AgEcon SEARCH
RESEARCH IN AGRICULTURAL & APPLIED ECONOMICS

The World's Largest Open Access Agricultural & Applied Economics Digital Library

This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.

Help ensure our sustainability.

Give to AgEcon Search

AgEcon Search
http://ageconsearch.umn.edu
aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

GIANNINI FOUNDATION OF
AGRICULTURAL ECONOMICS
LIBRARY

FEB 1971

AN ANALYSIS OF SUPPLY RELATIONSHIPS FOR TRADITIONAL SWINE PRODUCERS

by

Arlo J. Minden, Lawrence Smith,

DEPARTMENT OF AGRICULTURAL ECONOMICS

Purdue University

Lafayette, Indiana

AN ANALYSIS OF SUPPLY RELATIONSHIPS FOR TRADITIONAL SWINE PRODUCERS

by

Arlo J. Minden, Lawrence Smith, and Ralph Lattimore

This report presents the results of a first analysis of supply relationships for swine produced with traditional production systems. Traditional systems are interpreted here to refer to those currently used by most swine producers including low capital, range farrowing-open lot finishing and various smaller scale confined farrowing-finishing set-ups with no type of "coordination or integration" scheme. Although no attempt is made here to specify the precise characteristics of systems included in the "traditional" system, work currently underway at Michigan State University is specifically designed for that purpose.

The research reported here draws on previous work at Purdue University and Michigan State University and provides direct input to activity in the U.S.D.A., all of which contribute to analysis of the hog-pork subsector. Data used in this analysis are derived from published series of the Statistical Reporting Service of the U.S. Department of Agriculture for the period 1965-69. However, all errors of analysis and interpretation are those of the authors.

The Problem

The central focus of the research reported here was to estimate the number of barrows and gilts available for slaughter by various weight groups by month. It was necessary to develop estimates of 1) the number

of sows bred and farrowed each month, 2) the price of barrows and gilts, d) the number of gilts saved and bred, 4) the number of sows and boars sold, 5) the number of barrows and gilts available for sale by weight group, and 6) a retention function which characterizes the decision to feed slaughter animals to heavier weights. Each of these estimates is discussed in detail in the following paragraphs.

1) Total Females Bred

The model used as an initial framework for the analysis was:

$$(1)\ TFB_t = f(P^{\bar{B}\bar{G}}_{t-1}, PRC_{t-1}, EP^{\bar{B}\bar{G}}_{t+10})$$

where:

TFB_t = the total number of females bred in month 1

$P^{\bar{B}\bar{G}}_{t-1}$ = price of barrows and gilts lagged one month

PRC_{t-1} = price of corn lagged one month

$EP^{\bar{B}\bar{G}}_{t+10}$ = expected price of barrows and gilts ten months in the future

Data series for TFB were obtained by multiplying the number of sow farrowings in period t+4 by 1.25 for it was assumed that 20 percent of the females bred were not farrowed. Sows farrowed by month were reported by the SRS-USDA. The average dispersion for the 1965-67 period was used to estimate monthly farrowings for 1968 and 1969. A total of 60 observations resulted. The average U.S. corn price and price of hogs was taken from "Agricultural Prices" and "Livestock and Meat", respectively. Prices for barrows and gilts for the eight "interior" markets were also used.

The data series for total females bred was regressed against the national average corn price in t-1, average price of barrows and gilts (com-

bined) in t-1 and the expected price of barrows and gilts in t+10. Expected price in period t+10 was used since pigs from sows bred in month t are normally marketed 10 months later. Dummy variables (0,1) were used to account for seasonality assumed to exist in hog supply relations.

It was expected that fewer sows are bred as the price of corn increases and more are bred if expected price of barrows and gilts is high. The number of females bred in the current month may be indeterminate with respect to barrow and gilt prices the past month t-1. For, if the price of slaughter animals is high farmers may either sell sows to generate additional revenue or retain gilts if they expect high prices to persist.

Simple correlation among the explanatory variables in Equation 1 and between the explanatory variables and total females bred for the 1965-69 period are presented in Table 1. The estimated form of Equation 1 with related t and F statistics is presented in Table 2.

Table 1. Simple Correlation Coefficients for Variables Used in Estimating Equation 1.

	Fall Shift X_1	Summer Shift X_2	Spring Shift X_3	t-1 Corn Price X_4	t-1 Barrows And Gilts Price X_5	t+10 Barrows And Gilts Price X_6	t Females Bred Y_1
Fall	1.00	-.33	-.33	-.09	.18	.15	.15
Summer		1.00	-.33	.22	.25	-.02	-.63
Spring			1.00	.06	-.25	.05	.22
Corn Price				1.00	.25	-.24	-.27
Lagged Barrow Gilt Price					1.00	-.15	-.32
Expected Barrow Gilt Price						1.00	-.229

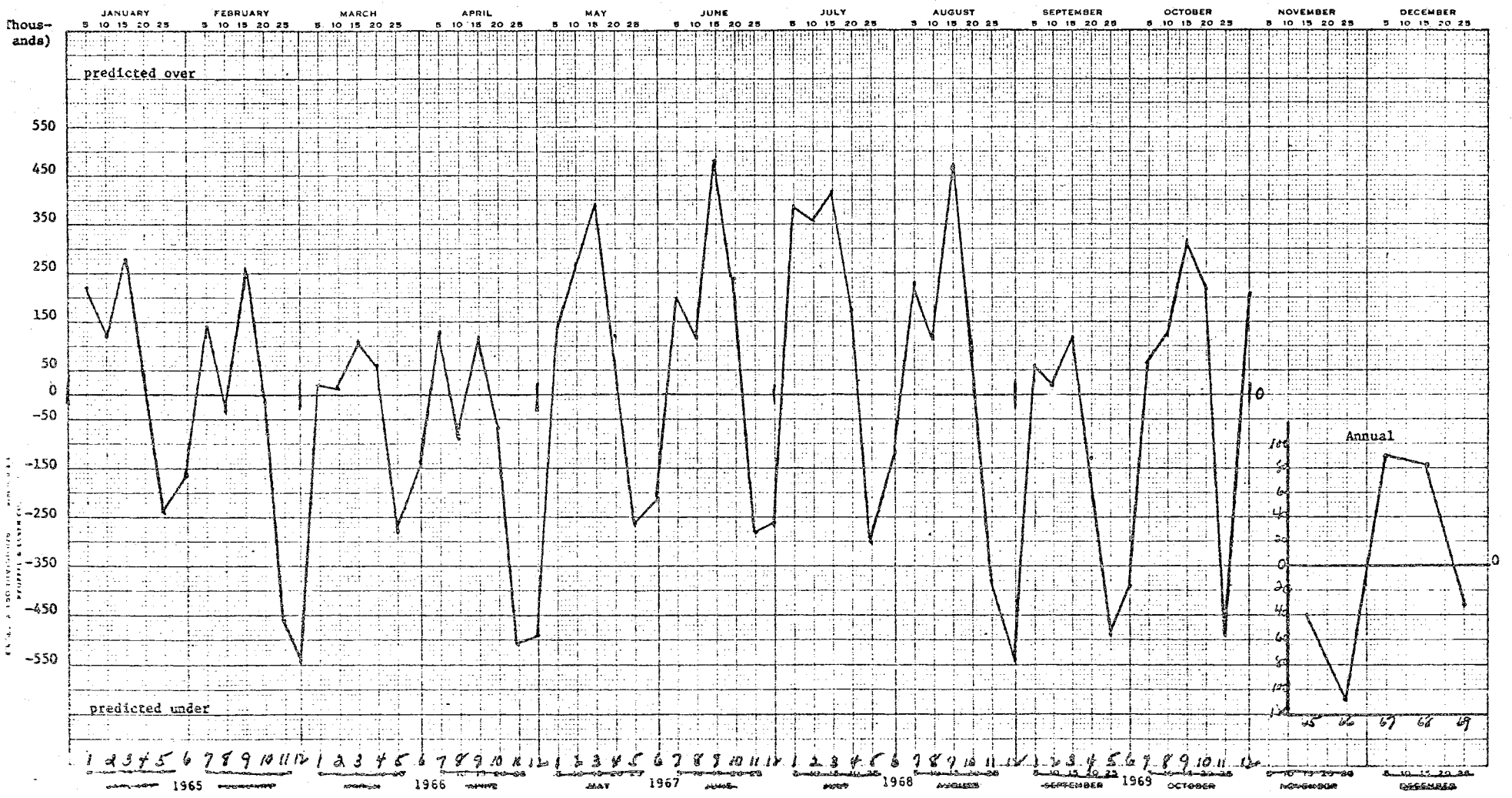
Table 2. Estimated Equation for Total Females Bred and Related Statistics 1965-69.

Variance	= 8.67436184E+04			
Standard Deviation	= 2.94522696E+02			
Correlation Coef.	= 7.33911206E-01			
R Squared	= 5.38625658E-01			
	COEFFICIENTS	STANDARD ERROR	T RATIO	F RATIO
Winter	3.7914551E+03	6.5545960E+02	5.7844222	33.4595401
Fall	7.4142485E+01	1.1441554E+02	.6480106	.4199178
Summer	- 4.6785099E+02	1.1637011E+02	-4.0203708	16.1633814
Spring	5.9374643E+01	1.1119331E+02	.5339768	.2851312
PRCt-1/B6	- 8.3796991E+02	4.3877184E+02	-1.9098079	3.6473661
P_{t-1}/B6	- 2.5727801E+01	1.3238383E+01	-1.9434248	3.7768999
EP_{t+10}	- 4.2701127E+01	1.2693730E+01	-3.3639543	11.3161887

About 53 percent of the variation in TFB was explained by the variables in Equation 1 including the seasonal dummy variables. The results in Table 2 verify that more sows are bred in the fall and spring season relative to winter and less in the summer. The coefficient estimated for corn price was negative as expected. Both pig price variables had negative signs <u>not</u> consistent with expectations. Negative signs on the price variables probably results from the cyclical and/or trend phenomena not appropriately considered in the model or perhaps from incorrect specification of lagged relationships.

Figure 1 portrays the pattern of predicted versus actual numbers of females bred (Y-Y which resulted from estimation of Equation 1. The months of April and September were generally predicted high and May and December low. Trend, not included in the model, is reflected in the annual over and under predictions. Females bred were predicted lower than actual in 1965, 1966, and 1969, while the annual totals for 1967 and 1968 were predicted higher than actual.

FIGURE 1. Monthly Predicted versus Actual Numbers of Females Bred, Based on Estimation of Equation 1.



2) Expected Prices of Barrows and Gilts

Expected prices for barrows and gilts was considered to be a relevant determinant of the hog supply relationships. The conceptual model for expected prices was:

(2) $$EP^{\bar{B}\bar{G}}_{t+10} = f(SE_t, P^{\bar{B}\bar{G}}_{t-1}, 0(EP^{\bar{B}\bar{G}}_{t-1} - EP^{\bar{B}\bar{G}}_{t-2})$$

where:

SE_t = a monthly price correction factor

0 = a coefficient of adjustment considering the differences in ex-expected price lagged one and two months.

There was some concern as to the appropriateness of deriving expected prices from an exponentially smoothed series of prices. Specifically, should expected price be smoothed before being used and should expected prices be used in conjunction with direct observation of actual lagged prices? Myers and Havlicek[1/] found that farmers seemd to project current prices as their future price expectations. Deviations from current prices may occur because of changing production and demand patterns resulting from seasonal and cyclical influences. All exponential smoothing formulas use historical data to forecast price. As such, they contain no economic information pertinent to interpretation of results.

The monthly price correction factor (SE_t) was calculated mechanistically from historical price data. Data and results of these calculations are presented in Table 3. Monthly prices for barrows and gilts were obtained from eight interior markets for each year 1965-69. Annual and seasonal average prices were calculated for each year. December, January,

1/ Myers, Lester H. and Havlicek, Joseph Jr., "Some Theoretical Aspects of Short-Term Hog Supply", Reprinted from Journal of Farm Economics Vol 49, No. 5, December, 1967.

and February are included in the winter season; March, April, and May in the spring; June, July, and August in the summer; and September, October, and November in the fall. Monthly average prices were calculated for the five year period. The values were then considered in relation to the five year annual average price. For example, the five year annual average price was $21.41 and the monthly average for December was $19.87, which yielded a monthly to five year average ratio of .927. Similar calculations could be made for monthly versus average seasonal prices.

Average monthly prices for the five year period and monthly price correction factors are presented in Table 3. Visual comparison of price correction factors obtained from the mechanistic procedure and a regression analysis suggested that the former procedure is at least as accurate as the latter and much easier to obtain. Patterns for seasonal and monthly price correction factors are portrayed in Figures 2 and 3.

3) Gilts Bred

The number of gilts bred for the first time was considered since these animals are removed from the total number of barrows and gilts available for slaughter. The model used to calculate the number of gilts bred was:

$$(3)\ GS_t = TFB_t - k(TFB_{t-6})\ SCR$$

where:

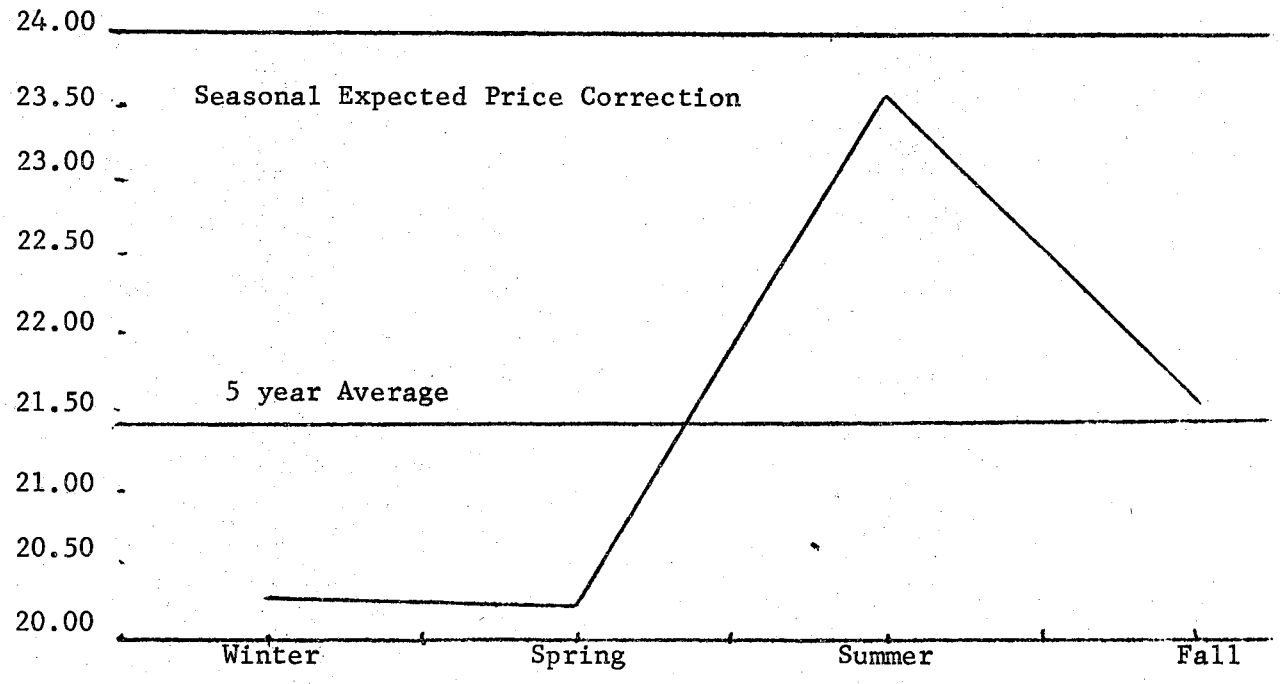
GS_t = gilts bred in month t

K = sow cull factor

SCR = net seasonal correction factor for sow cull versul gilt retention.

The number of sows sold by season and month for each of the years 1965-69 is presented in Table 4. Comparison of seasonal and monthly sow sales data with seasonal and monthly sow farrowings data (Table 5), provides a basis for developing seasonal correction factors for use in adjusting sow

Figure 2.



Dec.	-1.56	March	-1.51	June	1.77	Sept.	.85
Jan.	-1.12	April	-2.05	July	2.46	Oct.	- .05
Feb.	- .63	May	+ .03	Aug.	2.26	Nov.	- .38

Figure 3.

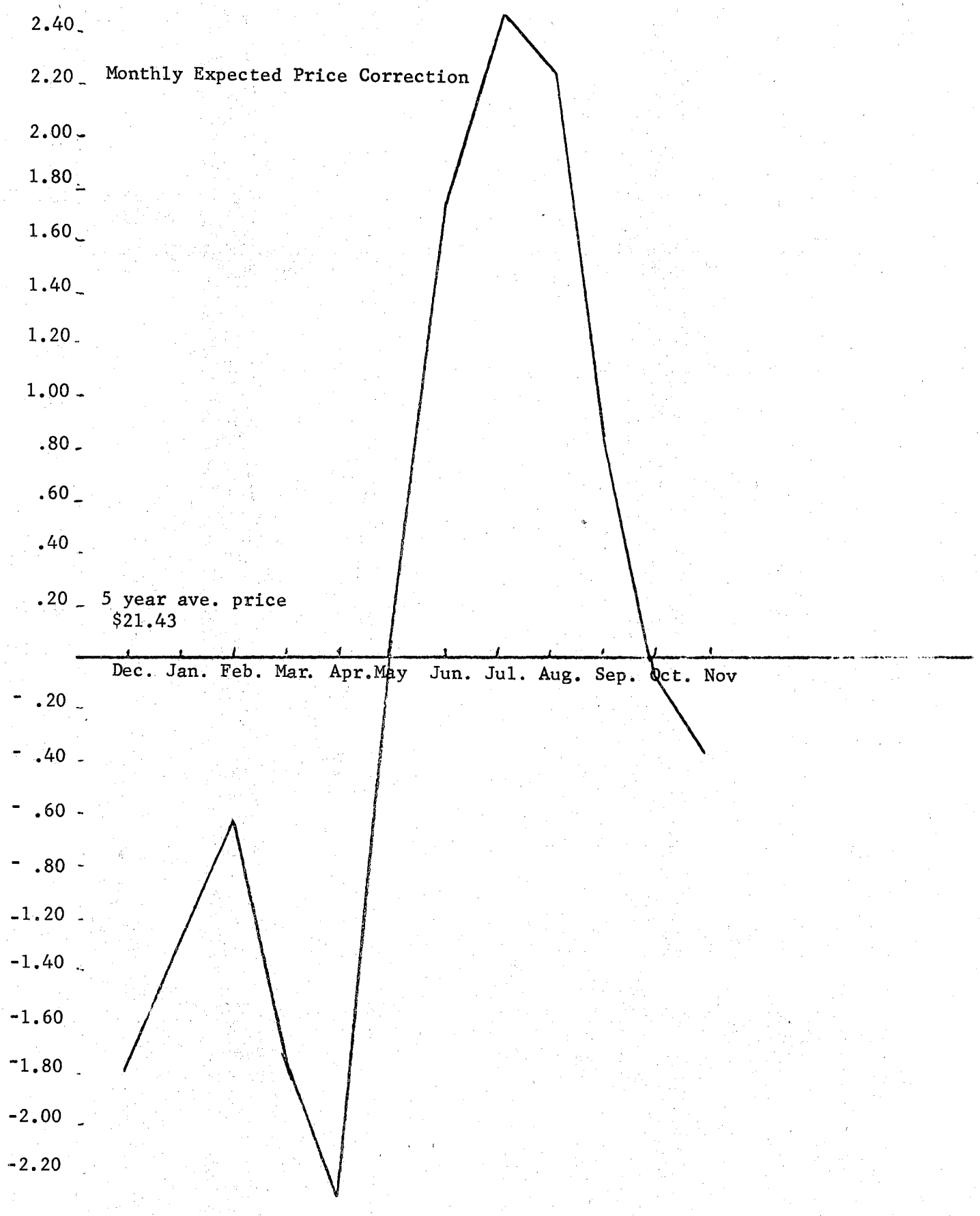

Monthly Expected Price Correction
2.40
2.20
2.00
1.80
1.60
1.40
1.20
1.00
.80
.60
.40
.20
5 year ave. price
$21.43
Dec. Jan. Feb. Mar. Apr. May Jun. Jul. Aug. Sep. Oct. Nov
- .20
- .40
- .60
- .80
-1.20
-1.40
-1.60
-1.80
-2.00
-2.20

Table 3. Average Prices of Barrows and Gilts Per Hundred Weight, Eight Markets, Five-Year Monthly Average Prices and Monthly Price Correction Factors, 1965-69.

Year	Annual Average Price	Average Prices Barrows and Gilts			
		Winter	Spring	Summer	Fall
1965	21.30	16.20	18.30	24.11	23.54
1966	23.49	27.93	23.28	25.19	21.53
1967	19.37	19.50	19.29	21.97	18.32
1968	19.19	18.30	18.98	20.67	18.71
1969	23.71	19.64	21.40	26.04	25.75
5-yr. ave.	21.41				

Month	5-yr. Monthly Ave. Price & Corrections from 5-yr. Ave.		
	Price	Correction	Monthly Price / 5-yr. Average
January	20.31	-1.10	.947
February	20.80	- .61	.971
March	19.92	-1.49	.929
April	19.38	-2.03	.904
May	21.46	.05	1.000
June	23.20	1.79	1.080
July	23.89	2.48	1.114
August	23.69	2.28	1.105
September	22.28	.87	1.039
October	21.38	- .03	.997
November	21.05	- .36	.982
December	19.87	-1.54	.927

5-Year Monthly Averages
5-Year Seasonal Averages
Annual Monthly
Annual Seasonal Averages

Table 4. Sows and Boars Sold.

Year	December	January	February		Annual Seasonal
1965	371	381	270	1022	341
1966	429	283	237	949	316
1967	427	384	311	1122	374
1968	477	403	308	1188	396
1969	428	402	325	1155	365
Over/5	2132	1853	1451	5436	1792
5-Year Monthly	426	371	290	1087	358 5-yr. seasonal

Year	March	April	May		Annual Seasonal
1965	340	400	396	1136	379
1966	302	324	437	1063	355
1967	357	358	457	1172	391
1968	287	389	436	1112	371
1969	313	349	375	1037	349
	1599	1820	2101		1845=369 5-yr.seasonal

Year	June	July	August		Annual Seasonal
1965	519	456	446	1421	474
1966	551	550	595	1696	565
1967	533	474	540	1547	516
1968	425	491	440	1356	452
1969	447	488	462	1397	466
	2475	2459	2483		2473=495 5-yr.seasonal

Year	September	October	November		Annual Seasonal
1965	378	358	424	1160	387
1966	471	417	459	1347	449
1967	458	408	450	1316	439
1968	400	400	473	1273	425
1969	483	497	392	1372	457
	2190	2080	2198		2157-431
	438	416	439		

Table 5. Comparison of Sow Farrowings and Sales, Average Monthly and Seasonal.

		Sows Farrowing			Sows Sold				Seasonal Change
5 year Average Seasonal	Month	5 year Average Monthly	5 year Monthly Difference %	5 year Seasonal Difference %	5 year Average Seasonal	5 year Monthly Average	5 year Monthly Difference %	5 year Seasonal Difference %	in Gilts Saved %
Winter	Dec	560	-45.0			426	+ 2.9		
775	Jan	758	-25.0	-23.0	358	371	-10.4	-13.6	-10.0
	Feb	1007	+ 0.5			290	-30.0		
Spring	Mar	1487	+47.0			320	-22.8		
1340	Apr	1522	+50.0	+32.0	369	364	-12.0	-11.0	+21.0
	May	1010	0.0			420	+ 1.4		
Summer	June	959	- 5.0			495	+20.0		
971	July	990	- 2.0	- 4.0	495	492	+19.0	+19.5	+15.0
	Aug	1063	+ 5.0			497	+20.0		
Fall	Sept	1408	+39.0			438	+ 5.7		
929	Oct	872	-14.0	- 8.0	431	416	+ 0.4	+ 4.0	- 4.0
	Nov	506	-50.0			439	+ 6.0		

cull and gilt retention. Sow sales data for the five year period indicate that 43.4%, 42.0%, 41.5%, 38.5%, and 40.9% of the sows farrowed were sold in 1965, 1966, 1967, 1968, and 1969 respectively. As a result a value of 40 percent was assumed for the cull factor (x) in Equation 3.

In the winter season about 23 percent fewer sows were farrowed than the five year annual average. And there were 13.6 percent fewer sows sold than the five year annual average. This implies that approximately 10 percent fewer gilts were bred during the winter season than the five year annual average. These data further indicate that about 21 percent more gilts were bred in the spring, 15 percent more in the summer, and 4 percent less in the fall than the five year annual average.

Derivation of the gilts bred data indicated that the average number of litters per female has ranged from 1.3 to 1.4 per year over the last five years. Basically this means that large numbers of gilts are bred in the late winter, farrowed in the spring or summer and sold soon after the pigs are weaned in the fall. Such a production pattern is consistent with the idea of a traditional producer who may lack capital, facilities, and/or labor for more intensified year round swine production. Since data upon which the above observations are based are nationally aggregated it seems reasonable to conclude that the "traditional" producer has and does dominate the swine production arena. This adds credibility to the need for anticipating the economic potentials for and consequences of shifts to more capital intensive and/or coordinated swine production systems.

4) Sows and Boars Sold

Data representing sales of sows and boars for the period 1965-69 was analyzed in an attempt to determine seasonal patterns. The conceptual model was simply:

(5) $SBT_t = f\,[\,SS_t,\ (TFB_{t-1} - TFB_t)]$

where:

SBT_t = number of sows and boars sold in month t

SS_t = seasonal correction factor for sow and boar sales

Results of this analysis are included in Tables 4 and 5. Table 4 includes annual, seasonal, and monthly numbers of sows and boars sold for the study period. Monthly and seasonal deviations from the five year annual average are calculated in percentage terms in Table 5.

From Table 5 it is estimated that about 13 percent fewer sows were sold in the winter than the five year annual average. In the spring 11 percent fewer were sold, 19.5 percent more in the summer, and 4 percent more in the annual average. These seasonal departures from the five year annual average correspond to season correction factors for sow and boar sales in Equation 5.

5) Barrows Available for Sale by Weight Group

Relationships considered in Equation 1, 2, 3, and 4 provide the basis for developing estimates of the number of barrows and gilts available for slaughter in various weight groups. Equations 5, 6, and 7 represent the models used in developing these estimates.

(6) $BGI_t = (SF_{t-6})(PPL_{t-6})\ 0.9 + (SF_{t-7})(PPL_{t-7})\ 0.15 - GS_t$

where:

BGI_t = number of barrows and gilts available for slaughter in the 180-220 pound weight group

SF = sows farrowed

PPL = pigs farrowed per litter in months t-5 and t-6.

Once equation 6 has been estimated, the next step is to estimate the

actual slaughter weight distribution from the flow of marketable barrows and gilts becoming available each month. This is the retention behavior of hog producers and is the subject of the next section.

6) The Retention Behaviors

The last section of this report describes an attempt to model the ex ante retention behavior which explains the slaughter weight distribution of barrows and gilts.

Although this work was of an exploratory nature, the preliminary results seem to indicate that more data is required if retention behavior is to be modeled successfully.

In the model specification, the retention functions follow logically from Equation 6.

The number of barrows and gilts in the 220-240 pound weight group available for slaughter in the period t+1 is estimated as a proportion of the number of barrows and gilts in the 180-220 pound group corrected by an economic adjustment factor, Equation 7.

$$(7)\ BG2_{t+1} = \left[(BG1_t) - ((BG1_t)\ (BS1s_t))\ SLCF1 \right] + GNFt$$

where:

BG2 = number of barrows and gilts available for slaughter 220-240 pounds.

BG1S = average proportion of 180-220 pound hogs slaughtered

SLCF1 = monthly adjustment factor for slaughter of 180-220 pound hogs

GNF = (gilts saved for breeding the previous month but not fertilized) $.15\ (GS_t)$

and where:

SLCF = f (profitability of keeping hogs to heavier weights)

The slaughter weight correction factor, SLCF1, is determined, based on the cost of carrying hogs from 200 to 240 pounds vs. the expected incremental revenue from the heavier hogs. For example:

$$(7A)\ \left[(EP^{BG2}_{t+1} - CBG2)\ 240\right] - \left[(PBG1_t - CBG1)\ 200\right]$$

Where:

EP^{BG2}_{t+1} = expected price of barrows and gilts 220-240 pounds in period t+1 ($/lb.)

CBG2 = cost per pound of barrows and gilts to 240 pounds

PBG1 = current price of barrows and gilts in weight group 1 ($/lb.)

CBG1 = cost per pound of barrows and gilts to 200 pounds.

If the value for 7A in a particular period were greater than zero more pigs would be held to heavier weights; otherwise, the normal slaughter weight distribution would apply.

Equation 8 is used to estimate the number of hogs available for slaughter at 240-270 pounds.

$$(8)\ BG3_{t+2} = BG1_t - (BG1_t)\ (BG1S_t)\ (SLCF1) - (BG1_t)\ (BG2S_{t+1})\ (SLCF2)$$

where:

$BG3_{t+2}$ = barrows and gilts available for slaughter 240-260 in period t+2

$BG2S_{t+1}$ = average proportion of barrows and gilts 180-200 pounds in t normally slaughtered as 220-240 pound hogs in t+1

SLCF2 = monthly adjustment for slaughter of 220-240 pound hogs based on profitability as in Equation 6A with appropriate weight considerations.

All hogs not slaughtered in weight groups 1, 2, and 3 are assumed slaughtered in the group greater than 270 pounds. Therefore:

$$(9)\ BG4_{t+3} = BG1_t - (BG1_t)(BG1S_t)(SLCF1) - (BG1_t)(BG2S_{t+1})(SLCF2) - (BG1_t)(BG3S_{t+2})(SLCF3)$$

where:

$BG3S_{t+2}$ = average proportion of barrows and gilts 180-220 pounds in t slaughtered at 240-260 pounds in t+2

SLCF3 = monthly adjustment for slaughter of 240-260 pound hogs as in Equation 7A with appropriate weight considerations.

However, there were no published data series of the slaughter weight distribution of barrows and gilts by months so two alternative models were developed. First, an attempt was made to derive the lagged relationships involved in retention behavior using spectral analysis. Secondly, an L-P model was built based on Equations 7, 7A, 8 and 9 to test various price hypotheses of retention behavior.

A) <u>Frequency Domain Regression</u>

The purpose of this model was to explore the usefulness of spectral analysis for estimating its lagged relationships involved in the retention function.

e.g.,

$$BG\ (I)_t = f\ (\ (H/c)_{t-1},\ (H/C)_{t-2},\ \ldots\ldots,\ (H/C)_{t-24})$$

where $BG\ (I)_t$ = # barrows and gilts slaughtered in weight group i in month t.

$(H/C)_{t-j}$ = Hog: corn price ratio in month t-j.

However, since there was insufficient data available on the weight distribution of hogs slaughtered, an expository analysis was performed with the following function:

$$BG_t = F\ (\ (H/C)_t,\ (H/C)_{t-1},\ \ldots.\ (H/C)_{t-12})$$

where BG_t = total number barrows and gilts slaughtered in t.

This function was estimated with a spectral analysis program which contained a subroutine to compute time domain regression coefficients. A sample spectral density function is given in figure 4.

Although the numerical results of the analysis were not directly of value to the project, the authors believe that this approach to the estimation of unconstrained lagged relationships[1/] like the retention function would be valuable provided the data was available.

The modified computer program used in this analysis is presented in Appendix 1.

B) Linear Programming Model

The model assumes that a farmer's decision to sell marketable barrows and gilts in any month is based on the current price, the cost of holding over to the following month, the expected price in that month, and that his objective is to maximise expected profit over the two month planning horizon. Then, given an expected price hypothesis, the model becomes a simple linear programming problem.

This model was used to test various expected price hypothesis by measuring how closely the slaughter weight distribution of barrows and gilts produced by the model matched the sample slaughter weight distribution given in Table 6.

Three expected price hypotheses were tested:

i) $EP_{t+1} = P_{t-1}$

1/ As opposed to lags specified a prisni.

FIGURE 4

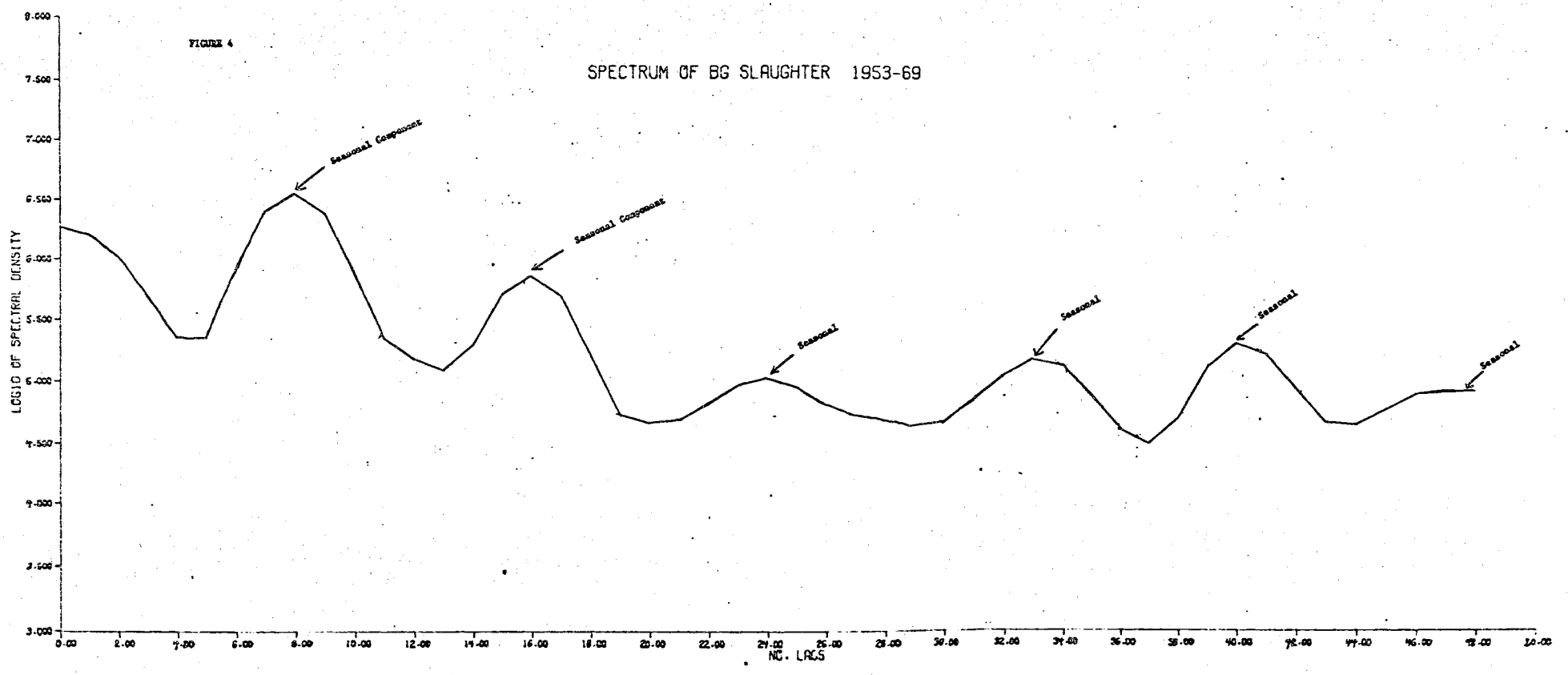

SPECTRUM OF BG SLAUGHTER 1953-69
LOG10 OF SPECTRAL DENSITY
NO. LAGS
Seasonal Component
Seasonal Component
Seasonal
Seasonal
Seasonal
Seasonal

Table 6. The Percentage of Barrows and Gilts slaughtered in four weight groups by months in a sample of seven terminal and three interior markets (July 1969, - June, 1970)

Month	Weight Group (lbs)			
	1 180 - 219	2 220 - 239	3 240 - 270	4 over 270
Jan.	22.93	33.26	30.89	12.91
Feb.	29.26	34.36	28.42	7.97
Mar.	25.32	37.91	28.75	8.02
Apr.	19.30	33.01	35.02	12.67
May	17.24	35.65	33.36	13.76
June	24.95	30.85	26.97	17.23
July	33.44	37.69	22.87	6.00
Aug.	32.61	52.88	12.00	2.52
Sept.	38.44	45.93	14.11	1.52
Oct.	21.13	40.92	34.38	3.58
Nov.	19.74	36.32	33.19	10.75
Dec.	28.23	26.27	32.94	12.56

Table 7. Generalized Matrix

Max. Profit C_j --->			C_1	C_2	C_3	C_4	C_5	C_6	C_7	-1000							-1000
			Selling Activ.				Reten. Activ.			Infeasibilities							
			1	2	3	4											
			S1T1	S2T1	S3T1	S4T1	T12	T23	T34	Cos1	Cost1	Cos2	Cost2	Cos3	Cost3	Cos4	Cost4
B&G avail. in wt.gr.																	
1	b_1	$\geq$	+1				+1										
2	b_2	$\geq$		+1				+1									
3	b_3	$\geq$			+1				+1								
4	b_4	$\geq$				+1											
Tot. Sales	b_5	=	+1	+1	+1	+1											
Slaught.wt Distr.																	
1	b_6	=								-1	+1						
2	b_7	=										-1	+1				
3	b_8	=												-1	+1		
4	b_9	=														-1	+1

ii) $EP_{t+1} = P_{t-11}$

iii) $EP_{t+1} = P_{t-1} + (P_{t-2} - P_{t-3})$

Where EP_{t+1} = the expected price in the following month.

P_{t-1} = the price in the preceding month taken to be the current price

The 'goodness of fit' of each hypothesis to the sample data was measured by constraining the model to sell hogs according to the sample distribution and allowing the program to deviate from this pattern only at high cost by the inclusion of a set of infeasibility activities. The model for one time period is presented in table 7.

The criterion for choosing between hypotheses was to minimize the number of infeasibilities produced when the model was solved sequentially over a nine month period.

Data

The weight distribution of marketable barrows and gilts on farms for the initial period was calculated from the total number sold in the following four months (L & M statistics) and the sample weight distribution in Table 6.

$$BG_t\ (1) = BG^s_t\ (1) + BG^s_{t+1}\ (2) + BG^s_{t+2}\ (3) + BG^s_{t+3}\ (4)\ \ldots.$$

where BG_t (i) = barrows and gilts in weight group i in period t available on farms

s = slaughtered

The total number of barrows and gilts slaughtered each month was calculated from Livestock and Meat Statistics (see Table 8).

Table 8

Barrow and Gilt Slaughter (U.S.)

Calculated from L & M statistics. Barrows and Gilts slaughtered = Commercial slaughter of hogs times percentage of Federally inspected hogs that were barrows and gilts

1,000 head

Month	1965	1966	1967	1968	1969
Jan.	6513.0	5173.4	6822.2	7060.3	7187.9
Feb.	5816.8	5110.9	6193.3	6341.7	6584.0
Mar.	7089.6	6334.4	7250.3	6766.0	7126.7
Apr.	6182.3	5739.7	6328.1	6888.2	7105.3
May	5017.6	5187.8	5640.7	6740.9	6215.8
June	4832.5	4806.8	5355.1	5354.9	5628.1
July	4586.7	4271.4	4948.7	5607.4	5744.5
Aug.	4960.6	5224.2	6072.4	6193.2	5737.3
Sept.	5865.3	6163.4	6447.9	6645.9	6636.3
Oct.	5805.0	6430.4	7176.9	7818.7	7173.7
Nov.	5815.1	6622.8	6935.3	6851.2	5977.4
Dec.	5345.2	6732.7	6601.5	6950.7	6552.2

Table 9

Sows Slaughtered (U.S.)

Calculated from L & M. Sows slaughtered = commercial hog slaughter times percentage of Federally inspected hogs that were sows.

			1,00 head		
Month	1965	1966	1967	1968	1969
Jan.	440.7	332.0	445.6	469.2	434.5
Feb.	314.3	275.8	362.0	358.2	364.2
Mar.	391.4	349.3	407.5	328.0	346.2
Apr.	461.7	374.5	412.8	442.0	385.1
May	463.2	509.1	533.7	494.0	414.4
June	602.7	646.8	619.1	487.4	494.8
July	529.6	642.7	553.6	558.9	540.2
Aug.	519.7	695.4	558.8	497.6	509.0
Sept.	437.5	540.1	525.6	448.7	527.7
Oct.	412.9	479.2	468.2	448.2	544.0
Nov.	487.8	531.0	523.7	534.4	433.0
Dec.	430.2	500.6	499.3	543.4	481.7

Table 10.

Monthly Farrowings (U.S.)

Published till 1967. 1968-69 calculated from the 1965-67 dispersion on a quarterly basis

Source: Dr. L. Duewer

1,000 head

Month	1965	1966	1967	1968	1969
Jan.	708	732	794	834	838
Feb.	953	963	1051	1107	1113
Mar.	1404	1448	1525	1521	1406
Apr.	1395	1531	1552	1559	1441
May	913	1001	1063	1032	954
June	855	977	968	1038	969
July	749	951	899	962	898
Aug.	944	1081	1080	1152	1075
Sept.	1251	1423	1433	1806	1413
Oct.	768	887	892	932	874
Nov.	439	492	548	539	506
Dec.	526	606	616	620	663

Table 11.

	Weight Group							
	1		2		3		4	
Year	Grade	Wt. Range	Grade	Wt. Range	Grade	Wt. Change	Grade	Wt. Chg.
1965	1-2	200-220	1-2	220-240	2-3	240-270	2-3	270-300
1966	1-2	200-220	1-2	220-240	2-3	240-270	2-3	270-300
1967	1-2	200-220	1-2	220-240	2-3	240-270	2-3	270-300
*	1-2	200-220						
1968	2-3	200-220	2-3	220-240	2-3-4	240-270	3-4	240-270
1969	2-3	200-220	2-3	220-240	2-3	240-270	3-4	240-270

* Wherever possible the revised grading system of July 1968 was adjusted for.

The number of barrows and gilts reaching weight group one (180-200 lbs) was calculated using the following set of equations utilizing the data given in tables 9 and 10 and LM statistics.

(1) BG_t (I) = TSF_{t-6} X PPL^*_{t-6}) - G $S._t$

(2) $GS_t = GB_{t+2}$

(3) $GB_t = TFB_t - (TFB_{t-1} - SS_t)$

(4) $TFB_t = (TSF_{t+4})$ (1.25)

Where

BG_t (I) = barrows and gilts available for slaughter in weight group I (180 - 200 lbs) in month t.

TSF = total sows farrowed (Dec. - Nov)

PPL* = pigs per litter minus deaths to slaughter (assumed to be 4).(L&M)

GS = gilts saved

GB = gilts bred

TFB - Total females bred

SS = sows slaughtered (L&M State)

(1.25) = conception rate (assumes a 20% misconception rate)

The transfer activities enabling hogs to move to the next weight group in the next period (1 month) were assumed to require the following amounts of feed.

	Wt. Gain/Month	F.C.E.*	Feed Required lbs.
Transfer 1 to 2	30 lbs.	4.12	124 lbs.
Transfer 2 to 3	25 lbs.	4.25	106 lbs.
Transfer 3 to 4	20 lbs.	4.37	871 lbs.

* Feed Conversion efficiency.

The feed cost was taken as the Chicago price of No. 3 Yellow Corn in the month of transfer.

Table 12. Sample Output of Linear Programming Model Using Price Hypothesis 1.

Period (months)	Infeasibility Cost (1 unit = $1000) Weight Group 1	2	3	4
1	0	0	0	0
2	0	0	0	0
3	-21 1/	0	0	-1
4	0	0	+192 2/	-192
5	0	0	+31	-31
6	0	0	0	0
7	0	+485	-661	+176
8	0	+669	-524	-145
9	0	+877	-776	-101
10	0	+781	-524	-257
11	0	0	+643	-643

1/ Minus sign indicates lower bound

2/ Plus sign indicates upper bound

Results

The model was not sensitive to the different expected price hypotheses for three reasons. They are: the deterministic nature of the model, the very simple price expectations used, and the oversimplified mechanism whereby weight group four hogs must be sold in the current month.

Nevertheless, we consider that if the model was refined and more slaughter weight distribution data were available, the model could be useful for

testing price hypothesis. Sample output for the L-P model using price hypothesis (i) (page 11) is presented in Table 12.

The Computer Program

A program was written to enable the series of L-P problems to be solved sequentially using as a base the standard RADESLP program.

The complete routine is outlined as follows:

```
__________, _______________, T40, CM76000, P5, L2000
:
MAP (PART)
FUN (S,,,,, KGO)
LIBCOPY (CSEBIN, LGO, RADESLP)
LGO (          )
:
KGO
:
LGO
etc.
EXIT
7/8/9
        (Standard RADESLP input)
7/8/9
        (Change Data)
7/8/9
        Program DELTA
7/8/9
6/7/8/9
```

The control cards and program delta are presented in Appendix 2

APPENDIX 1

Spectral Analysis Computer Program*

(Tukan Weights)

* A subprogram (DLAG) has been added to this program to compute regression coefficients for a lagged function of two variables.

```
                PROGRAM MULTS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
000003          DIMENSION C(50),X(1200),Y(1200),SPECX(300),SPECY(300),B(50),FL
               1(300),RSPECX(300),RSPECY(300),COHSQ(300),SPECRS(300),WEIGTS(30
               2SPECTR(6,6,300)
000003          EQUIVALENCE (X(301),RSPECX),(X(901),SPECRS),(Y(301),RSPECY),
               1(Y(901),FLTFCN)
000003          COMMON X,Y,SPECX,SPECY,COHSQ,WEIGTS,SPECTR,NRSERS,NRDATA,NRLAG
               1NRLSP1,PI,L,M
          C  TUKAN VERSION OF POWER AND CROSS SPECTRUN POUS CORRELATION PROGRA
000003          READ(5,1) NRSETS
000011          K=1
000012       44 READ(5,1) NRSERS,NRDATA,NRFICS,NRLAGS
000026        1 FORMAT(4I10)
000026          WRITE(6,2)
000032        2 FORMAT(//4X,6HNRSERS,4X,6HNRDATA,4X,6HNRFICS,4X,6HNRLAGS)
000032          WRITE(6,1) NRSERS,NRDATA,NRFICS,NRLAGS
000046          IF(NRFICS) 6,6,3
000050        3 READ(5,4)  (C(I),I=1,NRFICS)
000063        4 FORMAT (6F12.0)
000063          WRITE(6,5)
000067        5 FORMAT(/26X,20H FILTER COEFFICIENTS)
000067          WRITE(6,50) (C(I),I=1,NRFICS)
000102       50 FORMAT(1H ,6F20.8)
000102       6  L=1
000103       39 M=L+1
000105       35 READ(5,4)  (X(I),I=1,NRDATA)
000120          WRITE(6,7)  L
000126        7 FORMAT(/25X,20H ORIGINAL SERIES NO I2//)
000126          WRITE(6,50) (X(I),I=1,NRDATA)
000141          IF(NRFICS) 12,12,8
000143        8 NRDATA=NRDATA-(NRFICS-1)
000146          DO 10 I=1,NRDATA
000147          CX=0.0
000150          DO 9 J=1,NRFICS
000152          IPJM1=I+J-1
000155        9 CX=CX+C(J)*X(IPJM1)
000162       10 X(I)=CX
000166          WRITE(6,11)  L
000174       11 FORMAT(/25X,20H FILTERED SERIES NO I2//)
000174          WRITE(6,50) (X(I),I=1,NRDATA)
000207          NRDATA=NRDATA+(NRFICS-1)
000212       12 READ(5,4)  (Y(I),I=1,NRDATA)
000225          WRITE(6,7)  M
000233          WRITE(6,50) (Y(I),I=1,NRDATA)
000246          IF(NRFICS) 16,16,13
000250       13 NRDATA=NRDATA-(NRFICS-1)
000253          DO 15 I=1,NRDATA
000254          CY=0.0
000255          DO 14 J=1,NRFICS
000257          IPJM1=I+J-1
000262       14 CY=CY+C(J)*Y(IPJM1)
000267       15 Y(I)=CY
000273          WRITE(6,11)  M
000301          WRITE(6,50) (Y(I),I=1,NRDATA)
000314       16 CALL POCROP
000315          IF(NRFICS) 17,17,19
```

```
000317      17 DO 18 I=1,NRLSP1
000321         RSPECX(I)=SPECX(I)
000323      18 RSPECY(I)=SPECY(I)
000327         GO TO 27
000330      19 DO 22 I=1,NRFICS
000332         B(I)=0.0
000333         NRFCS=NRFICS-(I-1)
000336         DO 20 J=1,NRFCS
000337         JPIM1=J+I-1
000342      20 B(I)=B(I)+C(J)*C(JPIM1)
000350         IF(I-1) 22,22,21
000352      21 B(I)=2.0*B(I)
000355      22 CONTINUE
000360         FNRLS=NRLAGS
000361         ANG=PI/FNRLS
000363         DO 23 I=1,NRLSP1
000364         FI=I-1
000366         FLTFCN(I)=0.0
000367         DO 23 J=1,NRFICS
000371         FJ=J-1
000373      23 FLTFCN(I)=FLTFCN(I)+B(J)*COS(FJ*FI*ANG)
000410         DO 26 I=1,NRLSP1
000412         IF(FLTFCN(I)) 24,24,25
000414      24 RSPECX(I)=10.0**35
000421         RSPECY(I)=10.0**35
000426         GO TO 26
000427      25 RSPECX(I)=SPECX(I)/FLTFCN(I)
000432         RSPECY(I)=SPECY(I)/FLTFCN(I)
000434      26 CONTINUE
000437      27 DO 28 I=1,NRLSP1
000441      28 SPECRS(I)=(1.0-COHSQ(I))*RSPECY(I)
000447         WRITE(6,29)  L,M
000456      29 FORMAT(/21X,29HRECOLORED SPECTRUM OF SERIES I2,5H AND I2,41H AN
             1SPECTRAL ESTIMATES OF THE RESIDUALS//)
000456         WRITE(6,30)
000462      30 FORMAT(24X,1HI,8X,9HRSPECX(I),12X,1HI,8X,9HRSPECY(I),12X,1HI,8)
             1 9HSPECRS(I),18X//)
000462         DO 31 I=1,NRLSP1
000464         IM1=I-1
000466      31 WRITE(6,32) IM1,RSPECX(I),IM1,RSPECY(I),IM1,SPECRS(I)
000510      32 FORMAT(15X,3(6X,I4,1PE20.7))
000510         IF(NRSERS-M) 36,36,33
000512      33 M=M+1
000514         IF(NRFICS) 35,35,34
000515      34 NRDATA=NRDATA+(NRFICS-1)
000520         GO TO 35
000520      36 IF(NRSERS-(L+1)) 40,40,37
000523      37 L=L+1
000525         IF(NRFICS) 39,39,38
000526      38 NRDATA=NRDATA+(NRFICS-1)
000531         GO TO 39
000531      40 IF(NRSERS-2) 42,42,41
000534      41 CALL PARCOR
000535      42 IF(NRSETS-K) 45,45,43
000540      43 K=K+1
000542         GO TO 44
000542      45 CALL EXIT
000543         END
```

```
PROGRAM LENGTH INCLUDING I/O BUFFERS
003114

UNUSED COMPILER SPACE
036700


                 SUBROUTINE POCROP
000002           DIMENSION X(1200),Y(1200),
                1     PRODXX(300),PRODXY(300),PRODYX(300),PRODYY(300),AUCVX(300)
                2AUCVY(300),CRCVXY(300),CRCVYX(300),WACVX(300),WACVY(300),WCCVX
                3(300),WCCVYX(300), ACXPR(300),ACYPR(300),CCXYPR(300),CCYXPR(30
                5CRAMPL(300),COHSQ(300),GAIN(300),PHASE(300),WEIGTS(300),
                4            SPECX(300),SPECY(300),WCO(300),WQUAD(300),RSQ(300),
                6SPECTR(6,6,300)
000002           EQUIVALENCE (X,AUCVX,WCO),(X(301),CRCVXY),(X(601),WCCVXY),(X(9
                1ACXPR),
                2     (Y,AUCVY,WQUAD),(Y(301),CRCVYX),(Y(601),WCCVYX),(Y(901),
                3ACYPR),(SPECX,WACVX),(SPECY,WACVY),(COHSQ,CCXYPR),(WEIGTS,CCYX
                4(PRODXX,RSQ),(PRODXY,CRAMPL),(PRODYX,GAIN),(PRODYY,PHASE)
000002           COMMON X,Y,SPECX,SPECY,COHSQ,WEIGTS,SPECTR,NRSERS,NRDATA,NRLAG
                1NRLSP1,PI,L,M
000002           NRLSP1=NRLAGS+1
000004           NRDTML=NRDATA-NRLAGS
000006           FNRLS=NRLAGS
000007           PI = 3.14159265359
000010           ANG = PI/FNRLS
000012           SUMX=0.0
000013           SUMY=0.0
000014           DO 1 J = 1,NRDATA
000015           SUMX=SUMX+X(J)
000017         1 SUMY=SUMY+Y(J)
000023           Z1 = NRDATA
000024           SUMX = SUMX/Z1
000026           SUMY = SUMY/Z1
000027           DO 900 I = 1,NRDATA
000030           X(I) = X(I) - SUMX
000033    900    Y(I) = Y(I) - SUMY
000040           NRLSO2 =FNRLS/2.0+1.0
000044           DO 2 I = 1,NRLSO2
000045           FI = I -1
000047    2      WEIGTS(I) = 1.0 -(6.0*FI**2/FNRLS**2)*(1.0-FI/FNRLS)
000062           NLO2P1 = NRLSO2 + 1
000064           DO 3 I = NLO2P1,NRLSP1
000065           FI = I - 1
000067    3      WEIGTS(I) = 2.0*(1.0-FI/FNRLS)**3
000076           DO 4 J=1,NRLSP1
000100           PRODXX(J)=0.0
000101           PRODXY(J)=0.0
000103           PRODYX(J)=0.0
000104           PRODYY(J)=0.0
000106           MN=NRDATA-(J-1)
000110           JM=J
000111           DO 4 I=1,MN
000112           PRODXX(J)=PRODXX(J)+X(I)*X(JM)
```

```
000116          PRODYY(J)=PRODYY(J)+Y(I)*Y(JM)
000123          PRODXY(J)=PRODXY(J)+X(I)*Y(JM)
000127          PRODYX(J)=PRODYX(J)+Y(I)*X(JM)
000134        4 JM=JM+1
000142          DENOM=NRDATA
000144          FDEN=1.0/DENOM
000145          DO 5 I=1,NRLSP1
000147          AUCVX(I)=FDEN*(PRODXX(I))
000152          AUCVY(I)=FDEN*(PRODYY(I))
000155          CRCVXY(I)=FDEN*(PRODXY(I))
000157        5 CRCVYX(I)=FDEN*(PRODYX(I))
000164          WRITE(6,6)  L,M
000173        6 FORMAT(/32X,46HAUTO AND CROSS COVARIANCE FUNCTIONS OF SERIES I
               1 5H AND I2//)
000173          WRITE(6,7)
000177        7 FORMAT(9X,1HI,9X,8HAUCVX(I),12X,1HI,9X,8HAUCVY(I),12X,1HI,8X,
               1 9HCRCVXY(I),12X,1HI,8X,9HCRCVYX(I),3X//)
000177          DO 8 I=1,NRLSP1
000201          IM1=I-1
000203        8 WRITE(6,9) IM1,AUCVX(I),IM1,AUCVY(I),IM1,CRCVXY(I),
               1 IM1,CRCVYX(I)
000231        9 FORMAT(4(6X,I4,1PE20.7))
000231          DO 10 I = 1,NRLSP1
000232          WACVX(I)=WEIGTS(I)*AUCVX(I)
000234          WACVY(I)=WEIGTS(I)*AUCVY(I)
000237          WCCVXY(I)=WEIGTS(I)*CRCVXY(I)
000241   10     WCCVYX(I)=WEIGTS(I)*CRCVYX(I)
000246          WRITE(6,12)  L,M
000256       12 FORMAT(/27X,56HAUTO AND CROSS COVARIANCE TRANSFORM FUNCTIONS O
               1RIES I2,5H AND I2//)
000256          WRITE(6,13)
000262       13 FORMAT(9X,1HI,8X,9H WACVX(I),12X,1HI,8X,9H WACVY(I),12X,1HI,10
               19HWCCVXY(I),10X,1H1,9X,9HWCCVYX(I),4X//)
000262          DO 14 I=1,NRLSP1
000264          IM1=I-1
000266       14 WRITE(6,9) IM1, WACVX(I),IM1, WACVY(I),IM1,WCCVXY(I),IM1,
               1WCCVYX(I)
000314          ACXPR (1)=WACVX (1)
000315          ACYPR (1)=WACVY (1)
000317          CCXYPR(1) = WCCVXY(1)
000320          CCYXPR(1) = WCCVYX(1)
000322          DO 15 I = 2,NRLAGS
000323          ACXPR(I)=2.0*WACVX(I)
000325          ACYPR(I)=2.0*WACVY(I)
000330          CCXYPR(I)=2.0*WCCVXY(I)
000332   15     CCYXPR(I)=2.0*WCCVYX(I)
000336          ACXPR(NRLSP1)=WACVX(NRLSP1)
000340          ACYPR(NRLSP1)=WACVY(NRLSP1)
000342          CCXYPR(NRLSP1)=WCCVXY(NRLSP1)
000344          CCYXPR(NRLSP1)=WCCVYX(NRLSP1)
000346          DO 17 I = 1,NRLSP1
000347          SPECX(I)=0.0
000350          SPECY(I)=0.0
000352          WCO(I)=0.0
000353          WQUAD(I)=0.0
000355          FI = I-1
000357          DO 17 J = 1,NRLSP1
000360          FJ = J-1
```

```
000362          ANGLE=FJ*FI*ANG
000365           SPECX(I)=SPECX(I)+ACXPR(J)*COS(ANGLE)
000372          SPECY(I)=SPECY(I)+ACYPR(J)*COS(ANGLE)
000400          WCO(I)=WCO(I)+0.50*(CCXYPR(J)+CCYXPR(J))*COS(ANGLE)
000411    17    WQUAD(I)=WQUAD(I) + 0.50*(CCXYPR(J)-CCYXPR(J))*SIN(ANGLE)
000427          CALL DLAG(WCO,WQUAD,SPECX)
000431          WRITE(6,18)  L,M
000441       18 FORMAT(/36X,38HSPECTRUM AND CROSS SPECTRUM OF SERIES I2,5H AND
               1//)
000441          WRITE(6,19)
000445       19 FORMAT(9X,1HI,9X,8HSPECX(I),12X,1HI,9X,8HSPECY(I),12X,1HI,10X,
               1 6HWCO(I),13X,1HI,9X,8HWQUAD(I),3X//)
000445          DO 20 I=1,NRLSP1
000447          IM1=I-1
000451       20 WRITE(6,9) IM1,SPECX(I),IM1,SPECY(I),IM1,WCO(I),IM1,
               1 WQUAD(I)
000477          DO 24 I=1,NRLSP1
000500          SPECTR(L,L,I)=SPECX(I)
000505          SPECTR(M,M,I)=SPECY(I)
000512          SPECTR(L,M,I)=WCO(I)
000517       24 SPECTR(M,L,I)=WQUAD(I)
000527          DO 26 I=1,NRLSP1
000530          RSQ(I)=WCO(I)**2+WQUAD(I)**2
000533          CRAMPL(I)=SQRT(RSQ(I))
000537          COHSQ(I)=RSQ(I)/(SPECX(I)*SPECY(I))
000543          GAIN(I)=CRAMPL(I)/SPECX(I)
000545          ADDITN=PI
000547          IF(WCO(I)) 26,26,25
000551       25 ADDITN=ADDITN-SIGN(PI,WQUAD(I))
000556       26 PHASE(I)=ATAN(WQUAD(I)/WCO(I))+ADDITN
000567          WRITE(6,27)  L,M
000577       27 FORMAT(/38X,35HCROSS SPECTRAL ESTIMATES OF SERIES I2,5H AND I2
000577          WRITE(6,28)
000603       28 FORMAT(/9X,1HI,8X,9HCRAMPL(I),12X,1HI,11X,9HCOH.SQ(I),9X,1HI,1
               1 7HGAIN(I),10X,1HI,12X,8HPHASE(I)//)
000603          DO 29 I=1,NRLSP1
000605          IM1=I-1
000607       29 WRITE(6,30) IM1,CRAMPL(I),IM1,COHSQ(I),IM1,GAIN(I),IM1,
               1 PHASE(I)
000635       30 FORMAT(6X,I4,1PE20.7,3(6X,I4,0PF20.8))
000635          RETURN
000635          END

SUBPROGRAM LENGTH
003337

UNUSED COMPILER SPACE
036100


                SUBROUTINE PARCOR
000002          DIMENSION SPECTR(6,6,300),CC(10,10),CORELN(6,6,300),CORESP(5,6,
               1),X(1200),SPECX(300),Y(1200),SPECY(300),COHSQ(300),WEIGTS(300)
000002          DIMENSION COVARS(7,14),A(2,10),B(5,4),C(5,10),CI(5,10),AXCI(2,1
```

```
                1 AXCIXB(2,4),D(2,4)
000002           EQUIVALENCE (X,DUMX),(Y,DUMY),(SPECX,DUMSPX),(SPECY,DUMSPY),(CC
                1,DUMCOS),(WEIGTS,DUMWET),(SPECTR,CORELN)
000002           COMMON X,Y,SPECX,SPECY,COHSQ,WEIGTS,SPECTR,NRSERS,NRDATA,NRLAGS
                1NRLSP1,PI,L,M
000002           NSP1=NRSERS+1
000004           NSM1=NRSERS-1
000005           NSM1X2=NSM1*2
000006           NSM2=NRSERS-2
000010           NSM2X2=NSM2*2
000011           KK=1
000012        43 DO 1 J=1,NRSERS
000014           DO 1 I=1,J
000015           COVARS(I,J)=SPECTR(I,J,KK)
000024           COVARS(J,I)=SPECTR(I,J,KK)
000032           COVARS(I,J+7)=SPECTR(J,I,KK)
000040         1 COVARS(J,I+7)=-SPECTR(J,I,KK)
000051           DO 2 I=1,NRSERS
000053         2 COVARS(I,I+7)=0.0
000060           L=1
000061        15 DO 3 I=1,NSM1
000063           A(1,I)=COVARS(1,I+1)
000066           A(1,I+5)=COVARS(1,I+8)
000071           B(I,3)=COVARS(I+1,8)
000073         3 B(I,1)=COVARS(I+1,1)
000076           DO 4 I=1,NSM1
000100           DO 4 J=1,NSM1
000101           C(I,J+5)=COVARS(I+1,J+8)
000106         4 C(I,J)=COVARS(I+1,J+1)
000116           DO 5 I=1,NSM1
000117           II=I+NSM1
000121           DO 5 J=1,NSM1
000122           JJ=J+NSM1
000124           CC(I,J)=C(I,J)
000131           CC(I,JJ)=C(I,J+5)
000136           CC(II,J)=-C(I,J+5)
000142         5 CC(II,JJ)=C(I,J)
000153           CALL INVERT(NSM1X2,CC,SING)
000155           IF(SING) 8,8,6
000157         6 WRITE(6,7)
000163         7 FORMAT(/40X,39H INVERSE NOT COMPUTABLE BY THIS ROUTINE)
000163           GO TO 12
000164         8 DO 9 I=1,NSM1
000166           DO 9 J=1,NSM1
000167           JJ=J+NSM1
000171           CI(I,J)=CC(I,J)
000175         9 CI(I,J+5)=CC(I,JJ)
000206           DO 10 I=1,NSM1
000207           AXCI(1,I)=0.0
000211           AXCI(1,I+5)=0.0
000212           DO 10 J=1,NSM1
000214           AXCI(1,I+5)=AXCI(1,I+5)+A(1,J)*CI(J,I+5)+A(1,J+5)*CI(J,I)
000227        10 AXCI(1,I)=AXCI(1,I)+A(1,J)*CI(J,I)-A(1,J+5)*CI(J,I+5)
000245           AXCIXB(1,1)=0.0
000246           AXCIXB(1,3)=0.0
000247           DO 11 I=1,NSM1
000250           AXCIXB(1,3)=AXCIXB(1,3)+AXCI(1,I)*B(I,3)+AXCI(1,I+5)*B(I,1)
000257        11 AXCIXB(1,1)=AXCIXB(1,1)+AXCI(1,I)*B(I,1)-AXCI(1,I+5)*B(I,3)
```

```
000270          CORELN(NRSERS,L,KK)=AXCIXB(1,1)/COVARS(1,1)
000276       12 DO 13 I=1,NRSERS
000300          COVARS(NSP1,I)=COVARS(1,I)
000305          COVARS(NSP1,I+7)=COVARS(1,I+7)
000307          COVARS(I,NSP1+7)=COVARS(I,8)
000312       13 COVARS(I,NSP1)=COVARS(I,1)
000316          COVARS(NSP1,NSP1)=COVARS(1,1)
000321          COVARS(NSP1,NSP1+7)=COVARS(1,8)
000324          DO 14 I=1,NRSERS
000325          DO 14 J=1,NRSERS
000326          COVARS(I,J+7)=COVARS(I+1,J+8)
000333       14 COVARS(I,J)=COVARS(I+1,J+1)
000342          L=L+1
000343          IF(L-NRSERS) 15,15,16
000345       16 L=1
000346       17 M=1
000347       18 DO 19 I=1,2
000351          DO 19 J=1,NSM2
000352          A(I,J)=COVARS(I,J+2)
000356          A(I,J+5)=COVARS(I,J+9)
000361          B(J,I+2)=COVARS(J+2,I+7)
000365       19 B(J,I)=COVARS(J+2,I)
000375          DO 20 I=1,NSM2
000376          DO 20 J=1,NSM2
000377          C(I,J+5)=COVARS(I+2,J+9)
000404       20 C(I,J)=COVARS(I+2,J+2)
000414          DO 21 I=1,NSM2
000415          II=I+NSM2
000417          DO 21 J=1,NSM2
000420          JJ=J+NSM2
000422          CC(I,J)=C(I,J)
000427          CC(I,JJ)=C(I,J+5)
000434          CC(II,J)=-C(I,J+5)
000440       21 CC(II,JJ)=C(I,J)
000451          CALL INVERT(NSM2X2,CC,SING)
000453          IF(SING) 23,23,22
000455       22 WRITE(6,7)
000461          GO TO 34
000462       23 DO 24 I=1,NSM2
000464          DO 24 J=1,NSM2
000465          JJ=J+NSM2
000467          CI(I,J)=CC(I,J)
000473       24 CI(I,J+5)=CC(I,JJ)
000504          DO 25 I=1,2
000505          DO 25 K=1,NSM2
000506          AXCI(I,K)=0.0
000510          AXCI(I,K+5)=0.0
000512          DO 25 J=1,NSM2
000514          AXCI(I,K+5)=AXCI(I,K+5)+A(I,J)*CI(J,K+5)+A(I,J+5)*CI(J,K)
000531       25 AXCI(I,K)=AXCI(I,K)+A(I,J)*CI(J,K)-A(I,J+5)*CI(J,K+5)
000552          DO 26 I=1,2
000553          DO 26 K=1,2
000554          AXCIXB(I,K)=0.0
000556          AXCIXB(I,K+2)=0.0
000560          DO 26 J=1,NSM2
000562          AXCIXB(I,K+2)=AXCIXB(I,K+2)+AXCI(I,J)*B(J,K+2)+AXCI(I,J+5)*B(J,K
000577       26 AXCIXB(I,K)=AXCIXB(I,K)+AXCI(I,J)*B(J,K)-AXCI(I,J+5)*B(J,K+2)
000620          DO 27 I=1,2
```

```
000621          DO 27 J=1,2
000622          D(I,J+2)=COVARS(I,J+7)-AXCIXB(I,J+2)
000630       27 D(I,J)=COVARS(I,J)-AXCIXB(I,J)
000641          DIVISR=D(1,1)*D(2,2)
000643          IF(DIVISR) 28,28,29
000644       28 DIVISR=10.0**36
000650          GO TO 30
000651       29 DIVISR=SQRT(DIVISR)
000654       30 CORELN(L,M,KK)=D(1,2)/DIVISR
000663          NRSRML=NRSERS-L
000665          LPM=L+M
000667          CORELN(NRSRML,LPM,KK)=D(1,4)/DIVISR
000675          ADDITN=PI
000676          IF(CORELN(L,M,KK)) 32,32,31
000704       31 ADDITN=ADDITN-SIGN(PI,CORELN(NRSRML,LPM,KK))
000715       32 CORESP(L,M,KK)=ATAN(CORELN(NRSRML,LPM,KK)/CORELN(L,M,KK))+ADDIT
000740       33 CORESP(NRSRML,LPM,KK)=CORELN(L,M,KK)**2+CORELN(NRSRML,LPM,KK)**
000756       34 DO 35 I=1,NRSERS
000760          COVARS(NSP1,I)=COVARS(2,I)
000765          COVARS(NSP1,I+7)=COVARS(2,I+7)
000767          COVARS(I,NSP1+7)=COVARS(I,9)
000772       35 COVARS(I,NSP1)=COVARS(I,2)
000776          COVARS(NSP1,NSP1)=COVARS(2,2)
001001          COVARS(NSP1,NSP1+7)=COVARS(2,9)
001004          DO 36 I=1,NSP1
001005          DO 36 J=2,NRSERS
001006          COVARS(I,J+7)=COVARS(I,J+8)
001013       36 COVARS(I,J)=COVARS(I,J+1)
001022          DO 37 I=2,NRSERS
001024          DO 37 J=1,NSP1
001025          COVARS(I,J+7)=COVARS(I+1,J+7)
001032       37 COVARS(I,J)=COVARS(I+1,J)
001041          M=M+1
001043          IF(M-NRSRML) 18,18,38
001045       38 DO 39 I=1,NRSERS
001047          COVARS(NSP1,I)=COVARS(1,I)
001054          COVARS(NSP1,I+7)=COVARS(1,I+7)
001056          COVARS(I,NSP1+7)=COVARS(I,8)
001061       39 COVARS(I,NSP1)=COVARS(I,1)
001065          COVARS(NSP1,NSP1)=COVARS(1,1)
001070          COVARS(NSP1,NSP1+7)=COVARS(1,8)
001073          DO 40 I=1,NRSERS
001074          DO 40 J=1,NRSERS
001075          COVARS(I,J+7)=COVARS(I+1,J+8)
001102       40 COVARS(I,J)=COVARS(I+1,J+1)
001111          M=M+1
001112          IF(M-NRSERS) 38,38,41
001114       41 L=L+1
001116          IF(L-NRSERS) 17,42,42
001120       42 KK=KK+1
001122          IF(KK-NRLSP1) 43,43,44
001124       44 WRITE(6,45)  NRSERS
001132       45 FORMAT(/18X,37H MULTIPLE CORRELATION COEFFICIENT OF  I2,44H  TI
             1SERIES AT SUCCESSIVE FREQUENCY-POINTS//)
001132          WRITE(6,46)  (L,L=1,NRSERS)
001144       46 FORMAT(5(5X,1HI,6X,10HTIMESERIESI2))
001144          WRITE(6,47)
001150       47 FORMAT(/1X)
```

```
001150          DO 48 K=1,NRLSP1
001152          KM1=K-1
001154       48 WRITE(6,49)  (KM1,CORELN(NRSERS,L,K),L=1,NRSERS)
001176       49 FORMAT(5(I6,F18.8))
001176          WRITE(6,50)  NRSERS
001203       50 FORMAT(/18X,36H PARTIAL CORRELATION COEFFICIENT OF  I3,45H   TI
             1SERIES AT SUCCESSIVE FREQUENCY-POINTS//)
001203          WRITE(6,51)
001207       51 FORMAT(11X,1HI,11X,9HREAL PART,7X,1HI,11X,9HIMAG PART,7X,1HI,11
             1 9HCOH.SQ(I),7X,1HI,12X,8HPHASE(I),4X)
001207       52 FORMAT(/50X,10HTIMESERIES I3,4H AND I3//)
001207          DO 53 L=1,NSM1
001211          NRSRML=NRSERS-L
001213          DO 53 M=1,NRSRML
001214          LPM=L+M
001216          WRITE(6,52)  L,LPM
001225          DO 53 K=1,NRLSP1
001227          KM1=K-1
001231       53 WRITE(6,54) KM1,CORELN(L,M,K),KM1,CORELN(NRSRML,LPM,K),
             1KM1,CORESP(NRSRML,LPM,K),KM1,CORESP(L,M,K)
001303       54 FORMAT(4X,2(I8,F20.8),2(I8,F20.8))
001303          RETURN
001303          END

SUBPROGRAM LENGTH
023704

UNUSED COMPILER SPACE
034300


                SUBROUTINE INVERT(N,A,SING)
          C     DOUBLE PIVOT PROGRAM FOR MATRIX INVERSION
000006          DIMENSION A(10,10),P(10,10),Q(10,10)
000006          THRES=1.0E-20
000007          SING=0.0
000010          NLESS1=N-1
000011          DO 3 I=1,N
000013          DO 3 J=1,N
000014          IF(I-J) 1,2,1
000015        1 P(I,J)=0.0
000021          Q(I,J)=0.0
000023          GO TO 3
000023        2 P(I,J)=1.0
000027          Q(I,J)=1.0
000032        3 CONTINUE
000037          DO 20 K=1,NLESS1
000040          BIGA=0.0
000041          KPLUS1=K+1
000043          DO 8 I=K,N
000044          DO 8 J=K,N
000046          IF(A(I,J)) 4,5,5
000052        4 ABSA=-A(I,J)
000056          GO TO 6
```

```
000057        5 ABSA=A(I,J)
000063        6 IF(BIGA-ABSA) 7,8,8
000066        7 BIGA=ABSA
000070          LARGJ=J
000071          LARGI=I
000073        8 CONTINUE
000100          IF(LARGJ-K) 25,12,9
000102        9 DO 10 I=K,N
000104          ASTORE=A(I,K)
000110          A(I,K)=A(I,LARGJ)
000115       10 A(I,LARGJ)=ASTORE
000122          DO 11 I=1,N
000123          QSTORE = Q(I,K)
000126          Q(I,K)=Q(I,LARGJ)
000133       11 Q(I,LARGJ)=QSTORE
000140       12 IF(LARGI-K) 25,16,13
000143       13 DO 14 J=K,N
000145          ASTORE=A(K,J)
000151          A(K,J)=A(LARGI,J)
000156       14 A(LARGI,J)=ASTORE
000163          DO 15 J=1,N
000164          PSTORE=P(K,J)
000167          P(K,J)=P(LARGI,J)
000174       15 P(LARGI,J)=PSTORE
000200       16 AMAG=ABS(A(K,K))
000204          IF(AMAG-THRES) 24,24,17
000207       17 DO 19 I=K,NLESS1
000211          RMPY=A(I+1,K)/A(K,K)
000217          DO 18 L=KPLUS1,N
000221       18 A(I+1,L)=A(I+1,L)-RMPY*A(K,L)
000234          A(I+1,K)=0.0
000240          DO 19 LL=1,N
000241       19 P(I+1,LL)=P(I+1,LL)-RMPY*P(K,LL)
000256          DO 20 J=K,NLESS1
000260          CMPY=A(K,J+1)/A(K,K)
000266          A(K,J+1)=0.0
000271          DO 20 L=1,N
000272   20     Q(L,J+1)=Q(L,J+1)-CMPY*Q(L,K)
000311          AMAG=ABS(A(N,N))
000315          IF(AMAG-THRES) 24,24,21
000317       21 DO 22 J=1,N
000321          DO 22 I=1,N
000322       22 Q(I,J)=Q(I,J)/A(J,J)
000335          DO 23 I=1,N
000336          DO 23 J=1,N
000337          A(I,J)=0.0
000342          DO 23 L=1,N
000344       23 A(I,J)=A(I,J)+Q(I,L)*P(L,J)
000364          GO TO 25
000364       24 SING=1.0
000365       25 RETURN
000366          END
```

SUBPROGRAM LENGTH
000735

```
UNUSED COMPILER SPACE
037500

                SUBROUTINE  DLAG(WCO,WQUAD,SPECX)
000006          COMPLEX  BHAT,SUM,BETA,XYFW
000006          DIMENSION  WCO(100), WQUAD(100), BHAT(100),BETA(20),XYFW(100),S
               1X(100)
000006          LAG1=11.0
         CCCCC  NUMBER  LAGS  =  LAG1-1
000007          SUM=0.0
000011          M=72.0
000013          M1=M+1
000015          DO 101  I=1,M1
000017   101    XYFW(I)=CMPLX(WCO(I),-WQUAD(I))
000031          DO 102  I=1,M1
000032   102    BHAT(I)=XYFW(I)/SPECX(I)
000044          PI=3.14159265359
000045          DO 103 J=1,LAG1
000047          DO 104 K=1,M1
000050          OMEGA=(PI*(K-1)/M)*(J-1)
000060   104    SUM=SUM+BHAT(K)*CMPLX(COS(OMEGA),SIN(OMEGA))
000105          M2=M-1
000107          DO 105 K=2,M2
000111          OMEGA=-(PI*(K-1)/M)*(J-1)
000121   105    SUM=SUM+(CONJG(BHAT(K)))*(CMPLX(COS(OMEGA),SIN(OMEGA)))
000150   103    BETA(J)=SUM/(2.0*M)
000163          WRITE(6,2)
000167   2      FORMAT(//4X,38HBETA  ESTIMATES  FOR  DISTRIBUTED  LAG)
000167          WRITE(6,3)(BETA(I),I=1,LAG1)
000202   3      FORMAT(2F20.8)
000202          RETURN
000203          END

SUBPROGRAM LENGTH
001153

UNUSED COMPILER SPACE
040400
```

APPENDIX 2

Sequential L-P Program

a) Mace Control Cards

b) Program Delta

```
XKA0399.   09/16/71.PURDUE MACE 71/09/12.

10.42.28.XKA03,    14286,CANDLER,T40,CM76000,P5,L
10.42.28.5000.
10.42.29.COPYBR(INPUT,TAPE11)
10.42.29.CX       .001 SEC., NL  13000 WORDS
10.42.29.COPYBR(INPUT,TAPE12,11)
10.42.30.REWIND(TAPE11,TAPE12)
10.42.35.CX       .028 SEC., NL  76000 WORDS
10.42.35.MAP(PART)
10.42.35.FUN(S,,,,,KGO)
10.42.37. CTIME 001.041 SEC. FUN MOD LEVEL 60F
10.42.38.LIBCOPY(CSCBIN,LGO,RADESLP)
10.42.39.LGO(TAPE11,TAPE13)
10.42.41.CX      2.246 SEC., NL  22200 WORDS
10.42.41.CX      2.256 SEC., NL  22400 WORDS
10.42.41.CX      2.257 SEC., NL  22700 WORDS
10.42.42.STOP
10.42.42.REWIND(TAPE13)
10.42.42.COPYSBF(TAPE13,OUTPUT)
10.42.43.RFL,50000.
10.42.43.CX      2.905 SEC., NL  50000 WORDS
10.42.43.KGO.
10.42.44.CX      3.678 SEC., NL  15100 WORDS
10.42.45.STOP
10.42.45.REWIND(TAPE11,TAPE13)
10.42.46.COPYBF(TAPE15,TAPE11)
10.42.46.REWIND(TAPE11,TAPE15)
10.42.46.RFL,31000.
10.42.47.CX      4.174 SEC., NL  31000 WORDS
10.42.47.LGO(TAPE11,TAPE13)
10.42.49.CX      5.116 SEC., NL  22200 WORDS
10.42.49.CX      5.126 SEC., NL  22400 WORDS
10.42.49.CX      5.128 SEC., NL  22700 WORDS
10.42.50.STOP
10.42.50.REWIND(TAPE13)
10.42.50.COPYSBF(TAPE13,OUTPUT)
10.42.51.RFL,50000.
10.42.51.CX      5.770 SEC., NL  50000 WORDS
10.42.51.KGO.
10.42.52.CX      6.568 SEC., NL  15100 WORDS
10.42.54.STOP
10.42.54.REWIND(TAPE11,TAPE13)
10.42.54.COPYBF(TAPE15,TAPE11)
10.42.55.REWIND(TAPE11,TAPE15)
10.42.55.RFL,31000.
10.42.55.CX      7.081 SEC., NL  31000 WORDS
10.42.55.LGO(TAPE11,TAPE13)
10.42.58.CX      8.109 SEC., NL  22200 WORDS
10.42.58.CX      8.119 SEC., NL  22400 WORDS
10.42.58.CX      8.120 SEC., NL  22700 WORDS
10.42.58.STOP
10.42.58.REWIND(TAPE13)
10.42.59.COPYSBF(TAPE13,OUTPUT)
10.42.59.RFL,50000.
10.43.00.CX      8.753 SEC., NL  50000 WORDS
10.43.00.KGO.
10.43.01.CX      9.543 SEC., NL  15100 WORDS
10.43.02.STOP
10.43.03.REWIND(TAPE11,TAPE13)
10.43.03.COPYBF(TAPE15,TAPE11)
10.43.03.REWIND(TAPE11,TAPE15)
```

```
10.43.03.RFU,31000.
10.43.03.CX     10.058 SEC., NL  31000 WORDS
10.43.03.LGO(TARB11,TAPE13)
10.43.06.CX     11.195 SEC., NL  22200 WORDS
10.43.06.CX     11.206 SEC., NL  22400 WORDS
10.43.06.CX     11.207 SEC., NL  22700 WORDS
10.43.06.STOP
10.43.07.REWIND(TARE13)
10.43.08.COPYSBF(TAPB13,OUTPUT)
10.43.09.RFU,50000.
10.43.11.CX     11.869 SEC., NL  50000 WORDS
10.43.11.KGO.
10.43.12.CX     12.507 SEC., NL  15100 WORDS
10.43.14.STOP
10.43.14.REWIND(TARE11,TAPE13)
10.43.15.COPYBF(TARE15,TAPE11)
10.43.15.REWIND(TARE11,TAPE15)
10.43.15.RFL,31000.
10.43.15.CX     13.044 SEC., NL  31000 WORDS
10.43.16.LGO(TARB11,TAPE13)
10.43.17.CX     14.188 SEC., NL  22200 WORDS
10.43.18.CX     14.200 SEC., NL  22400 WORDS
10.43.18.CX     14.202 SEC., NL  22700 WORDS
10.43.18.STOP
10.43.18.REWIND(TAPE13)
10.43.18.COPYSBF(TAPB13,OUTPUT)
10.43.19.RFL,50000.
10.43.19.CX     14.816 SEC., NL  50000 WORDS
10.43.19.KGO.
10.43.20.CX     15.591 SEC., NL  15100 WORDS
10.43.22.STOP
10.43.22.REWIND(TARE11,TAPE13)
10.43.22.COPYBF(TARE15,TAPE11)
10.43.22.REWIND(TAPE11,TAPE15)
10.43.22.RFU,31000.
10.43.22.CX     16.103 SEC., NL  31000 WORDS
10.43.23.LGO(TARB11,TAPE13)
10.43.24.CX     17.266 SEC., NL  22200 WORDS
10.43.24.CX     17.275 SEC., NL  22400 WORDS
10.43.24.CX     17.277 SEC., NL  22700 WORDS
10.43.25.STOP
10.43.25.REWIND(TAPE13)
10.43.25.COPYSBF(TAPB13,OUTPUT)
10.43.26.RFU,50000.
10.43.26.CX     17.905 SEC., NL  50000 WORDS
10.43.26.KGO.
10.43.27.CX     18.590 SEC., NL  15100 WORDS
10.43.28.STOP
10.43.28.REWIND(TARE11,TAPE13)
10.43.28.COPYBF(TARE15,TAPE11)
10.43.29.REWIND(TAPE11,TAPE15)
10.43.29.RFU,31000.
10.43.29.CX     19.095 SEC., NL  31000 WORDS
10.43.29.LGO(TARE11,TAPE13)
10.43.31.CX     20.130 SEC., NL  22200 WORDS
10.43.31.CX     20.142 SEC., NL  22400 WORDS
10.43.31.CX     20.143 SEC., NL  22700 WORDS
10.43.32.STOP
10.43.32.REWIND(TARE13)
10.43.32.COPYSBF(TAPE13,OUTPUT)
10.43.33.RFU,50000.
10.43.33.CX     20.789 SEC., NL  50000 WORDS
10.43.34.KGO.
```

```
10.43.36.STOR
10.43.36.REWIND(TAPE11,TAPE13)
10.43.36.COPYBF(TAPE15,TAPE11)
10.43.36.REWIND(TAPE11,TAPE15)
10.43.36.RFL,31000.
10.43.36.CX     22.024 SEC., NL  31000 WORDS
10.43.37.LGO(TARB11,TAPE13)
10.43.38.CX     23.076 SEC., NL  22200 WORDS
10.43.38.CX     23.088 SEC., NL  22400 WORDS
10.43.38.CX     23.090 SEC., NL  22700 WORDS
10.43.39.STOR
10.43.39.REWIND(TAPE13)
10.43.39.COPYSBF(TAPB13,OUTPUT)
10.43.40.RFL,50000.
10.43.40.CX     23.724 SEC., NL  50000 WORDS
10.43.40.KGO.
10.43.41.CX     24.469 SEC., NL  15100 WORDS
10.43.42.STOR
10.43.42.REWIND(TAPE11,TAPE13)
10.43.42.COPYBF(TAPE15,TAPE11)
10.43.43.REWIND(TAPE11,TAPE15)
10.43.43.RFL,31000.
10.43.43.CX     24.980 SEC., NL  31000 WORDS
10.43.43.LGO(TARB11,TAPE13)
10.43.44.CX     25.928 SEC., NL  22200 WORDS
10.43.44.CX     25.944 SEC., NL  22400 WORDS
10.43.44.CX     25.945 SEC., NL  22700 WORDS
10.43.45.STOR
10.43.45.REWIND(TAPE13)
10.43.45.COPYSBF(TAPB13,OUTPUT)
10.43.46.RFL,50000.
10.43.46.CX     26.565 SEC., NL  50000 WORDS
10.43.46.KGO.
10.43.47.CX     27.278 SEC., NL  15100 WORDS
10.43.48.STOP
10.43.48.REWIND(TAPE11,TAPE13)
10.43.48.COPYBF(TAPE15,TAPE11)
10.43.48.REWIND(TAPE11,TAPE15)
10.43.48.RFL,31000.
10.43.48.CX     27.783 SEC., NL  31000 WORDS
10.43.49.LGO(TARB11,TAPE13)
10.43.50.CX     28.736 SEC., NL  22200 WORDS
10.43.50.CX     28.745 SEC., NL  22400 WORDS
10.43.50.CX     28.746 SEC., NL  22700 WORDS
10.43.51.STOR
10.43.51.REWIND(TAPE13)
10.43.51.COPYSBF(TAPB13,OUTPUT)
10.43.52.RFL,50000.
10.43.52.CX     29.379 SEC., NL  50000 WORDS
10.43.52.KGO.
10.43.53.CX     30.084 SEC., NL  15100 WORDS
10.43.54.STOR
10.43.54.REWIND(TAPE11,TAPB13)
10.43.54.COPYBF(TAPE15,TAPE11)
10.43.54.REWIND(TAPE11,TAPE15)
10.43.54.RFL,31000.
10.43.54.CX     30.587 SEC., NL  31000 WORDS
10.43.55.LGO(TARB11,TAPE13)
10.43.56.CX     31.578 SEC., NL  22200 WORDS
10.43.56.CX     31.589 SEC., NL  22400 WORDS
10.43.56.CX     31.590 SEC., NL  22700 WORDS
10.43.57.STOR
10.43.57.REWIND(TAPE13)
```

```
10.43.57.COPYSBF(TAPE13,OUTPUT)
10.43.57.RFU,50000.
10.43.57.CX     32.215 SEC., NL   50000 WORDS
10.43.57.KGO.
10.43.59.CX     32.942 SEC., NL   15100 WORDS
10.43.59.STOP
10.44.00.REWIND(TAPE11,TAPE13)
10.44.00.COPYBF(TAPE15,TAPE11)
10.44.00.REWIND(TAPE11,TAPE15)
10.44.00.CONTINUE.
10.44.00.REWIND(TAPE14)
10.44.00.COPYSBF(TAPE14,OUTPUT,3)
10.44.01.EXIT.
10.44.01.CP    33.457 SEC., IO       3777 UNITS.
10.44.01.LINES    3227
10.44.01.CM    12.226 MWD-SEC.,  RL   15100 WORDS
```

```
                PROGRAM DELTA(INPUT,OUTPUT,TAPE11,TAPE12,TAPE13,TAPE15,TAP
               1UT)
000002          DIMENSION CC(3),NAMES(3),DC(7),DB(6),TEMP(3)
000002          INTEGER TEST
000002          DATA(NAMES(I),I=1,3)/3HT12,3HT23,3HT34/
000002          NOLD=11
000003          NDEL=12
000004          NLP=13
000005          NOUT=14
000006          NNEW=15
000007          REWIND NOLD
000011          REWIND NLP
000013          DO 100 I=1,3
000020    100   CC(I)=0.0
000022          DO 8 I=1,18
000023    8     READ(NLP,9)BOSS
000032    9     FORMAT(A70)
000032          DO 120 I=1,9
000034          READ(NLP,10) TEST,WORK
000043          DO 110 J=1,3
000051    110   IF(TEST.EQ.NAMES(J)) CC(J)=WORK
000054    10    FORMAT(7X,A3,3X,F20.5)
000054          WRITE(NOUT,9000)TEST,WORK
000064    9000  FORMAT(2X,5HTEST=,A4,5X,6HWORK= ,F12.4)
000064    120   CONTINUE
000066          WRITE(NOUT,3001)(CC(J),J=1,3)
0C0074    3001  FORMAT(3X,3F12.0)
000074          READ(NDEL,11)ID
000102    11    FORMAT(I5)
000102          READ(NDEL,34)(DC(I),I=1,7)
000110          WRITE(NOUT,9001)(DC(I),I=1,7)
000116    9001  FORMAT(7F10.2)
000116          READ(NDEL,12)(DB(I),I=1,6)
000124          WRITE(NOUT,9002)(DB(I),I=1,6)
000132    9002  FORMAT(6F12.2)
000132    12    FORMAT(6F6.0)
000132    34    FORMAT(7F5.0)
000132          DO 111 I=1,3
000137    111   TEMP(I) =0.0
000141          DO 130 J=1,16
000142          READ(NOLD,13)(TEMP(I),I=1,3)
000147    13    FORMAT(3A10)
000147    130   WRITE(NNEW,13)(TEMP(I),I=1,3)
000157          WRITE(NNEW,14)(DC(1))
000165    14    FORMAT(6X,4HS1T1,2X,4HCOST,3X,F5.2)
000165          READ(NOLD,13)(TEMP(I),I=1,3)
000173          DO 140 J=1,3
000175          READ(NOLD,13)(TEMP(I),I=1,3)
000202    140   WRITE(NNEW,13)(TEMP(I),I=1,3)
000212          WRITE(NNEW,15)(DC(2))
000220    15    FORMAT(6X,4HS2T1,2X,4HCOST,3X,F5.2)
000220          READ(NOLD,13)(TEMP(I),I=1,3)
000226          DO 150 J=1,3
000230          READ(NOLD,13)(TEMP(I),I=1,3)
000235    150   WRITE(NNEW,13)(TEMP(I),I=1,3)
000245          WRITE(NNEW,16)(DC(3))
000253    16    FORMAT(6X,4HS3T1,2X,4HCOST,3X,F5.2)
000253          READ(NOLD,13)(TEMP(I),I=1,3)
```

```
000261          DO 160 J=1,3
000263          READ(NOLD,13)(TEMP(I),I=1,3)
000270   160    WRITE(NNEW,13)(TEMP(I),I=1,3)
000300          WRITE(NNEW,17)(DC(4))
000306   17     FORMAT(6X,4HS4T1,2X,4HCOST,3X,F5.2)
000306          READ(NOLD,13)(TEMP(I),I=1,3)
000314          DO 170 J=1,3
000316          READ(NOLD,13)(TEMP(I),I=1,3)
000323   170    WRITE(NNEW,13)(TEMP(I),I=1,3)
000333          WRITE(NNEW,18)(DC(5))
000341   18     FORMAT(6X,3HT12,3X,4HCOST,3X,F5.2)
000341          READ(NOLD,13)(TEMP(I),I=1,3)
000347          READ(NOLD,13)(TEMP(I),I=1,3)
000355          WRITE(NNEW,13)(TEMP(I),I=1,3)
000363          WRITE(NNEW,19)(DC(6))
000371   19     FORMAT(6X,3HT23,3X,4HCOST,3X,F5.2)
000371          READ(NOLD,13)(TEMP(I),I=1,3)
000377          READ(NOLD,13)(TEMP(I),I=1,3)
000405          WRITE(NNEW,13)(TEMP(I),I=1,3)
000413          WRITE(NNEW,20)(DC(7))
000421   20     FORMAT(6X,3HT34,3X,4HCOST,3X,F5.2)
000421          READ(NOLD,13)(TEMP(I),I=1,3)
000427          DO 180 J=1,18
000431          READ(NOLD,13)(TEMP(I),I=1,3)
000436   180    WRITE(NNEW,13)(TEMP(I),I=1,3)
000446          WRITE(NNEW,21)(DB(1))
000454   21     FORMAT(12X,4HT1G1,3X,F6.1)
000454          READ(NOLD,13)(TEMP(I),I=1,3)
000462          WRITE(NNEW,22)(CC(1))
000470   22     FORMAT(12X,4HT1G2,3X,F6.1)
000470          WRITE(NNEW,23)(CC(2))
000476   23     FORMAT(12X,4HT1G3,3X,F6.1)
000476          WRITE(NNEW,24)(CC(3))
000504   24     FORMAT(12X,4HT1G4,3X,F6.1)
000504          WRITE(NNEW,25)(DB(2))
000512   25     FORMAT(12X,4HT1TO,3X,F6.1)
000512          WRITE(NNEW,26)(DB(3))
000520   26     FORMAT(12X,4HMIN1,3X,F6.1)
000520          WRITE(NNEW,27)(DB(4))
000526   27     FORMAT(12X,4HMIN2,3X,F6.1)
000526          WRITE(NNEW,28)(DB(5))
000534   28     FORMAT(12X,4HMIN3,3X,F6.1)
000534          WRITE(NNEW,29)(DB(6))
000542   29     FORMAT(12X,4HMIN4,3X,F6.1)
000542          DO 210 J=1,8
000544   210    READ(NOLD,13)(TEMP(I),I=1,3)
000553          DO 200 J=1,3
000555          READ(NOLD,13)(TEMP(I),I=1,3)
000562   200    WRITE(NNEW,13)(TEMP(I),I=1,3)
000572          REWIND NOLD
000574          REWIND NNEW
000576          DO 190 J=1,67
000600          READ(NNEW,13)(TEMP(I),I=1,3)
000605   190    WRITE(NOUT,13)(TEMP(I),I=1,3)
000615          REWIND NNEW
000617          STOP
000621          END
```