CHAPTER THIRTEEN

Refinery Linear Programming Modeling

OVERVIEW

The basic problem of linear programming (LP) is to maximize or minimize a function of several variables subject to a number of constraints. The functions being optimized and the constraints are linear. General linear programming deals with allocation of resources, seeking their optimization. In the context of an oil refinery, an LP model is a mathematical model of the refinery, simulating all refinery unit yields, unit capacities, utility consumption, and the like as well as product blending operations of the refinery by means of linear equations, each equation subject to a number of constraints. These equations are compiled in a matrix of rows and columns, the columns representing the unknowns or variables and the rows or equations representing the relations between variables. The values in the matrix are simply the coefficients that apply to unknowns in each equation. As the number of unknowns are more than the number of constraints relating them, a large number of solutions might satisfy all the problem parameters.

The optimal solution must be chosen from the set of only those solutions that satisfy all the problem parameters and, at the same time, maximize refinery profit or minimize operating cost. To aid the search for an optimum solution, LP is driven by a row in the matrix containing cost and revenue (the objective function row).

DEVELOPMENT OF THE REFINERY LP MODEL

In the oil industry, prior to the advent of LP techniques, all optimization studies were done by calculating several hand balances, moving toward

an optimal solution by trial and error. Carrying out simplex procedure by hand was very tedious and time consuming. The typical refinery LP model used for planning has approximately 300–500 equations and 800–1500 activities to optimize. With a simplex algorithm available as a computer program, interest quickly developed in optimizing via a linear programming. In the 1950s, a standard input format to describe a matrix was agreed on, opening the market to LP software from different vendors. These software are generally of two types:

1. Programs in which the user enters all refinery data, such as unit yields, product properties, and unit capacities, in the form of spreadsheets that can easily be updated. These programs convert the data tables into matrix form by using special programming languages (Omni, Magan, etc.), thus saving many hours in producing matrix input correctly. These programs are called *matrix generators*. These programming languages can also be used to create a "report writer" program to print out the optimized results.
2. Optimizer programs[1] that read the matrix description in the standardized input format, optimize the problem, and report the results in an "unscrambled report," which is simply a list of rows and columns and the associated optimized values.

A refinery LP model is designed to model a wide variety of activities, including, among others, the following:

Distillation of crudes.
Downstream processing units, such as cat reformers, hydrocrackers, desulfurizers, and visbreakers in various processing modes.
Pooling of streams.
Recursing on the assumed qualities of a rundown tank's content.
Finished product blending.
Refinery fuel blending.
Importing feedstocks to meet product demand.
Exporting surplus refinery streams to other refineries.

The refinery LP model, in fact, is simply a set of data tables in the form of spreadsheets that are converted into a matrix using special programming languages. As many solutions to the problem are possible, the criteria for choosing an optimum solution is that which, apart from satisfying all equations, gives maximum profit to the refinery.

The optimum solution of a refinery LP model yields the following:

A complete, unitwise, material balance of all refinery units. The material balance could be on a volume or weight basis.

The unit capacities available and utilized.

Feedstocks available and used for processing or blending.

Utilities (fuel, electricity, steam, cooling water), chemical, and catalyst consumption for all processing units and the overall refinery.

Blend composition of all products and the properties of blended products.

An economic summary, which may include the cost of crude, other feedstocks, utilities, chemicals, and catalyst consumed and the prices of blended finished product.

THE STRUCTURE OF A REFINERY LP MODEL

ROW AND COLUMNS NAMES AND TYPES

Row and column naming conventions are followed for easy identification and manipulation of data by different vendors of LP software. (The row and column naming and also data tables naming convention followed here for rows, columns, and data tables is from the popular Process Industries Modeling System (PIMS), a PC-based refinery LP package.[2] Here, row and column names are seven characters long: The first four characters generally identify the type of the row or column, while the last three characters identify the stream.

Row Names

ROW	CODE	EXAMPLE
OBJECTIVE FUNCTION ROW	OBJFN	
MATERIAL BALANCE (VOLUME BASIS)	VBALxxx	VBALNAP,VBALKER
MATERIAL BALANCE (MASS BASIS)	WBALxxx	WBALRES
UTILITY BALANCE	UBALxxx	UBALKWH,UBALFUL
PROCESS UNIT CAPACITY	CCAPxxx	CCAPFCC
VOLUME BLEND BALANCE	EVBLxxx	EVBL888
MASS BLEND BALANCE	EWBLxxx	EWBL440
MINIMUM BLEND SPECIFICATION	Nssxxx	NSPG395
MAXIMUM BLEND SPECIFICATION	Xssxxx	XRON395
USER-DEFINED EQUALITY ROW	Eabcxyz	ECHGFCU
USER-DEFINED "LESS THAN OR EQUAL TO ROW"	Labcxyz	
USER-DEFINED "GREATER THAN OR EQUAL TO ROW"	Gabcxyz	
RECURSION MATERIAL BALANCE ROW	RBALxxx	RBAL397
RECURSION PROPERTY BALANCE	Rprpxxx	RRON397

Column Names

COLUMN	CODE	EXAMPLE
PURCHASE 1000 UNITS OF MATERIAL OR UTILITY	PURCxxx	PURCABP
SELL 1000 UNITS OF MATERIAL OR PRODUCT	SELLxxx	SELLNAP
SPECIFICATION VOLUME BLEND 1000 UNITS OF PRODUCT, prd	BVBLprd	BVBL888
SPECIFICATION MASS BLEND 1000 UNITS OF PRODUCT, prd	BWBLprd	BWBL961
OPERATE 1000 UNITS OF PROCESS UNIT "PRS" IN MODE mmm	Sprsmmm	SFCUFC1

UNITS OF A ROW

The units of a column multiplied by the units of a coefficient in a row equal the units of a row. Thus, if the column activity represents thousands of tons per day and a coefficient in the utility balance row has unit of 000' Btu/ton, the units of row are

$$= 000' \text{ Tons/Day} * 000' \text{ Btu/Ton or million Btu/Day}$$

The types of restrictions encountered in a refinery LP model are as follows: feed availability; product demand; process yield; utility, catalyst, and chemical consumption; and product blending.

Feed Availability

These equations reflect the purchase of feedstock, such as crude and imported intermediate streams or blend stocks. The disposition of crudes could be to various crude units in different operation modes. The disposition of intermediate feedstock could be to a secondary processing unit or to product blending.

Table 13-1 (BUY Table) shows a typical data entry and Table 13-2 shows the matrix generated by the feed availability constraints. The row names are derived from the row names of the BUY Table. For example, row ABP in Table 13-1 generates a row name VBALABP in the matrix.

Table 13-1
BUY Table

TABLE BUY	TEXT	MIN	MAX	FIX	COST
	Crude Oil Imports				
ABP	AL CRUDE			165.1842	16.809
BAH	BAHRAIN CRUDE			42.000	15.109
	TOTAL FEED			207.1842	

In row VBALABP, column PURCABP has a negative coefficient, reflecting that the Arab crude purchased is made equal to its disposition to various crude units.

Product Demand

These equations relate the product demand to its blending. The product demand is inserted in the SELL Table (Table 13-3). The row names in this table are product codes. These rows generate a matrix (Table 13-4) by adding the prefix VBAL to the table row name, denoting that each row is, in fact, a material balance row relating the demand to its blending. For example, row 150 in the SELL Table generates the row VBAL150 (the material balance row for LPG).

Row VBAL150 has coefficients 1 and –1, respectively, for its two columns, SELL150 and BVBL150, indicating that variable SELL150 is made equal to BVBL150 or volume of LPG blended in product LPG.

If the demand for a product is fixed, this is called a *fixed-grade product*, and the LP solution meets this demand by fixing the value of the variable SELL150. If demand for a product to be produced is not fixed, it is called a *balancing or free grade*, and its production is optimized, based on the price of the product, within maximum or minimum demand constraints, if these exist. It is, therefore, necessary to insert the prices of balancing-grade products whose production is to be optimized.

Column names in Table SELL are generated from the table name and row names. For example, in Table 13-3 (Table SELL), row 150 generates a column or variable SELL150, whose production is optimized on the basis of its unit price, indicated in column price. Variable SELL150 has a value within the minimum and maximum constraints shown in this table.

Table 13-2
Matrix for Table 13-1

ROW	VBALABP	TYPE LE
	PURCABP	−1
	SCRDA1N	1
	SCRDA1K	1
	SCRDA2N	1
	SCRDA2K	1
	SCRDA3N	1
	SCRDA3K	1
	SCRDA4N	1
	SCRDA4K	1
	SCRDA5N	1
	SCRDA5K	1
ROW	**VBALBAH**	**TYPE LE**
	PURCBAH	−1
	SCRDB1N	1
	SCRDB1K	1
	SCRDB2N	1
	SCRDB2K	1
	SCRDB3N	1
	SCRDB3K	1
	SCRDB4N	1
	SCRDB4K	1
	SCRDB5N	1
	SCRDB5K	1
ROW	**MINOBJ**	**TYPE FR**
	PURCABP	−16.809
	PURBAH	−15.109
COLUMNS		**TYPE FIX**
	PURCABP	165.184
	PURCBAH	42.000

NOTE: SCRDA1N, SCRDA1K, ETC. ARE VECTORS INDICATING DISPOSITION OF ARAB CRUDE TO VARIOUS CRUDE UNITS AND MODES.

Process Yields

A refinery has a large number of process units. The most important units are the crude distillation unit (CDU), consisting of atmospheric distillation of crude and vacuum distillation of atmospheric resid. Downstream of the CDU are a number of other processing units, such as the

Table 13-3
SELL Table

TABLE SELL	TEXT		MIN	MAX	FIX	PRICE
	Finished Products					
150	I-150 LPG	BAL	0.000			8.495
220	I-220 LSR	BAL	0.000			12.059
210	I-210 WSR	BAL	0.000			12.454
97L	I-397L .4g/l		0.000	0.000		17.657
440	I-440 JET A1 FUEL	BAL	0.000			18.168
76Z	I-876ZP DIESEL G.OIL		0.000	0.000	7.737	17.003
888	I-888 DIESEL G.O	BAL	0.000			15.489
928	I-928 FUEL OIL		0.000	0.000	13.455	9.431
961	I-961	BAL	0.000			8.948
ASP	ASPHALT		0.000	0.000	2.128	30.000
SUL	SULFUR		0.000			25.000
	TOTAL FINISHED PRODUCT		0.000	0.00	23.320	

fluid cat cracker (FCCU), distillate hydrocracker, diesel desulfurizer, and cat reformer unit.

The basic structure of all process submodels are as follows. A process unit may operate in a number of modes. Each operating mode becomes a column in the process submodel. The column names may be feed names or processing modes of a unit. The rows are, in fact, material balance rows for each product produced as a result of processing. The coefficients are the yield of the product in the mode represented by column heading. Columns can be compared to pipes in a plant through which material flows and the various rows, to taps from which products are drawn off.

Tables 13-5 and 13-7 show the typical data input for two process units, FCCU and VBU. Tables 13-6 and 13-9 present the matrix generated by these tables. For example, referring to Table 13-5, the FCCU submodel, the column headings FC1 and FC2 indicate operating modes of the FCCU.

Rows VBALC4U, VBALPOR, VBALLCN, VBALMCN, VBALLCO, VBALHCO, and so forth indicate material balances for the yields of cracked LPG, light cat naphtha, medium cat naphtha, light cycle gas oil, heavy cycle gas oil, and the like, which are produced in the FCCU and disposed of somewhere else.

Table 13-4
Matrix for Table 13-3

ROW	**VBAL150**	**TYPE LE**
	SELL150	1
	BVBL150	−1
ROW	**VBAL220**	**TYPE LE**
	SELL210	1
	BVBL210	−1
ROW	**VBAL210**	**TYPE LE**
	SELL210	1
	BVBL210	−1
ROW	**VBAL97L**	**TYPE LE**
	SELL97L	1
	BVBL150	−1
ROW	**VBAL440**	**TYPE LE**
	SELL440	1
	BVBL440	−1
ROW	**VBAL76Z**	**TYPE LE**
	SELL76Z	1
	BVBL76Z	−1
ROW	**VBAL888**	**TYPE LE**
	SELL888	1
	BVBL888	−1
ROW	**VBAL928**	**TYPE LE**
	SELL928	1
	BVBL928	−1
ROW	**VBAL961**	**TYPE LE**
	SELL961	1
	BVBL961	−1
ROW	**VBALASP**	**TYPE LE**
	SELLASP	1
	BVBLASP	−1
ROW	**VBALSUL**	**TYPE LE**
	SELLSUL	1
	BVBLSUL	−1
ROW	**MINOBJ**	**TYPE FR**
	SELL150	8.495
	SELL220	12.059
	SELL210	12.454
	SELL97L	17.657
	SELL440	18.168
	SELL76Z	17.003
	SELL888	15.489
	SELL928	9.431
	SELL961	8.948
	SELLASP	30.000
	SELLSUL	25.000

Table 13-5
FCCU Submodel

TABLE SFCU	TEXT	FC1 HMI	FC2 HVGO
	Feed streams to FCCU		
EBALFC1	Isomate feed	1	
EBALFC2	HVGO feed		1
VBALC4U	UNSAT BUTANES	−0.0322	−0.0196
VBALPOR	RERUN POLYMER	−0.0381	−0.0337
VBALLCN	LIGHT CAT NAPHTHA	−0.2990	−0.2160
VBALMCN	MEDIUM CAT NAPHTHA	−0.0670	−0.0850
VBALHCN	HEAVY CAT NAPHTHA	−0.1150	−0.1250
VBALLCO	LIGHT CYCLE GAS OIL	−0.2600	−0.2080
VBALHDO	HEAVY/DECANT OIL	−0.2170	−0.3140
	BALANCE CHECK	−1.0283	−1.0013
UBALFUL	REBOILER FUEL	0.2524	0.2524
UBALCCC	CAT $ CHEM. $/BBL	0.0534	0.0534
CCAPFCU	FEED LIMITATION	1	1

The coefficient at a column/row intersection is the yield of a product represented by row name from operation in mode indicated by column name.

Utility, Catalyst, and Chemical Consumption

Similar to VBAL rows, every process submodel contains a number of UBAL or utility balance rows. The coefficients of these rows in any column indicate the utility, chemical, and catalyst consumption per unit feed processed in the mode indicated by column heading.

For material and utility balance rows, by convention, all feeds are shown with positive sign and all products with negative sign.

User-Defined Rows

In Table 13-5, EBALFC1 and EBALFC2 are user-defined equality rows, which make the disposition of the two FCCU feeds equal to their production.

Table 13-6
Matrix for Table 13-5

ROW	**EBALFC1**	**TYPE EQ**		
	SFCUHM1	−1		
	SFCUFC1	1		
ROW	**EBALFC2**	**TYPE EQ**		
	SFCUHV2	−1		
	SFCUFC2	1		
ROW	**VBALC4U**	**TYPE EQ**		
	SFCUFC1	−0.0322		
	SFCUFC2	−0.0196		
ROW	**VBALPOR**	**TYPE EQ**		
	SFCUFC1	−0.0381		
	SFCUFC2	−0.0337		
ROW	**VBALLCN**	**TYPE EQ**		
	SFCUFC1	−0.299		
	SFCUFC2	−0.216		
ROW	**VBALMCN**	**TYPE EQ**		
	SFCUFC1	−0.067		
	SFCUFC2	−0.085		
ROW	**VBALHCN**	**TYPE EQ**		
	SFCUFC1	−0.115		
	SFCUFC2	−0.125		
ROW	**VBALLCO**	**TYPE EQ**		
	SFCUFC1	−0.260		
	SFCUFC2	−0.208		
ROW	**VBALHDO**	**TYPE EQ**		
	SFCUFC1	−0.217		
	SFCUFC2	−0.314		
ROW	**UBALFUL**	**TYPE EQ**		
	SCFUFC1	0.2524		
	SCFUFC2	0.2524		
ROW	**UBALCCC**	**TYPE EQ**		
	SCFUFC1	0.0534		
	SCFUFC2	0.0534		
ROW	**CCAPFCU**	**TYPE LE**	RHS	38.556
	SFCUFC1	1		
	SFCUFC2	1		

Process Unit Capacities

We see that every process unit submodel has a capacity row. The various operation modes of the process have a coefficient in this row, showing capacity consumed for processing one unit of feed. Thus, refer-

Table 13-7
VBU Submodel

TABLE SVBU	TEXT	BAS
VBALVBP	VBU FEED POOL	1
VBALGA1	OFF GASES	−0.036
VBALNVB	VBU NAPHTHA	−0.049
VBALVRS	VISBROKEN RESID	−0.951
	BALANCE CHECK	−0.036
UBALFUL	FUEL REQ. MSCF	0.3015
CCAPVBU	VBU CAPACITY	1
CCAPVBX	MAX VISBREAKING	1

Table 13-8
Matrix for Table 13-7

ROW	VBALVBP	TYPE EQ
	SVBUV31	−1
	SVBUV11	−1
	SVBUVA4	−1
	SVBUV15	−1
	SVBUV16	−1
	SVBUV26	−1
	SVBUBAS	1
ROW	**VBALGA1**	**TYPE EQ**
	SVBUBAS	−0.036
ROW	**VBALNVB**	**TYPE EQ**
	SVBUBAS	−0.049
ROW	**VBALVRS**	**TYPE EQ**
	SVBUBAS	0.951

ring to Table 13-5 showing the FCCU submodel, row CCAPFCU (FCCU capacity), the columns named FC1 and FC2 are the two operation modes of the FCCU. The sum of the column activity multiplied by the respective row coefficient of these two vectors must be equal to or less than the RHS of the CCAPFCU row.

The right-hand side of this equation is provided by the CAP Table, which provides the maximum and minimum FCCU capacity available for processing the various feeds to the unit. Table 13-9 shows the format of Table "CAP." The first column is process unit name codes. Columns

Table 13-9
CAP Table

TABLE CAPS	TEXT	MAX	MIN
CAT1	CRUDE UNIT NO 1	18.468	0.000
CAT2	CRUDE UNIT NO 2	18.351	0.000
CAT3	CRUDE/VAC UNIT NO 3	58.238	0.000
CAT4	CRUDE/VAC UNIT NO 4	87.811	0.000
CAT5	CRUDE UNIT NO 5	42.215	0.000
CVD6	6 VDU	41.909	0.000
CVBU	VISBREAKER	7.534	0.000
CACR	ASPHALT CONVERTER	4.191	0.000
CHDU	1 HDU	17.701	0.000
CIXM	2 HDU – MED SEVERITY	44.708	0.000
CPTF	PLAT FEED CAPACITY	10.822	0.000
CFCU	FCCU	38.556	0.000
CH2P	H2 PRODUCTION UNIT	7.827	0.000
CSFR	SULF. RECOVERY UNIT	0.149	0.000

"MAX" and "MIN" are maximum and minimum values of capacity of the units. These become the lower and upper bound of RHS of the matrix (Table 13-10) generated by capacity rows.

Product Blending

These equations ensure that the quantity of streams produced by process units are equal to the quantity of blend stock available for blending plus any loss and that going to refinery fuel. Also, the quantity of blend stock used in each product is made equal to the quantity of finished product. By convention, material or utility consumed by a process is shown as positive and material produced by a process is shown as negative. For example, if a unit consumes 100 units of a crude, that is entered in the matrix as +100, whereas if this crude unit produces 20 units of naphtha, 25 units of kerosene, 30 units of diesel, and 25 units of topped crude, all these outputs are entered in the matrix with a negative sign.

Product blending data are entered in the model as a table blend mix or as the properties of blend streams.

The table BLNMIX is, in fact, a blending map. The rows are the stream names and the column headings are the names of various product grades. An

Table 13-10
Matrix for Table 13-9

ROW	**CCAPAT1**	**TYPE LE**	**RHS**	**18.468**
	SLC1A1N	1		
	SLC1A1K	1		
	SLC1B1N	1		
	SLC1B1K	1		
ROW	**CCAPAT2**	**TYPE LE**	**RHS**	**18.351**
	SLC2A2N	1		
	SLC2A2K	1		
	SLC2B2N	1		
	SLC2B2K	1		
ROW	**CCAPAT3**	**TYPE LE**	**RHS**	**58.2382**
	SLC3A3N	1		
	SLC3A3K	1		
	SLC3B3N	1		
	SLC3B3K	1		
ROW	**CCAPAT4**	**TYPE LE**	**RHS**	**87.8108**
	SLCA4N	1		
	SLCA4K	1		
	SLCB4N	1		
	SLCB4K	1		
ROW	**CCAPAT5**	**TYPE LE**	**RHS**	**42.2148**
	SLCA5N	1		
	SLCA5K	1		
	SLCB5N	1		
	SLCB5K	1		
ROW	**CCAPVD6**	**TYPE LE**	**RHS**	**41.9089**
	SVD6RA1	1		
	SVD6RB1	1		
	SVD6RA2	1		
	SVD6RB2	1		
	SVD6RA5	1		
	SVD6RB5	1		
	SVD6VA3	1		
	SVD6VA4	1		
	SVD6W11	1		
	SVD6W21	1		
	SVD6W31	1		
	SVD6ARD	1		
	SVD6BS1	1		
	SVD6BS3	1		
ROW	**CCAPVBU**	**TYPE LE**	**RHS**	**7.543**
	SVBUBAS	1		

Table 13-10
Continued

ROW	**CCAPACR**	**TYPE LE**	**RHS**	**4.191**
	SACRV15	1		
	SACRV16	1		
	SACRV26	1		
ROW	**CCAPHDU**	**TYPE LE**	**RHS**	**17.006**
	SHDUMH1	1		
	SHDUMH2	1		
ROW	**CCAPIXM**	**TYPE LE**	**RHS**	**44.705**
	SIXMIX1	1		
	SIXMIX2	1		
ROW	**CCAPPLT**	**TYPE LE**	**RHS**	**10.8217**
	SREFNM0	1		
	SREFMM0	1		
	SREFNM3	1.059		
	SREFMM3	1.059		
	SREFNM5	1.161		
	SREFMM5	1.161		
	SREFNM6	1.161		
	SREFMM6	1.161		
	SREFNM7	1.240		
	SREFMM7	1.240		
ROW	**CCAPFCU**	**TYPE LE**	**RHS**	**38.5562**
	SFCUFC1	1		
	SFCUFC2	1		
	SFCUFC3	1		
ROW	**CCAPH2P**	**TYPE LE**	**RHS**	**7.8268**
	SH2PGAS	1		
ROW	**CCAPSFR**	**TYPE LE**	**RHS**	**0.1492**
	SSFRH2S	0.39714		

entry of 1 on the intersection of a row and column indicates that the stream indicated by row name is allowed to be blended in the grade indicated by the column name. The lack of an entry at this row/column intersection implies that this blend stock is not allowed to be blended in that grade.

Table BLNMIX, shows the process streams allowed into fuel oil grade I-961. The row names are blend stocks stream code. For example, an entry of 1 under column 961, in row UKE, shows that blend stock UKE is allowed in I-961 blending. Table 13-12 shows the matrix generated by

Table 13-11
BLNMIX TABLE

TABLE BLNMIX	TEXT	961
UKE	UNTREATED KEROSENE	1
TLD	SR.DIESEL POOL	1
WGO	DIESEL CUTTERS POOL	1
LVO	LT.VAC. GASOIL POOL	1
HVO	HVY.VAC. GASOIL POOL	
TRS	RESID POOL	1
AS1	1VDU ASPHALT	1
AS5	5VDU ASPHALT	1
HCN	HVY CAT NAPHTHAS	1
MCD	MEDIUM CAT NAPHTHA D/D	
TFC	HVY/LT CYCLE POOL	1

Table 13-12
Matrix for Table 13-11

ROW	EVBL961	TYPE EQ
	BVBL961	1
	BUKE961	−1
	BTLD961	−1
	BWGO961	−1
	BLVO961	−1
	BTRS961	−1
	BAS1961	−1
	BHCN961	−1
	BTFC961	−1

this table. The matrix shows that, in row EVBL961, the following columns have an entry:

BVBL961	Volume of fuel oil grade 961 blended.
BUKE961	Volume of untreated kerosene blended in grade 961.
BLTD961	Volume of light diesel blended in grade 961.
BWGO961	Volume of wet gas oil blended in grade 961.
BLVO961	Volume of LVGO blended in grade 961.
BTRS961	Volume of vacuum resid blended in grade 961.
BAS1961	Volume of asphalt blended in grade 961
BHCN961	Volume of heavy catalytic naphtha from FCCU blended in fuel oil.
BTFC961	Volume of cutter stocks blended in fuel oil 961.

The coefficient in the column BVBL961 is +1, while the coefficients of all other columns are −1, showing that the volume of the blended 961 is equal to the sum of the volumes of the individual blend stocks, as EVBL961 is an equality row.

The properties of blend streams coming from the crude and vacuum distillation units are entered in an ASSAYS table. The properties of process unit streams are entered in a number of tables called *blend properties*, or BLNPROP.

If a stream is formed by pooling a number of streams, its properties are unknown. However, a first guess of its properties is required to start the optimization process. Such guessed data are also provided in a separate table.

PROPERTY PROPAGATION TO OTHER TABLES

Many LP packages have a system of property propagation from property tables to process submodels and other tables. Properties of straight-run streams are entered in either the ASSAYS or BLNPROP tables. These property data, if needed in any other table, need not be reentered, placeholders (999 or some other symbol) are put in, which are replaced with data from the relevant data table.

Let us see how place holders are resolved. Suppose the ASSAYS table contains an entry for quality QQQ in stream SSS. This can be used to resolve placeholders in column SSS with row name xQQQabc in any submodel table. The row and column must be suitably named for placeholder resolution to occur.

EXAMPLE 13-1

In an ASSAY table, the data SG of kerosene stream from A1N mode of CDU1 is entered in row ISPGKN1 as follows:

A1N	
ISPGKN1	0.7879

These data are retrieved in a submodel table by the following entry:

	KN1
ESPGFDP	−999

Here, placeholder −999 is replaced with 0.7879, the specific gravity of KN1 as entered in the ASSAYS table. By convention, a negative sign is used for streams entering a recursed pool and their properties and a positive sign for the pool produced. Placeholder resolution can occur in user-defined E-, L-, or G-type rows as well as in recursion rows. Placeholders for recursed properties are replaced by the latest value of the recursed property of a stream every time the matrix is updated.

BLENDING SPECIFICATIONS

These equations ensure that each optimal product blend meets the specifications set for it. For example, Table 13-13 shows a part of a product specification table, listing properties of a fuel oil grade 961. Table 13-14 shows the matrix generated from this table. Row XVB1961 has as columns the blend components of fuel grade 961. The column coefficients are the VBI of individual blend components (see Table 13-15).

SPECIALIZED RESTRICTIONS

Several equations may be included in the matrix to reflect special situations in a refinery. For example, these may be the ratio of crude processed, restricting a processing unit to a particular mode of operation, or the ratio of the two products to be produced.

STREAM POOLING (RECURSION PROCESS)

Another class of refinery operation is important and must be modeled. This is the pooling of streams. A number of streams may be pooled to

Table 13-13
Specifications for Fuel Oil Grade 961

TABLE BLNSPEC	TEXT	961
XSUL	MAX. SUL	3.4
XVBI	MAX. VISCOSITY INDEX	461
N144	MIN. F.I. AT 144F	0.0001
XCON	MAX. CON. CARBON	14.3
XKLP	MAX. KERO INCLUSION	5

Table 13-14
Matrix for Table 13-13

ROW	XSUL961	TYPE GE
	(MAX SULFUR IN 961 FUEL OIL)	
	BWBL961	3.4000
	BUKE961	0.1078
	BTLD961	1.0182
	BWGO961	0.4842
	BLVO961	1.5451
	BTRS961	3.9652
	BAS1961	5.0974
	BAS5961	5.0829
	BHCN961	0.0137
	BTFC961	0.2619
ROW	XVBI961	TYPE GE
	(MAX VISCOSITY INDEX OF 961 F.O.)	
	BVBL961	4.6100
	BUKE961	−281.5921
	BTLD961	−37.7430
	BWGO961	−158.8042
	BLVO961	64.9132
	BTRS961	631.2328
	BAS1961	827.0000
	BAS5961	827.0000
	BHCN961	−365.0000
	BTFC961	108.4017
ROW	XCON961	TYPE GE
	(MAX CON CARBON OF 961 F.O.)	
	BWBL961	1.4300
	BUKE961	0.0000
	BTLD961	0.0000
	BWGO961	−0.0082
	BLVO961	−0.0174
	BTRS961	−15.2657
	BAS1961	−27.8173
	BAS5961	−27.8173
	BHCN961	0.0000
	BTFC961	−0.2573
ROW	N144961	TYPE LE
	(MIN FLASH INDEX AT 144 DEG F FOR 961 F.O.)	
	BKN4961	−0.1544
	BUKE961	1.6442
	BTLD961	−0.4645
	BWGO961	−3.3418

Table 13-14
Continued

	BLVO961	−0.6529
	BTRS961	−0.1598
	BAS1961	−0.0019
	BAS5961	−0.0019
	BHCN961	1.1520
	BTFC961	−0.3623
		−0.7700
ROW	**XKLP961**	**TYPE GE**
	(MAX KERO INCLUSION IN 961 F.O.)	
	BVBL961	0.0500
	BUKE961	−1.0000
	BHCN961	−1.0000

Table 13-15
Fuel Oil Blend Components

TABLE	BLNPROP	SG	SUL	VBI	CON	144
UKE	KEROSENE POOL	0.7891	0.1366	−281.592	0.000	−164.424
TLD	DIESEL POOL	0.8465	1.2022	−37.743	0.000	46.451
WGO	WET GAS OIL	0.8165	0.6018	−158.804	0.010	−334.182
LVO	LVGO	0.8683	1.7674	64.913	0.020	65.289
HVO	HVGO	0.9388	2.9855	396.298	0.000	
TRS	RESID POOL	0.9940	3.9986	631.233	15.358	15.918
AS1	ASPHALT 1 VDU	1.0350	4.9250	827.000	26.900	0.193
AS5	ASPHALT 5 VDU	1.0350	4.9110	827.000	26.900	0.193
HCN	HVY CAT NAPHTHA	0.8070	0.0170	−365.000	0.000	−115.200
TFC	FCC CUTTER POOL	0.9221	0.3256	108.402	0.279	36.232

form a single stream, which may become feed to a process unit or used for blending one or more products.[3]

The user must supply data on the properties of the pooled stream for optimizer to reach a solution. The user, however, has a problem, because the composition of the pooled stream is not known until after the optimum solution is reached. An iterative approach (recursion) is employed. The user provides a first guess on the properties of pooled stream. The optimizer then solves the model with the estimated data in it. After solving the model, an external program recalculates the physical properties of the pooled stream, which was earlier guessed. The revised physical

property data are inserted in the model, and the model is run again. The cycle continues until the delta between the input and output properties of the pooled stream are within specified tolerance limits. Recursion is, therefore, a process of solving a model, examining the optimal solution, using an external program, calculating the physical property data, updating the model using the calculated data, and solving the model again. This process is repeated until the changes in the calculated data are within the specified tolerance.

The structure for pooling a number of LVGO streams into a LVGO pool is shown in Table 13-16. The first recursion row is RBALLVO, which pools all streams into a LVO pool. By convention, the streams entering the pool have a negative sign and the stream produced by pooling has a positive sign. The name of this row is always RBALxxx, where xxx is the pool tag.

The next few rows have names starting with Rxxx; for example, RSPGxxx (specific gravity of the recursed stream), RSULxxx (the sulfur of the recursed stream), and RVBIxxx (the viscosity index of the recursed stream).

The properties (specific gravity, sulfur, VBI index, etc.) of the individual streams are known and provided in the model. The properties of the pooled stream are not known; however, a guess is provided in a separate PGUESS table, whose format follows:

	SG	SUL	VBI
LVO	0.8728	1.855	70.38

These PGUESS entries replace the placeholders (999) under column LVO and the first cycle of solving the matrix begins.

Suppose, after the model is solved based on these properties of the LVO pool, the activities of the vectors in the pooling model are as follows:

W11 = 0
W21 = 0
W31 = 0
L15 = 0
L25 = 0
L26 = 6.1380
MI4 = 10.9150

Table 13-16
LVGO Pooling

TABLE SLVO	LVGO POOLING TEXT	W11	W21	W31	L15	L25	L26	MI4	LVO
VBALW11	1 VDU WGO (FUEL MODE)	1							
VBALW21	1 VDU WGO (ASP MODE)		1						
VBALW31	1 VDU WGO (BSGO MODE)			1					
VBALL15	5VDU LVGO Normal				1				
VBALL25	5 VDU LVGO Asphalt					1			
VBALL26	6VDU LVGO (65 MBPD)						1		
VBALMI4	4A M/I DSL							1	
VBALLVO	LVGO POOL	−1	−1	−1	−1	−1	−1	−1	
	LVGO POOL								
RBALLVO		−1	−1	−1	−1	−1	−1	−1	1
RSPGLVO		−0.8672	−0.8498	0.8672	−0.8739	−0.8742	−0.8726	−0.8692	999
RSULLVO		−1.4977	−1.4676	1.4977	−1.5092	−1.5097	−1.5070	−1.5011	999
RVBILVO		−54.1025	29.0000	−54.1025	−82.4343	−63.0000	−78.1173	−54.3376	999

NOTE: THE COLUMN COEFFICIENTS IN RSULLVO ROW ARE THE PRODUCT OF THE SPECIFIC GRAVITY AND SULFUR (SG* SULFUR) OF THE BLEND COMPONENTS.

Substituting these values in row RSPGLVO gives

$$\text{S. gravity of LVO pool} = \frac{6.1380^*(0.8726) + 10.9150^*(0.8692)}{(6.1380 + 10.9150)}$$

$$= 0.8704$$

$$\text{Sulfur of LVGO pool} = \frac{6.1380^*1.5070 + 10.9150^*(1.5011)}{(6.1380 + 10.9150)}$$

$$= 1.5032$$

$$\text{Similarly VBI} = 62.896$$

The model is next run with these pool properties and the pool properties recalculated, based on new activities of the vectors. If the recalculated pool properties are unchanged or within the tolerance limits of the earlier values, the recursion process is stopped and the solution is said to have converged.

DISTRIBUTIVE RECURSION

In the simple recursion process, the difference between the user's guess and the optimum solved value is calculated in an external program, updated, and resolved. The distributive recursion model moves the error calculation procedure from outside the linear program to the LP matrix itself. With this arrangement, the optimum that is reached has the physical property data for all recursed streams exactly matching the composition of the pool used to create those properties.

After the current matrix is solved, using the initial physical property estimates or guesses, the new values are computed and inserted into the matrix for another LP solution. The major difference between distributive recursion and normal recursion is the handling of the difference between the actual solution and the guess. This difference is referred to as an *error.* When a user guesses at the recursed property of a pooled stream, the "error" is created because the user guess is always incorrect. The distributive part of the distributive recursion is that this error is distributed to where the quality is being used.

A pooled stream can go to a number of product grades or become feed to a process unit. The error vector is distributed wherever the pooled stream property is used. In other words, it can be said that the pooled stream properties are represented by two vectors, one is the initial guess of the property and the second is the error or correction vector that seeks

to bring the property in line with property computed from the composition of the blend.

EXAMPLE 13-2

Consider three catalytic reformate streams, R90, R95, and R98, from a catalytic reformer. The reformate streams are pooled into a single stream, SPL. The pooled reformate stream is used for blending three gasoline grades: I-390, I-395, and I-397 (see Figure 13-1).

All three gasoline grades are blended from following blend components:

BUT	Butane
LSR	Light Naphtha
LCN	Light Cat Naphtha
SPL	Pooled reformate stream

And, the pooled reformate stream is

ROW	R90	R95	R98	SPL	RRONSPL
*					
VBALR90	1				
VBALR95		1			
VBALR98			1		
*					
VBALSPL	−1	−1	−1		
RBALSPL	−1	−1	−1	1	
RRONSPL	90	95	98	94	1

To start with, the recurse process, an initial guess, is made as to the RON of the pooled stream. Let us assume it is 94 as shown in the matrix, in row RRONSPL.

Now, an error vector, RRONSPL, is introduced in the matrix to absorb any error made in estimating the RON of the pooled octane (i.e., 94).

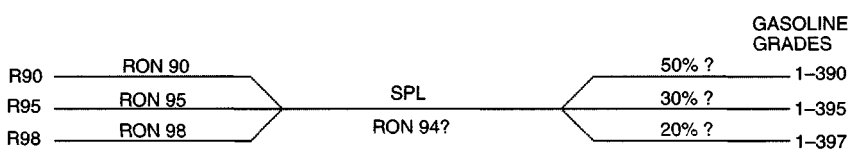

Figure 13-1. Principle of distributive recursion.

Let us assume that activity of columns R90, R95, and R98 are 5, 3, and 2, respectively, and calculate the activity of the error vector RRONSPL.

ROW	R90	R95	R98	SPL	RRONSPL
COL ACTIVITY	5	3	2		
RBALSPL	−1	−1	−1	1	
RRONSPL	−90	−95	−98	94	1
ROW ACTIVITY	−450	−285	−196	940	−9

By column arithmetic, the activity of the error vector RRONSPL is computed at −9.

This error is distributed in all the grades where the pooled stream is used. To start the distributive recursion process, a guess is made as to the distribution of the error in the three gasoline grades as follows:

I-390	50%
I-395	30%
I-397	20%

We consider the blending and octane balance of all these three gasoline grades. For I-390,

	BVBL390	BBUT390	BLSR390	BLCN390	BSPL390
EVBL390	1	−1	−1	−1	−1

VBL390 is volume blend balance. E shows that it is an equality row. For the RON of I-390,

	BVBL390	BBUT390	BLSR390	BLCN390	BSPL390	RRONSPL
NRON390	90.2	−93	−61.8	−91.2	94	(= −9*0.50)

Since I-390 gasoline happens to be one of the gasoline grades where pooled reformate stream SPL is being disposed of, we use the guessed value of its octane number for computing the RON of the I-390 blend. Also, a part of the error vector, RRONSPL (50%), is also included in this row. This error vector is designed to correct the error in guessing the octane number of SPL, and this aids in faster convergence of the solution to the optimum solution.

The matrix structures for I-395 and I-397 blending are similar, except for proportion of error vector included, which would be equal to the assumed distribution of the error vector.

For the RON of I-395,

	BVBL395	BBUT395	BLSR395	BLCN395	BSPL395	RRONSPL
NRON395	90.2	−93	−61.8	−91.2	94	= (− 9)*0.30

For the RON of I-397,

	BVBL397	BBUT397	BLSR397	BLCN397	BSPL397	RRONSPL
NRON397	90.2	−93	−61.8	−91.2	94	= (− 9)*0.20

The matrix representation of the pooled stream, if it becomes feed to another process unit, is discussed under delta-based modeling.

OBJECTIVE FUNCTION

The matrix picture discussed so far is not a complete LP model. To drive the optimization process, an extra row, called the *objective function*, is needed. The optimization process either minimizes or maximizes this function, depending on whether the function represents the refinery operating cost or profit. All cost vectors must have an entry in the objective function row. The coefficients in this row are the costs per unit of product produced. The cost of crude and other feeds, such as natural gas, or utilities are entered with a negative coefficient if the objective function represent overall profit to the refinery.

The unit cost data, which become the coefficients of the variables in the objective function row, are retrieved from following data tables of the model: the BUY table for crude and other feedstock prices, the SELL table for all product prices, or the UTILBUY (utility buy) table for all utilities, such as refinery fuel, electricity, cooling water, catalyst, and chemical costs.

OPTIMIZATION STEP

The simplex procedure for optimizing a set of linear equations was originally introduced in 1946 by Danzig[4] of the Rand Corporation (USA).

It did not become really popular until it was computerized in 1950s. In essence, the procedure is to first find a solution, any solution, that satisfies all the simultaneous equations. This may be as simple as assigning 0 to all unknowns, although this, by no means, assures a valid solution. Usually, it is best to start with a previous solution to a similar problem. Then, the activity of each matrix column or unknown is examined and the one is selected that yields the largest profit or, if no revenues are shown, the minimum cost per unit of use. Next, each row is examined to determine which equation restricts the use of this activity to the smallest value before the other activities in the equation are forced to go negative.

For example, in the equation $x + 2y = 10$, x restricts the activity of y to equal to or less than 5. Otherwise, x has to take a negative value to satisfy the equation.

The activity for the most restrictive equation is solved ($y = 5 - x/2$), and the solution substituted in all other equations containing this activity, including the objective function. In fact, this activity has been made part of the solution, or in the LP jargon, "brought into the basis." The matrix element or coefficients are modified, and the procedure is repeated. When selection of any activity reduces profit or increases cost, as indicated by negative coefficient in the modified objective function, the procedure is concluded.

SOLUTION CONVERGENCE

Remember that properties of all pooled streams are assumed or guessed to start the optimization process. After the first cycle, the optimum activities of all columns or variables are known, the properties of all blend streams are again computed, and the matrix reoptimized. This process is repeated until there are no changes in the input and output properties of the streams. The optimized matrix is next sent to the "report writer," which prints a report in a preset format using matrix data (see Figure 13-2).

INTERPRETING THE SOLUTION

The output from the optimizer is in the form of an unscrambled report, which is a listing by rows, then by columns, in the same order as the input

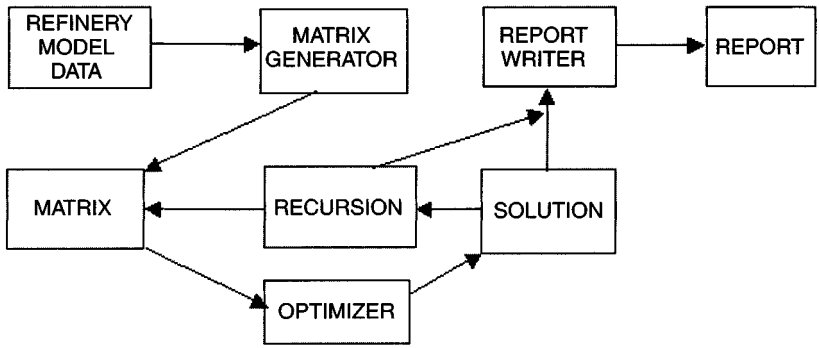

Figure 13-2. Refinery LP system.

in the original matrix. The basis is the collection of activities in the matrix, which are a part of the solution.

ROWS

The output for each row gives the row number and the row name supplied by the user.

The abbreviations used in the rows are:

BS. "Basis" is an indication that, in the final solution, this row is not limited to its upper or lower limit.

LL. "Lower limit" is an indication that the final solution is limited at a lower level by a lower bound established by this row.

UL. "Upper limit" is an indication that the final solution is limited at an upper level by an upper bound established by this row.

EQ. "Equal" means that the final solution is restrained to a fixed right-hand side. Such rows show upper and lower bounds that are identical.

Activity

This is the value of the left-hand side of the equation that the row represents, with values of the equation unknown supplied from final solution. For the objective function, activity is the maximum profit or minimum cost.

Slack Activity

This is the difference between upper (or lower) limits and the row activity just described. Values for slack activities are given only for rows that have "BS" status.

Lower Limit

This is the value given to the right-hand side in the original matrix that left-hand side must be greater than or equal to.

Upper Limit

This is the value given to the right-hand side in the original matrix that left-hand side must be less than or equal to.

Dual Activity

This is also known as *marginal cost*, *shadow price*, or *pi value*.

The pi value represents the rate of change of the objective function as the right-hand side of the row is increased. It is the total change in maximum profit or total minimum cost due to relaxing the upper or lower limit of a row by one unit. For example, if it is a crude availability row, it reflects the value of an incremental barrel of crude refined. It has nonzero value only if the limit is constraining. Runs that have the status of BS in the optimum solution show no dual activity. Marginal values are valid only over a small range around the optimal solution and should be used with caution.

COLUMNS

The output for each column gives

Column number. The column number starting from wherever the row numbers leave off.

Column name. Column names may be supplied by user or generated by matrix generation programs from input data tables.

Activity. This is the value of a variable in the final solution. Some variables in the basis may have no activity at all. Variables with zero activity may be in the basis because a certain number of variables (equal to the number of rows) are required to be in the basis by the nature of LP method.

Input costs. These are the coefficients found in the objective function row selected for optimization, and they allow total cost and revenues to be accounted for in the objective function row.

Lower limit. This is the lower limit (called *lower bound*) that value of an activity can have.

Upper limit. This is the upper limit (called *upper bound*) that value of an activity can have.

Reduced cost. This is also known as the *D-J* or *Delta-J* value. It is a reduction in net profit or increase in minimum cost, if an activity, not in the basis, is brought into the basis and given the value of 1. This is cost of using an activity that is not part of optimal solution. All reduced costs are zero or positive in an optimal solution.

REPORT WRITER PROGRAMS

The Report Writer Program retrieves data from the unscrambled solution and rearranges them in a more user-friendly format using a special programming language. A typical refinery LP solution may present the matrix solution report in the following format:

1. A summary report, showing an overall refinery material balance, starting with crude and other inputs; the overall production slate; and the refinery profit from operation or an economic summary.
2. The material balance of all refinery units.
3. A product blending report, showing the blend composition and properties of all blended product grades.

DELTA-BASED MODELING

Delta-based modeling (DBM) is a technique used to predict the yields and properties of the process units, when the yields and

the properties are a function of the feed quality. In many situations, the feed is a pool of streams whose composition must be determined by the optimization process itself. Delta-based modeling is especially useful in these situations when combined with distributive recursion techniques. To implement the DBM, a feed quality parameter is defined that can be measured easily and related to the yield of the unit. For example, the yield of reformate is known to be a function of naphthene plus aromatic (N + A) content of the feed. Suppose the base case reformer yields are defined for a feed with an (N + A) content of 30.

In most LP applications, the feed is a pool of streams whose composition must be determined by the optimization process. Therefore, the properties of the feed to the unit are unknown. After pooling various cat reformer feed streams, if the model computes the reformer feed at (N + A) content at 45, a yield correction vector corrects the base yields to correspond to a feed with an (N + A) content of 45.

EXAMPLE 13-3

Consider the following cat cracker model (FCCU Table):

	SFCUBAS	SFCUSUF	SFCUCFP
VBALC4U	−0.0322	0.0047	
VBALPOR	−0.0381	0.0016	
VBALLCN	−0.2990	0.0307	
VBALMCN	−0.0670	−0.0067	
VBALHCN	−0.1150	−0.0037	
VBALLCO	−0.2600	0.0192	
VBALHDO	−0.2170	−0.0359	
ECHGFCU	1		−1
ESUFFCU	0.284	1.0000	2.986

The activity of the column SFCUCFP is made equal to the total feed to the cat cracker unit by user-defined equality row ECGHFCU. The cat cracker feed pool is formed in another submodel and has only one disposition, which is the cat cracker unit. Thus, if the activity of the column SFCUCFP is 10 mbpcd, the activity of the column SFCUBAS is also made equal to 10 mbpcd, as they are driven by the row ECHGFCU as follows:

	SFCUBAS	SFCUCFP	
ECHGFCU	1	−1	= 0

The base yield of the FCCU is determined for a sulfur content of 0.284% by weight. The FCCU feed pool is formed by combining a number of streams. Suppose, by the recursion process, the sulfur content of FCCU feed pool (SFCUCFP) is found to be 2.986%. In this case, the base case yields (FCUBAS) are corrected by a correction vector SFCUSUF. The numbers in this column are adjusted to the yield of FCCU for a 1% change in the sulfur content of the feed.

The activity of this column is determined by the row EFCUSUF, as follows:

	SFCUBAS	SFCUSUF	SFCUCFP	
ACTIVITY	10	X	10	
ECHGFCU	1		−1	=0
EFCUSUF	0.284	1.0	−2.986	

By matrix arithmetic,

$$\text{EFCUSUF} = 10^{*}0.284 + \text{X} + (-10^{*}2.986) = 0$$

$$\text{X} = 27.02$$

Let us next see how it effects the yields from unit SFU.

	SFCBAS	SFCSUF	TOTAL, mb	YIELD %
ACTIVITY	10	27.02		
VBALC4U	10*(−0.0322) = −0.322	27.02*0.0047 = 0.1260	−0.1960	−0.0196
VBALPOR	10*(−0.0381) = −0.0381	27.02*0.0016 = 0.0440	−0.3370	−0.0337
VBALLCN	10*(−0.2990) = −2.990	27.02*0.0307 = 0.83	−2.1600	−0.2160
VBALMCN	10*(−0.0670) = −0.6700	27.02*(−0.0067) = −0.1800	−0.8500	−0.0850
VBALHCN	10*(−0.1150) = −1.1500	27.02*(−0.0037) = −0.1000	−1.2500	−0.1250
VBALLCO	10*(−0.2600) = −2.6000	27.02*(0.0192) = 0.5200	−2.0800	−0.2080
VBALHDO	10*(−0.2170) = −2.170	27.02*(−0.0459) = −1.2402	−3.4102	−0.3410

The base yields were derived for a sulfur content of 0.284 wt %. As the sulfur content of the FCCU feed was 2.986 wt %, the shift vector SFCUSUF adjusted the base yield to that for a sulfur content of 2.986%.

Therefore, we see that the yields of all products from the FCCU are changed as follows:

FEED	0.284% SULFUR	2.986% SULFUR
C4U	−0.0322	−0.0196
POR	−0.0381	−0.0337
LCN	−0.2990	−0.2160
MCN	−0.0670	−0.0850
HCN	−0.1150	−0.1250
LCO	−0.2600	0.2080
HDO	−0.2170	−0.3410

It is possible for the model to have more than one shift vector. This model structure is similar to the one discussed earlier, as shown in the following example.

EXAMPLE 13-4

Let us consider a delta-based model segment with two shift vectors instead of one:

	SFCUBAS	SFCUSUF	SFCURES	SFCUCFP
VBALC4U	−0.0300	0.0047	−0.0012	
VBALPOR	−0.0370	0.0016	−0.0011	
VBALLCN	−0.3290	0.0307	−0.0114	
VBALMCN	−0.0980	−0.0067	−0.0020	
VBALHCN	−0.0790	−0.0037	0.0000	
VBALLCO	−0.2380	0.0192	−0.0007	
VBALHCO	−0.0800	0.0000	0.0186	
VBALHDO	−0.1300	−0.0359	−0.0020	
ECHGFCU	1			−1
ESUFFCU	0.284	1.0000		2.986
ERESFCU	0		1.0	5.000

Here, the first shift vector corrects the base case yields for different sulfur content of the feed, while the second shift vector modifies the base case yields for inclusion of 5.0% residuum in the FCCU feed. It is implied

that the feed pool resid content is computed elsewhere by recursion of pooling streams and treats resid content as a property of the feed.

DATA FOR DELTA-BASED MODELS

The data used in the delta-based model must be developed by the user to reflect the parameters and their effect on the process yield. In the context of the refinery streams, only streams properties that are generally measured can be considered. For example, the cat reformer yield can be related to the naphthene plus aromatic content of the feed if the refinery has reference data to correlate the cat reformer yield at a given severity vs. (naphthene plus aromatic) content of the feed.

Care has to be taken that the shift vectors do not extrapolate data beyond the range for which these were developed. Also, as the shift vectors can take either positive or negative values, these should be declared free in the BOUND section of the LP model.

ATMOSPHERIC CRUDE DISTILLATION AND VDU MODELING

LP models for crude and vacuum distillation can be constructed exactly in the same manner as done in the case of other process unit submodels. Most refineries may have more than one crude unit and may be processing more than one crude oil. Also, a crude unit may operate in more than one mode; for example, one crude unit mode may maximize the production of naphtha and another may maximize the production of kerosene. It is, therefore, convenient to enter all data on yields and properties of the various cuts from different crude oils and crude and vacuum distillation units present and their utility consumption, pooling of various cuts from crude and vacuum units in three tables for easy updating.

ASSAYS TABLE

Data in the form of yields of various crude on different CDUs and properties of cut produced is entered in a single table, the Assay Table (Table 13-17). This table does not directly generate any matrix but only