version of this technique has been described more recently (Chamberlain et al., 1982). In order to study the effect of using the basic watchdog technique we disable it for some runs. The third row in Table 3 indicates whether the penalty function alone is used as the line search objective, or whether the watchdog technique is used.

2. In order to study the effect of scaling the objective function and the variables, the scaling procedure used in SIMMOD is disabled on some runs, as noted in the fourth row of Table 3.

3. The number of function evaluations performed, not counting those used for derivative evaluation, is the same as the number of line searches performed. Using SIMMOD, each function evaluation requires one sequential-modular iteration. Thus the average time per sequential-modular iteration for run 1 is about 2.664/151 = 0.0175. Using this figure we can express the derivative evaluation time and the overhead time for the NLP routine SQPHP in terms of an equivalent number of sequential-modular iterations. For instance, for run 1 the total equivalent number of sequential-modular iterations is 1+151+(6.452+1.552)/0.0175

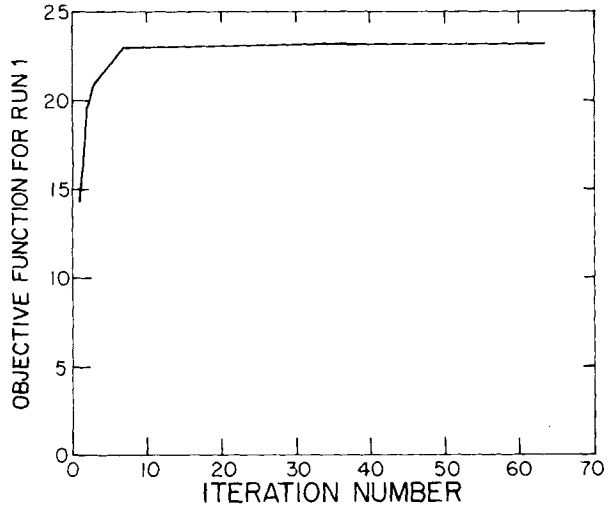

Figure 2. Progression of objective function vs. iteration number for Problem 1, run 1.

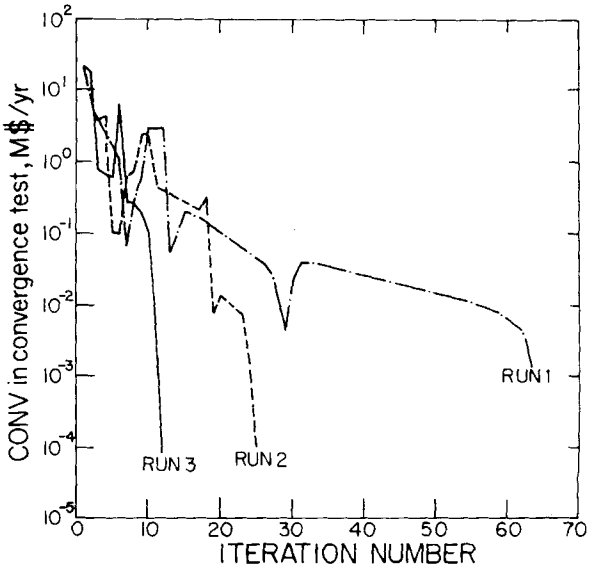

Figure 3. Progression of convergence function for Problem 1, runs 1-3.

TABLE 4. PERFORMANCE OF SIMMOD ON TWO FLASH-UNIT PROBLEM II

	Run Number					
	1	2	3	4	5	6
No. Seq. Iter. to Initialize	1	1	1	3	3	3
Line Search Obj.	Penalty	Watch	Watch	Penalty	Watch	Watch
Scaling	No	No	Yes	No	No	Yes
No. of Simul. Iterations	31	14	11	46	15	14
Total No. Line Searches	64	21	14	92	21	21
Initialization Time, s	0.016	0.017	0.017	0.050	0.050	0.049
Function Eval. Time, s	1.250	0.404	0.270	1.789	0.405	0.404
Derivative Eval. Time, s	3.449	1.553	1.221	5.094	1.659	1.530
SQPHP Overhead Time, s	0.736	0.334	0.263	1.086	0.360	0.335
Equiv. No. of Seq. Iter.	279	120	92	414	129	121
Final Obj. Fun. Value in M$/yr	23.150	23.150	23.150	23.150	23.150	23.150

Tear set {RCYCLE} was chosen by SIMMOD.
Design variables are flash vapor fractions and pressure.

= 609. It should be noted that the initialization time is not necessarily a good measure of the average time per sequential-modular iteration. These initial iterations may require somewhat less time than average because the presence of null streams may cause some units to be bypassed, or they may require somewhat more time because the intialization is poor and some modules do not converge. (See Problem 4 in Part II.)

Before proceeding to discuss the results shown in Table 3, we consider the results of run 1 in more detail. The progression of the objective function for this run is shown in Figure 2. It is clear from this figure that most of the improvements, from 14.26 to 22.98 M$/yr, are made in the first 8 iterations, while only very small improvements, from 22.98 to 23.15, are made in the following 55 iterations. A detailed analysis of the result shows that this slow convergence is caused by the line search technique used, i.e., although good correction steps are determined in the quadratic programming subproblems, full correction steps are not being accepted because of the line search procedure.

In the convergence test we use *CONV*, a weighted sum of constraint violations and possible objective function improvement. Convergence is declared when $CONV < ACC * \max[|F(x)|, 1.0]$, where *ACC* is the relative convergence tolerance and $F(x)$ is the objective function value. The progression of *CONV* for run 1 is shown in Figure 3. The curve fluctuates greatly, and indicates that the solution may be accepted in many fewer iterations if a looser convergence tolerance is used. For example, the solution would be accepted in 8 iterations if the convergence tolerance is set at 0.003, and in 30 iterations if the convergence tolerance is set at 0.0002. The fluctuation of the curve also indicates that any slight change in implementation, e.g., different thermodynamic property models, may greatly affect the number of iterations required to solve the problem.

Regarding the results in Table 3, the following comments are in order:

1. At the solution, flash temperatures are 387.7 K (FLSH1) and 374.3 K (FLSH2), flash pressure is at the lower bound, 0.7095 MPa, and the objective function is 23.150 M$/yr (a year is assumed to be 330 operating days). These agree well with Jirapongphan's results.

2. Using the one-sequential-iteration initialization, Jirapongphan solved this problem in 21 iterations and 36 line searches. Using the converged-simulation initialization, he solved this problem in 40 iterations and 83 line searches. Based on this result, he concluded that the one-sequential-iteration initialization is more effective. Since he does not consider scaling and he always uses the usual penalty function as the line search objective function, the options he used are similar to those used in runs 1 and 4. The results for run 4 agree quite well with his converged-simulation initialization case, but 63 iterations are required in our run 1 as opposed to his 21 iterations. Although the difference here may be attributed to the different formulations used, i.e., formulation I vs. formulation III, we suspect that the difference is due to the different thermodynamic models used and to sensitivity of the problem, as discussed above.

3. Because the poor performance of the NLP routine on this problem was caused by the line search function used, it is interesting to see whether the basic watchdog technique (Chamberlain et al., 1979) is effective. The results for runs 2 and 5, as compared to runs 1 and 4 show a dramatic reduction in the number of iterations and line searches. The watchdog technique appears to be very effective.

4. For this problem the number of iterations is reduced by about 40% because of the variable scaling and objective function scaling procedure used in SIMMOD. (Constraint scaling is not necessary in connection with the Han-Powell method.)

5. Comparing runs 1–3 to runs 4–6, the three-sequential-iteration initialization seems to be more effective than the one-sequential-iteration initialization. This is contrary to the results of Jirapongphan. The only observation we can make is that this problem is very sensitive to changes in many parameters, and thus it is difficult to make any definite conclusions.

6. When the full capabilities of SIMMOD are utilized (runs 3 and 6), the optimization can be performed very efficiently. Using the one-sequential-iteration initialization, Jirapongphan solves this problem in about 13.3 s on an IBM 370/168. Using SIMMOD requires about 1.7 s on a CDC Cyber 175, which is roughly the same as the IBM 370/168 in terms of speed.

It is interesting to see whether the problem of slow convergence in runs 1 and 4 can be avoided by using a different set of design variables. For the problem just considered, the fraction of

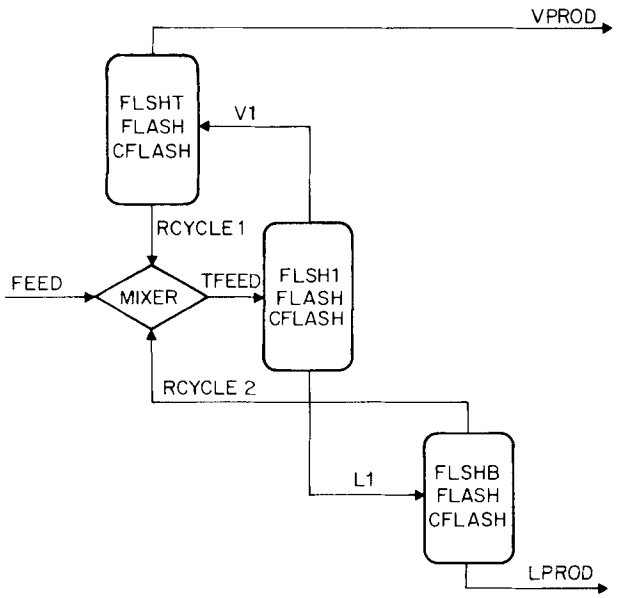

Figure 4. Block diagram for three-flash-unit process (Problem 2).

input vaporized instead of the temperature in each flash unit can be chosen as design variables. This choice of design variables has the advantage that the conditions in the flash units will always remain in the two-phase region, and that the design variables have well-defined upper and lower bounds. The results of optimizing the process with vapor fractions and pressure chosen as the design variables are shown in Table 4. These comments regarding the results are in order:

1. Some characteristics of the problem seem to remain unchanged. In particular, the slow convergence of runs 1 and 4 relative to runs 2–3 and 5–6 is still observed.
2. At least for the situation in which just one sequential iteration is used for initialization, the problem can be solved more efficiently using this set of design variables.
3. Comparing the results of runs 1–3 to those of runs 4–6, the one-sequential-iteration initialization is slightly more effective here.
4. The poor performance of runs 1 and 4 is again caused by the line search technique used. Again, the basic watchdog technique is effective in alleviating the problem.
5. Scaling again improves the efficiency, but the improvement is quite moderate.

PROBLEM 2: OPTIMIZATION OF A THREE-FLASH-UNIT SYSTEM

This is another hypothetical process studied by Jirapongphan (1980). It can be regarded as a distillation column with three ideal stages. The block diagram for this problem is shown in Figure 4, and the problem specifications are given in Table 5. The objective function used is the gross profit of the process, (product value) − (utility cost).

Using Jirapongphan's approach, the NLP problem has 37 variables and 33 constraints. Using Q/LAP, the NLP problem has only 4 variables and no constraints, but each line search requires one complete simulation as opposed to one sequential-modular iteration. Using SIMMOD's formulation III approach, the NLP problem has 9 variables and 5 constraints. It should be mentioned that the size of the NLP problem in Q/LAP and in the formulation III approach remains about the same as in the previous example, while the size of the NLP problem in Jirapongphan's approach increases by almost 50%.

TABLE 5. SPECIFICATIONS FOR THE THREE FLASH-UNIT PROBLEM

Maximize: product value − utility cost	
Product values	
VPROD: 15.17 \$/kgmol *n*-Butane	
LPROD: 12.79 \$/kgmol *n*-Hexane	
Utility costs	
Cooling: 0.00018964 \$/MJ cooling	
Heating: 0.0026636 \$/MJ heating	
Design variables (lower bound, starting value, upper bound)	
Flash pressure in FLSH1. (0.7095, 0.7095, 0.8936) MPa	
Flash temperature in FLSH1. (300.0, 380.0, 432) K	
Flash temperature in FLSHB. (300.0, 388.0, 432) K	
Flash temperature in FLSHT. (300.0, 372.0, 432) K	
Stream FEED: Temperature 121.1 C	
Pressure 1.72 MPa	
Component flow rates:	
n-Butane 0.0378 kgmol/s	
n-Pentane 0.0504 kgmol/s	
n-Hexane 0.0378 kgmol/s	
MIXER	Mixer
FLSH1	Isothermal flash
	Flash temperature is a design variable
	Flash pressure is a design variable
FLSH2	Isothermal flash
	Flash temperature is a design variable
	Pressure drop = 0.0
FLSHT	Isothermal flash
	Flash temperature is a design variable
	Pressure drop = 0.0

The performance of the simultaneous-modular approach on this problem with two different choices of design variables is given in Table 6. These comments are in order:

1. At the solution, flash temperatures are 381.7 K (FLSH1), 393.5 K (FLSHB), and 369.0 K (FLSHT), flash pressure is at the lower bound, 0.7095 MPa, and the objective function is 25.772 M\$/yr. These agree well with the results of Jirapongphan.
2. The slow convergence of runs 1 and 4 is caused by the use of the penalty function in the line search, just as in the previous problem.
3. Again the choice of vapor fractions and pressure as design variables somewhat improves the overall efficiency.
4. The watchdog technique is again quite effective. Although run 5 terminated abnormally, the solution found is close to the optimal solution.
5. Scaling increases the number of iterations in run 3, but it gives a smooth convergence in run 6.
6. Using the one-sequential-iteration initialization, and the temperatures and pressure as design variables, Jirapongphan solves this problem in about 108 s; SIMMOD requires about 5.7 s.

As a final experiment with Problems 1 and 2, we consider the feasibility of using simple thermodynamic models in the module calculations to approximate the derivatives required in the NLP routine. As discussed previously (Chen and Stadtherr, 1985b) the use of simple models for derivative evaluations is a relatively common suggestion that proves very effective on process simulation problems. However, as noted above, in the optimization case the use of approximate derivatives could have a much more serious effect on the rate of convergence.

Because the temperature and pressure of each flash unit will change from iteration to iteration, the effects of temperature and pressure should be included in the simple physical property

TABLE 6. PERFORMANCE OF SIMMOD ON THREE FLASH-UNIT PROBLEM

	Run Number					
	1	2	3	4	5	6
Design Variables	T & P	T & P	T & P	VF & P	VF & P	VF & P
Line Search Obj.	Penalty	Watch	Watch	Penalty	Watch	Watch
Scaling	No	No	Yes	No	No	Yes
No. of Simul. Iterations	42	21	25	50*	15**	16
Total Number of Line Searches	91	37	53	122	30	22
Initialization Time, in Seconds	0.042	0.045	0.042	0.054	0.053	0.055
Function Eval. Time, in Seconds	2.237	0.899	1.237	3.408	0.838	0.616
Derivative Eval. Time, in Seconds	6.391	3.174	3.806	8.107	2.431	2.593
SQPHP Overhead Time, in Seconds	1.117	0.558	0.673	1.319	0.405	0.434
Equiv. No. of Seq. Iter.	395	194	248	462	140	133
Final Obj. Fun. Value in M$/yr	25.750	25.765	25.771	25.755	25.700	25.772

Tear set {TFEED} was chosen by SIMMOD.
3 sequential iteration initialization was used.
* Program terminated because maximum number of iterations was reached, but good solution was still found.
** Program terminated because 5 line searches failed to make improvement, but good solution was still found.

models. In this study, we adopt the following simple models suggested by Boston (1980):

$$\ln(K_i) = A_i + B_i/T + C_i \ln(P)$$
$$HDEP^V = A^V + B^V T + C^V P$$
$$HDEP^L = A^L + B^L T + C^L P$$

TABLE 7. EFFECT OF USING APPROXIMATE DERIVATIVES ON THE OPTIMIZATION OF PROBLEMS 1 AND 2

	Run Number			
	1	2	3	4
Problem Number	1	1	2	2
Simple Thermo. Models Used?	No	Yes	No	Yes
No. of Simul. Iterations	14	14	16	18*
Total No. Line Searches	21	21	22	31
Initialization Time, seconds	0.067	0.067	0.072	0.069
Function Eval. Time, seconds	0.348	0.348	0.522	0.735
Derivative Eval. Time, seconds	1.326	0.719	2.232	1.383
SQPHP Overhead Time, seconds	0.339	0.336	0.429	0.490
Equiv. No. of Seq. Iter.	124	88	137	113
Final Obj. Fun. Value in M$/yr	23.149	23.148	25.772	25.770

3 sequential iteration initialization was used.
Scaling was performed by setting initial Hessian.
Watchdog technique was used in line search.
Design variables are vapor fraction and pressure.
* Program terminated because 5 line searches failed to make improvement, but good solution was still found.

where the *HDEP*'s are enthalpy departures per unit mass, and the constants A, B and C are generated by evaluating physical properties at different pressures and temperatures.

The performance of the simultaneous-modular approach using approximate derivatives is shown in Table 7. These comments are in order:

1. Using approximate derivatives in Problem 1, the number of iterations is not affected. Furthermore, runs 1 and 2 follow almost identical paths.

2. Using approximate derivatives in Problem 2, the program terminates because 5 line searches failed to make improvement. However, the final solution found is very close to the optimal solution. Moreover, if the relative convergence tolerance was set at 0.0002, then run 4 would also converge in 16 iterations.

3. These results show that the idea of using simple thermodynamic models to approximate derivatives is probably still feasible even in the optimization case. Although the reduction of CPU time for these problems is only moderate, it is expected that a more significant reduction could be achieved when the use of this approach for more complex unit modules is considered.

Before proceeding to the next problem it should be noted that Jirapongphan also considered another approach that he found to be very effective on Problems 1 and 2, as well as on Problem 4. In this approach, rather than use the modules to generate partial derivatives in linear models for the flowsheet level, he uses the modules to generate coefficients in simple nonlinear "engineering" models. It should be noted that this approach could be used with any of the three basic problem formulations discussed in Part I. Also it is worth observing that Jirapongphan finds that this approach requires a relatively large fraction of the overall computation time to be spent in the NLP routine. By using a more efficient NLP routine, such as the routine SQPHP used in SIMMOD, this approach might be made even more effective. The difficulty with this sort of approach however would seem to be the lack of good, simple "engineering" models for the most time-consuming modules. The problems studied by Jirapongphan do

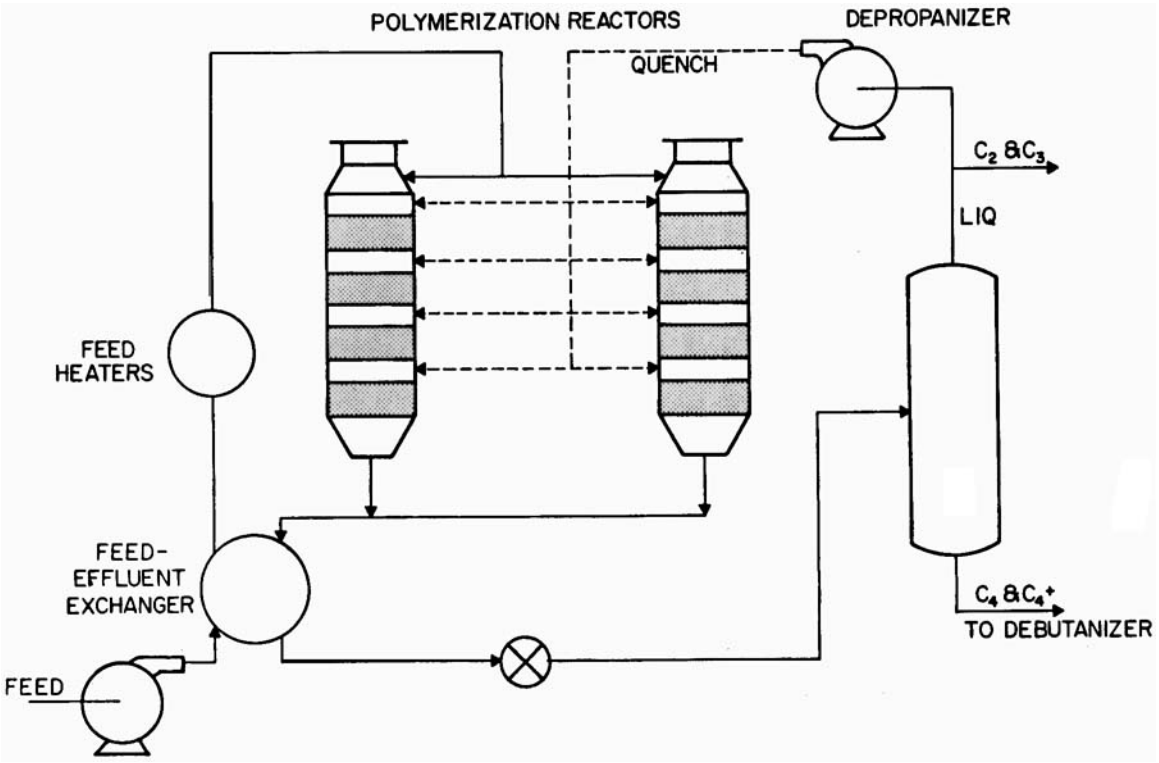

Figure 5. Flow diagram for gasoline polymerization process (Problem 3).

not involve any complicated modules. Also there is no guarantee that the coefficients in the simple engineering models will ever converge.

PROBLEM 3: OPTIMIZATION OF A GASOLINE POLYMERIZATION PROCESS

This is a problem studied by Friedman and Pinder (1972) and Gaines and Gaddy (1976). The flow diagram for this process is shown in Figure 5. A feed consisting of C_2 through C_5 hydrocarbons with 28–36 mol% olefins, enters the unit at about 2.07 MPa and 32.2°C. It is pumped at a pressure of about 3.79 MPa through the feed/effluent exchanger and the feed heater and enters the polymerization reactors at 3.55 MPa and 182.2°C. In each reactor, the dimerization of propene, butenes, and petenes occurs, and the temperature is controlled by introducing liquid distillate from the depropanizer, which serves as a quench between catalyst beds. The reactor effluent then passes through the feed/effluent exchanger and a pressure reduction valve, and enters the depropanizer where propane and lighter components are removed as the top product. The depropanizer bottoms product is then sent to the debutanizer where polymer gasoline is separated.

In Friedman and Pinder's study, the objective functions considered are plant profit, total dimer from the plant, and olefin conversion. Three optimization techniques—the DFP method with classical penalty function, the constrained pattern search method, and a modified complex method—are used with the sequential-modular simulator CHESS to optimize the process. The fractions of liquid distillate recycled to the reactor beds are chosen as the design variables. The condition that the temperatures of reactor beds must be less than 895°R (497 K) provides four inequality design specifications. The results of Friedman and Pinder show that the three optimization techniques considered are about equally efficient and that the CPU time for an optimization study is equivalent to about 50 complete simulations. In the Gaines and Gaddy study, the objective function used is the return on investment. Three optimization techniques, the complex method, the constrained pattern search, and the adaptive random search method, are used with PROPS (Gaddy, 1974), a modified version of CHESS, to optimize the process. Gaines and Gaddy also consider a few design variables and specifications besides those of Friedman and Pinder. The results of Gaines and Gaddy indicate that the adaptive random search method is the most reliable and that using it requires about 200 complete simulations to perform the optimization study.

The model we used for the gasoline polymerization process is shown in Figure 6. Complete specifications for this problem are given by Chen (1982). The complexity of this model is about the same as the model used by Friedman and Pinder. Following Friedman and Pinder, the two parallel reactors are modeled by four mixers and four plug-flow reactors. The three differential equations describing the conversion, energy balance, and pressure drop of the catalytic reactors are integrated using a Runge-Kutta-Fehlberg (4,5) routine, RKF45, written by Watts and Shampine (Forsythe et al, 1977). With a relative tolerance of 10^{-6}, RKF45 requires about 3–4 integration steps and 19–25 derivative evaluations for each plug-flow reactor.

Due to the deactivation of the catalyst, the effectiveness of the reactor beds will decrease with time. The result of running SIMMOD with different times onstream is shown in Table 8. Regarding these results, the following comments are in order:

1. Using the specifications given by Chen (1982), we are not able to reproduce the results of Friedman and Pinder. For example, we can get only about 77% conversion at start-up as opposed to their 92%. Since our objective here is not to optimize any particular process but to evaluate the performance of the process optimization technique, we feel the difference here is not important.

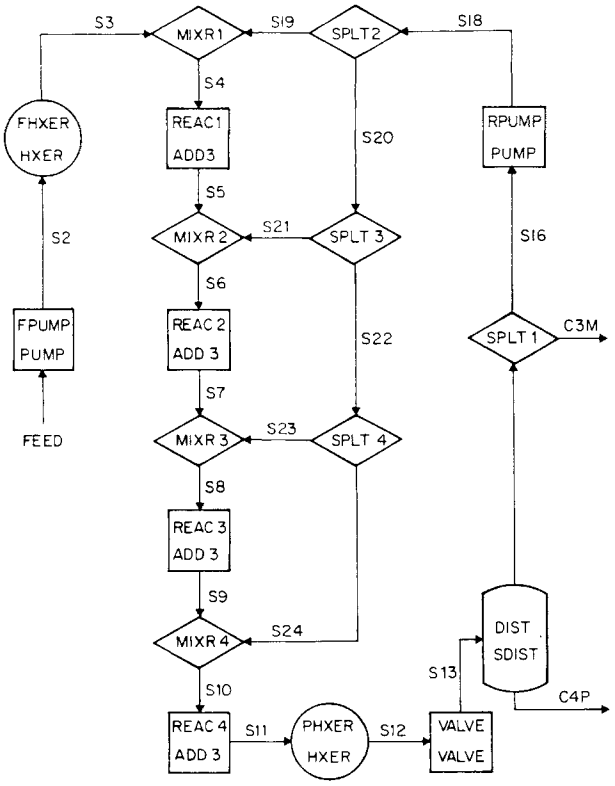

Figure 6. Block diagram for gasoline polymerization process (Problem 3).

2. Because the reaction considered is exothermic, and because the reaction rate increases as the reactor temperature increases, one would expect the quench recycle to each reactor in general to either be zero, or such that the outlet temperature from the reactor is at its upper bound. The solution shown in run 2 does not follow this rule, i.e., the quench recycle to REAC1 is not zero and the outlet temperature of REAC1 is below the upper bound 525 K. This is because the effectiveness factor of REAC1 is zero except in run 1, so there is little distinction between recycling to REAC1 or REAC2.

3. For all the runs studied, the correct active constraints are identified by the NLP routine in the first two iterations. Since the degrees of freedom are then zero, the process optimization problems in effect become controlled simulation problems.

4. For this problem, the simultaneous-modular approach requires about 60 sequential iterations to optimize the process. If we assume conservatively that one complete simulation requires about ten sequential iterations, then optimization by the simultaneous-modular approach requires only about six complete simulations, and is about an order of magnitude faster than the sequential-modular approach.

5. Besides determining an optimal solution, the Han-Powell method also provides some sensitivity information. The sensitivity of the optimal value to changes in the constraints is measured by the Lagrange multipliers, which are available as a result of solving the quadratic programming subproblems in the Han-Powell successive quadratic programming algorithm. For example, to know the effect of changing the maximun allowed temperature of REAC4 on the optimal value, all we need to do is to check the Lagrange multiplier for that constraint. This is an advantage of the Han-Powell method over the optimization methods considered by Friedman and Pinder and by Gaines and Gaddy.

TABLE 8. PERFORMANCE OF SIMMOD ON GASOLINE POLYMERIZATION PROCESS

	Run Number					
	1	2	3	4	5	6
Days on Stream	0	50	100	150	200	250
No. of Simul. Iterations	4	5	5	5	6	6
Initialization Time, s	0.998	0.992	0.951	0.958	0.936	0.886
Function Eval. Time, s	1.338	1.673	1.607	1.610	1.853	1.860
Derivative Eval. Time, s	14.628	18.353	18.099	18.915	21.135	21.122
SQPHP Overhead Time, s	0.201	0.254	0.249	0.245	0.294	0.296
Equiv. No. of Seq. Iter.	51	64	64	67	78	78
Final Obj. Fun. in Mkgmol/yr	0.4096	0.3948	0.3880	0.3798	0.3703	0.3593
Split Frac. of						
SPLT1	0.4029	0.3762	0.3626	0.3458	0.3258	0.3014
SPLT2	0.0	0.0361	0.0	0.0	0.0	0.0
SPLT3	0.3888	0.0	0.0	0.0	0.0	0.0
SPLT4	0.6991	0.7109	0.6971	0.6625	0.6190	0.5622
Outlet Temp. of						
REAC1 (K)	506.80	453.18	454.42	454.42	454.42	454.42
REAC2 (K)	525.00	525.00	523.74	519.84	515.63	511.05
REAC3 (K)	525.00	525.00	525.00	525.00	525.00	524.99
REAC4 (K)	525.00	525.00	525.00	525.00	525.00	525.00

Tear set {S18} was chosen by SIMMOD.
3 sequential iteration initialization was used.
Scaling was performed by setting initial Hessian.
Total number of line searches = number of simul. iterations.

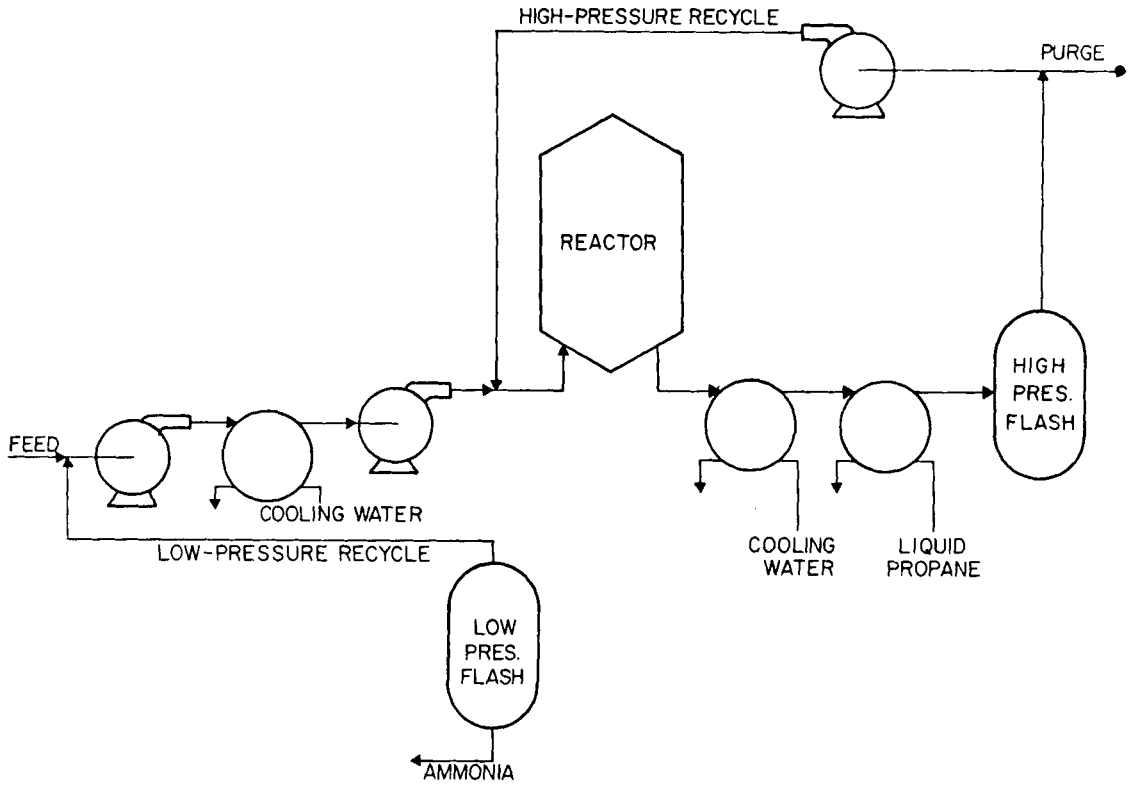

Figure 7. Flow diagram for ammonia synthesis process (Problem 4).

PROBLEM 4: OPTIMIZATION OF AN AMMONIA SYNTHESIS PROCESS

This is a problem studied by Parker and Hughes (1981), Biegler and Hughes (1981) and Jirapongphan (1980). The flow diagram of the process is shown in Figure 7. Feed consisting of H_2, N_2, Ar, and CH_4 enters the unit at about 1 MPa and 27°C. After combining with the low-pressure recycle, the feed is compressed in two stages to around 20–30 MPa. The compressed feed is then mixed with the high-pressure recycle before entering the reactor. The product gas from the reactor is cooled first with cooling water and then by boiling liquid propane at its normal boiling point. The resulting two-phase mixture is then flashed at high pressure to separate the majority of the unreacted synthesis gas from the ammonia product. A small fraction of the overhead is purged to the atmosphere to control the buildup of methane, argon, and excess hydrogen, and the remainder of the overhead becomes the high-pressure recycle stream. The liquid ammonia-rich stream from the flash is flashed again at reduced pressure to remove the remaining dissolved synthesis gas from the ammonia product. The overhead of this flash forms the low-pressure recycle stream.

A special module, ADD2, was written to simulate the catalytic reactor. Equilibrium is assumed for the product gas leaving the reactor. The conversion of the limiting reactant (N_2) is chosen as an independent variable. From the conversion and the inlet composition, the outlet composition and the equilibrium temperature can be determined. The heat duty of the reactor is assumed to be removed by cooling water. Following the study of Parker and Hughes, the following ten variables were chosen as design variables:

1. Outlet pressure from the first stage of feed compressor.
2. Outlet temperature from the compressor intercooler.
3. Outlet pressure from the second stage of the feed compressor.
4. Conversion of nitrogen in the reactor.
5. Outlet temperature from the water cooler.
6. Outlet temperature from the propane cooler.
7. Pressure drop in the high-pressure flash drum.
8. Fraction of the high-pressure overhead purged.
9. Outlet pressure of the high-pressure recycle compressor.
10. Flash pressure in the low-pressure flash drum.

The starting values, lower bounds, and upper bounds for these design variables are given in Table 9. The design specifications for this problem are given in Table 10. The objective function used in this study is the present value of the process with 15% interest, a 50% tax rate, and a 10-year project life.

In our study the process is modeled as shown in Figure 8. Specifications for this problem are given in Table 11. The performance of the simultaneous-modular approach on this problem is shown in Table 12. Several comments are in order:

TABLE 9. DESIGN VARIABLES FOR AMMONIA SYNTHESIS PROBLEM

Design Variable	L.B.*	S.V.*	U.B.*	O.V.*
1. Outlet pres. of FCS1, MPa	4.137	4.413	6.896	4.811
2. Outlet temp. of INTCL, K	310.9	333.1	366.4	310.9
3. Outlet pres. of FCS2, MPa	20.0	20.68	29.72	20.20
4. N_2 conversion in REACTOR	0.35	0.41	0.45	0.45
5. Outlet temp. of WCOOL, K	310.0	320.0	400.0	310.0
6. Outlet temp. of C3COOL, K	233.1	242.0	333.1	246.2
7. Pres. drop in HPFLSH, MPa	0.207	0.47	10.0	0.207
8. Purge fraction in SPLIT	0.05	0.10	0.12	0.065
9. Outlet pres. of RCOMP, MPa	20.0	20.68	29.72	20.20
10. Outlet pres. of LPFLSH, MPa	1.014	1.240	6.895	1.014

* L.B. = Lower bound, S.V. = Starting value, U.B. = Upper bound, O.V. = Optimal value found.

TABLE 10. DESIGN SPECIFICATIONS FOR AMMONIA SYNTHESIS PROBLEM

Constraint	Constraint active at the solution ?
1. Outlet pres. of FCS2 and RCOMP are equal	Yes
2. Reactor temp. < 810.9 K	No
3. Reactor heat duty < 0.0	No
4. Ammonia purged < 0.0003 kgmol/s	Yes
5. Outlet pres. of HPFLSH > 19.65 MPa	Yes
6. Product must be at least 99.5% NH_3	Yes

1. In Parker and Hughes's study the quadratic approximation programming (QAP) approach solves this problem in 9 iterations and 34.88 min of CPU time on an IBM 370/168, which is equivalent to about 70–90 complete simulations. Biegler and Hughes tried the QAP approach on the same problem but with different physical property options, and found that the QAP approach failed to make any significant improvement.

2. Using Q/LAP, Biegler and Hughes solve this problem in 4 iterations and about 14.4 min of CPU time on a CDC 6600 computer, which is equivalent to around 14–15 complete simula-

TABLE 11. SPECIFICATIONS FOR AMMONIA SYNTHESIS PROBLEM

Maximize 1.677 * (product value − raw material cost − utility cost − depreciation)
  Product value
    PURGE: 0.0 $/kgmol;
    PRODUCT: 6.614 $/kgmol
  Raw material value
    FEED: 0.6614 $/kgmol
  Utility cost
    Water cooling: 0.0006 $/MJ
    Propane cooling: 0.004 $/MJ
    Catalyst cost: 20000 $/$m^2$
  Depreciation:
    0.10/yr * investment
Design variables and bounds are shown in Table 9
Constraints are shown in Table 10
Stream FEED: Temperature: 26.67 C
Pressure: 1.013 MPa
Component flow rates in kgmol/s
Hydrogen: 0.1100
Nitrogen: 0.0355
Argon: 0.0009
Methane: 0.0016

MIXER1 Mixer

FCS1 Compressor
Outlet pressure is a design variable

INTCL Water cooler
Temperature of S4 is a design variable
Water inlet temperature = 298 K
Water temperature rise = 10 K

FCS2 Compressor
Outlet pressure is a design variable

MIXER2 Mixer

REACTOR Equilibrium reactor
$N_2 + 3H_2 \rightarrow 2NH_3$
Extent of reaction is a design variable
Heat of reaction = −91.4 MJ/kgmol N_2 @ 298 K
Reactor space time = 5 s
Catalyst life = 5 years
Catalyst cost = 20000 $/$M^2$
Reactor outlet temperature must not exceed 810.9 K
Heat duty must be negative

WCOOL Water cooler
temperature of S7A is a design variable

C3COOL Propane cooler
Temperature of S8 is a design variable
Coolant inlet temperature = 231 K
Coolant temperature rise = 8 K

HPFLSH Adiabatic flash
Pressure drop is a design variable
Pressure must be greater than 19.65 MPa

SPLIT Purge fraction is a design variable
Ammonia purged must not exceed 0.0003 kgmol/s

RCOMP Compressor
Outlet pressure is a design variable
Outlet pressure of RCOMP and FCS2 must be equal

LPFLSH Adiabatic flash
Flash pressure is a design variable
Mole fraction of ammonia in PRODUCT must exceed 0.995

TABLE 12. PERFORMANCE OF SIMMOD ON AMMONIA SYNTHESIS PROBLEM

	Run Number					
	1	2	3	4	5	6
No. Seq. Iter. to Initialize	5	10	15	5	10	15
Scaling	No	No	No	Yes	Yes	Yes
No. of Simul. Iterations	6	5	4	7	5	6
Initialization Time, s	0.433	0.876	1.335	0.436	0.907	1.310
Function Eval. Time, s	0.688	0.550	0.456	0.771	0.560	0.655
Derivative Eval. Time, s	4.680	4.625	3.808	6.414	4.612	5.500
SQPHP Overhead Time, s	0.365	0.345	0.282	0.543	0.398	0.465
Equiv. No. of Seq. Iter.	55	60	55	75	60	76
Final Obj. Fun. Value in M$	13.350	13.537	13.547	13.815	13.814	13.815
Final CONV in M$	0.0055	0.0064	0.0131	0.0020	0.0046	0.0014

Tear set {S8} was chosen by SIMMOD.
Total number of line searches = number of simul. iterations.
CONV is the weighted sum of predicted obj. function improvement and constraint violation used in convergence test.

tions. The number of iterations they used is the same as that of run 3. However, since Q/LAP is a feasible path method, each line search requires a complete controlled simulation instead of just one sequential iteration.

3. Using the formulation I approach of Jirapongphan, the NLP problem to be solved has 105 variables and 92 constraints. Using SIMMOD's formulation III approach, the NLP problem has 17 variables and 13 constraints. Jirapongphan solved this problem with 18 iterations, 29 line searches, and about 5.1 min of CPU time on an IBM 370/168 computer.

4. In our study, 5 bounds on design variables and 4 constraints are found to be active at the solution, so the problem has only one degree of freedom at the solution. This accounts for the relatively few iterations required to solve the problem. The optimal objective function value found is in good agreement with Jirapongphan's result of 6.105 M$, considering that we almost doubled his cost data and assumed a year to be 330 operating days as opposed to the 288 days he used.

5. When the three-sequential-iteration initialization was used the program failed. This seems to indicate that it is better to perform a few more sequential-modular iterations before switching to the simultaneous-modular iterations. The relatively large number of iterations required in Jirapongphan's study also indicates that the one-sequential-iteration initialization scheme he recommended is not suitable for this problem.

6. Comparing runs 1–3 to runs 4–6, the scaling method used by SIMMOD seems to increase the number of iterations slightly. However, a close look at the final objective function indicates that the optimal solutions found without scaling are not optimal. As a further check, we ran the problem again using the solution from run 1 as initial guess. Without scaling, the NLP routine terminated in one iteration and claimed the solution. With scaling, the

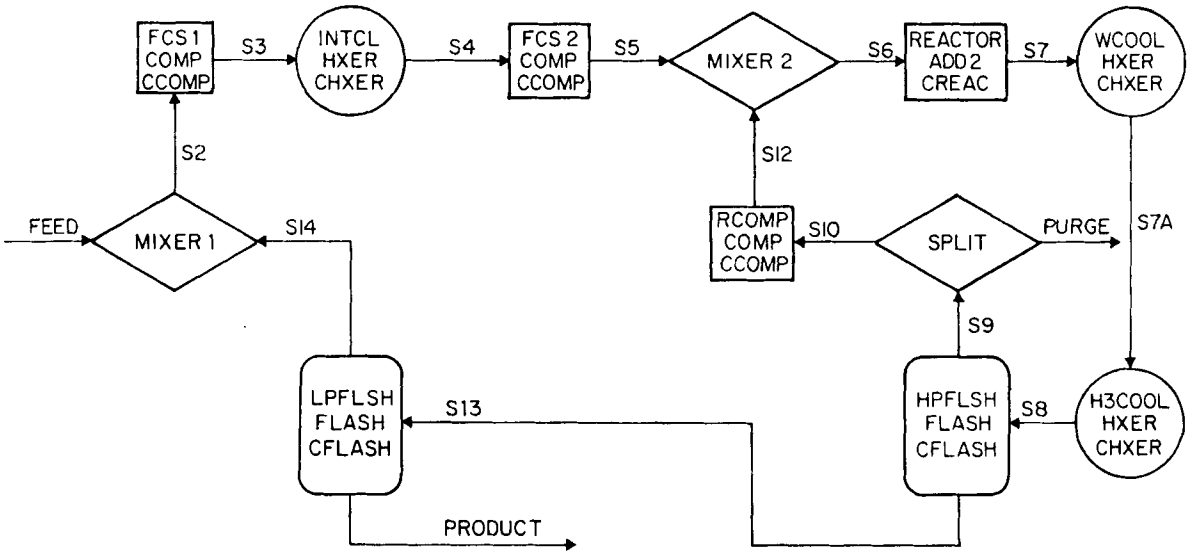

Figure 8. Block diagram for ammonia synthesis process (Problem 4).

NLP routine performed four more iterations and converged to the solution found in runs 4–6. This indicates that the solution found in run 1 is not even a local minimum. Because Biegler and Hughes do not perform scaling and they also use a loose convergence tolerance, it is likely that the solution they found is also a false solution.

7. To compare the CPU times given above, we note that the speed of the CDC Cyber 175 used in this study is about three times that of a CDC 6600, and is about the same as an IBM 370/168. After taking these factors into account, the CPU time used by SIMMOD is still at least an order of magnitude less than in the pervious studies. One reason for the very short execution time on this and the other problems is that the NLP routine SQPHP uses an enhanced version (Chen and Stadtherr, 1983) of the Han-Powell method, and is significantly more efficient than the NLP routines used in other studies. Another factor is that SIMMOD does not perform any unnecessary adiabatic flash calculations on the output streams of modules. Finally it should be noted that in their study Biegler and Hughes operated under the restriction that an existing sequential-modular executive routine must be used in connection with their version of the simultaneous-modular approach. An important point to be made on the basis of the results presented here is that to take full advantage of the potential of the simultaneous-modular approach, one needs to relax this constraint, and either provide a new executive routine designed for the simultaneous-modular approach, or make the appropriate modifications in the existing sequential-modular executive in order to permit the most effective computational strategies to be used.

8. As mentioned by Jirapongphan, a process simulation at optimal conditions using the sequential-modular simulator FLOWTRAN requires 63 sequential iterations. By contrast, SIMMOD requires only about 60 sequential iterations to optimize rather than only simulate the process. It is also worth noting that using SIMMOD, a process simulation at optimal conditions requires only three simultaneous-modular iterations. Again, this demonstrates the effectiveness of the simultaneous-modular approach for process simulation problems, as well as optimization problems.

9. A recent comparison (Biegler and Hughes, 1982b) of the Q/LAP, IPOSEQ, CFV, and RFV methods on a propylene chlorination process indicates that the feasible path methods CFV and RFV are significantly more efficient than the infeasible path method of IPOSEQ. As implemented in SIMMOD, the infeasible path approach appears to be very efficient. It is difficult to imagine that the performance of SIMMOD could be improved by using a feasible path approach. It is possible, however, that on some problems the overall performance may be improved by doing a few more sequential iterations during initialization.

CONCLUDING REMARKS

Based on the results presented above, we can make the following conclusions regarding the performance of the simultaneous-modular approach on process optimization problems:

1. The simultaneous-modular approach is very efficient for process optimization problems. Usually only about 4 to 20 iterations are required to solve a process optimization problem. Especially using SIMMOD, the CPU time required to perform one process optimization is quite small.

2. Although the advantages of performing process optimization studies are well recognized, process optimization has not been performed as often as it should. One of the goals of this study is to demonstrate the feasibility of developing a general-purpose optimization system. With such a tool an average engineer or student, with little or no knowledge about the details of the optimization technique, can easily learn to utilize it to solve process optimization problems. We feel that with the development of the strategies used by SIMMOD, significant progress has been made toward this goal. The use of these techniques, together with a good knowledge of the process in question, should allow the engineer to efficiently perform meaningful process optimization studies.

3. It is better to perform a number of sequential iterations before starting the optimization calculation. This is contrary to the recommendation made by Jirapongphan (1980).

4. The simple scaling scheme used by SIMMOD is found to be effective. It improves the efficiency and reliability of the NLP routine.

5. The stepsize procedure does sometimes cause the Han-Powell method to become very inefficient, but simply using the basic watchdog technique is effective in alleviating this problem.

6. When approximate derivatives are generated by using simple physical property models, the convergence of the simultaneous-modular approach is only slightly affected. This is demonstrated on Problems 1 and 2. However, further studies are needed to see if this is true in general.

7. The simultaneous-modular approach for process optimization seems quite reliable. However it is not as reliable as its counterpart for process simulation and controlled simulation problems. This is because the initial Hessian may be a very poor approximation of the true Hessian.

8. In many applications an engineer may be more interested in the sensitivity of the optimal value rather than the optimal value itself. The sensitivity of the optimal value to changes in constraints is measured by the Lagrange multipliers, which are readily available as a result of solving the quadratic programming subproblems when the Han-Powell method is used.

ACKNOWLEDGMENT

This work has been supported by the National Science Foundation under grant CPE 80-12428.

NOTATION

A_i,B_i,C_i = constants in simple model for K_i
A^L,B^L,C^L = constants in simple model for $HDEP^L$
A^V,B^V,C^V = constants in simple model for $HDEP^V$
ACC = relative convergence tolerance for optimization problems
$CONV$ = weighted sum of the constraint violations and possible objective function improvement
$F(x)$ = objective function for optimization problems
$HDEP^L$ = liquid enthalpy departure per unit mass
$HDEP^V$ = vapor enthalpy departure per unit mass
K_i = equilibrium vaporization factor for component i
P = pressure
T = temperature

LITERATURE CITED

Biegler, L. T., and R. R. Hughes, "Approximation Programming of Chemical Processes with Q/LAP," *Chem. Eng. Prog.*, 77(4), 76 (1981).

———, "Infeasible Path Optimization with Sequential Modular Simulators," *AIChE J.*, 28, 994 (1982a).

———, "Process Optimization: A Comparative Case Study," *Paper No. 23e*, AIChE Ann. Meet., Los Angeles (1982b).

———, "Feasible Path Optimization with Sequential Modular Simulators," *Comput. Chem. Eng.*, in press (1983).

Boston, J. F., "Inside-Out Algorithm for Multicomponent Separation

Process Calculations," *Computer Applications to Chemical Engineering*, R. G. Squires and G. V. Reklaitis, Eds., *ACS Symp. on Mathematical Programming*, Montreal (1979).
Chamberlain, R. M., et al., "The Watchdog Technique for Forcing Convergence in Algorithms for Constrained Optimization," *10th Int. Symp. on Mathematical Programming*, Montreal (1979).
———, "The Watchdog Technique for Forcing Convergence in Algorithms for Constrained Optimization," *Math. Prog. Study*, **16**, 1 (1982).
Chen, H. S., "Computational Strategies for Chemical Process Flowsheeting and Optimization," Ph.D. Thesis, Univ. of Illinois, Urbana (1982).
Chen, H. S., and M. A. Stadtherr, "Enhancements of the Han-Powell Method for Successive Quadratic Programming," *Comput. Chem. Eng.*, in press (1983).
———, "A Simultaneous-Modular Approach to Progress Flowsheeting and Optimization. I: Theory and Implementation," *AIChE J.*, in this issue.
———, "A Simultaneous-Modular Approach to Process Flowsheeting and Optimization. II: Performance on Simulation Problems," *AIChE J.* in this issue.
Fletcher, R., "Second-Order Corrections for Nondifferential Optimization," *Numerical Analysis, Dundee 1981, Lecture Notes in Math.*, **912**, Springer-Verlag, Berlin (1982).
Forsythe, G. E., M. A. Malcolm, and C. B. Moler, *Computer Methods for Mathematical Computations*, Prentice-Hall, Englewood Cliffs, NJ (1977).
Friedman, P., and K. L. Pinder, "Optimization of a Simulation Model of a Chemical Plant," *I&EC Proc. Des. Dev.*, **11**, 512 (1972).
Gaddy, J. L., "Use of Flowsheet Simulation Programs in Teaching Chemical Engineering," *Chem. Eng. Educ.*, **8**, 124 (1974).
Gaines, L. D., and J. L. Gaddy, "Process Optimization by Flowsheet Simulation," *I&EC Proc. Des. Dev.*, **15**, 206 (1976).
Jirapongphan, S., "Simultaneous Modular Convergence Concept in Process Flowsheet Optimization," Ph.D. Thesis, MIT (1980).
Mayne, D. Q., "On the Use of Exact Penalty Functions to Determine Step Length in Optimization Algorithms," *Numerical Analysis, Dundee 1979, Lecture Notes in Math.*, **773**, Springer-Verlag, Berlin (1980).
Parker, A. L., and R. R. Hughes, "Approximation Programming of Chemical Process, Part 1: Optimization of the FLOWTRAN Models, *Comput. Chem. Eng.*, **5**, 123 (1981).
Peng, D. Y., and D. B. Robinson, "A New Two-Constant Equation of State," *I&EC Fund.*, **15**, 59 (1976).
Powell, M. J. D., "Optimization Algorithms in 1979," *Optimization Techniques, Lecture Notes in Control and Information Science*, **22**, Springer-Verlag, Berlin (1980).
Schittkowski, K., "The Nonlinear Programming Method of Wilson, Han, and Powell with an Augmented Lagrangian Type Line Search Function," *Numer. Math.*, **38**, 883 (1981).
Yamashita, H., "A Globally Convergent Constrained Quasi-Newton Method with an Augmented Lagrangian Type Penalty Function," *Math. Prog.*, **23**, 75 (1982).

Manuscript received Apr. 19, 1983; Revision received Sept. 27, 1983, and accepted Jan. 8, 1985.