CHEMICAL PLANT LAYOUT

A Thesis presented to the

University of Sheffield

for the degree of

Doctor of Philosophy

by

Martin John Bush B.Sc.Tech.

Department of Chemical Engineering

University of Sheffield

February, 1972.

TEXT CUT OFF IN THE ORIGINAL

SUMMARY

The rules and considerations governing site and plant layout are presented. The philosophy of plant layout is discussed and the factors involved in the layout of the site modules are listed. The space requirements of each module are assessed and tabulated.

The considerations involved in the plant layout process are presented for the most common types of plant items. The clearances associated with each type are tabulated. Some arithmetic and algebraic methods of plant layout are assessed and an approximate method of determing the cost of stacking plant items is presented.

A simple technique for the simulation of plant layout, based on the string diagram analogue, is presented and a Toluene Hydrocracker plant is laid out by this method.

The application of computers in this field is discussed and a number of small available programs are assessed. An interactive computer graphics program that will effect and simultaneously cost a layout has been written and is presented. The program is described in detail and the Toluene Hydrocracker plant is laid out using this technique.

PUBLICATIONS

Some aspect of this thesis have been reported in the literature. British Chemical Engineering has published a paper [35,36] based on the initial stages of this work. The Institution of Chemical Engineers, Engineering Practice Committee Working Party on Plant Layout has also incorporated parts of this thesis in their forthcoming publication. A short film has also been made of the graphics program in the process of laying out a plant on the computer screen.

ACKNOWLEDGEMENTS

I must acknowledge the help of my tutor and mentor Dr. G.L. Wells without whose encouragement and unceasing vigour this work would not have been completed. Also my thanks go to Imperial Chemical Industries for allowing me to use their computer facilities at Central Management Services, Wilmslow. In particular I must thank Mr. G. Knights and Mr. R. Kissach for their help with programming difficulties.

CONTENTS

CHAPTER 1

INTRODUCTION

The layout of chemical process units is commonly accepted as the one activity in the design process where one moves from mathematical analysis into the ill-defined regions of subjective experience. In any attempt to produce a computer based system for the overall design of chemical plant - from process design to pipework detailing - chemical plant layout is the area where the tools of the mathematician are least effectual. It has been said[7] that chemical plant layout is more art than science. This study is an attempt to analyse the state of the art.

The work may be divided into three parts. A survey of existing layout philosophy, techniques and data; the simulation of plant layout and the application of the digital computer to this field.

Objective

The object of chemical plant layout is the minimisation of an objective function, namely net annual cash outflow. This cash flow is constituted by a number of components which may be isolated as follows:-

1. Capital cost of erected plant units.
2. Capital cost of erected pipework and fittings.
3. Operating costs of the plant.

The optimisation of process pipework is the main way in which analysis of the chemical plant layout leads to a cost saving. It is therefore important that the greatest possible consideration be given to pipework routing during the layout. Pipework routing is an integral part of chemical plant item layout and cannot be either ignored completely or treated separately. To do so is to attempt a cost reduction while ignoring the regime where this may be accomplished.

The attainment of the above objective is constrained by certain practical considerations which must be taken into account at the layout stage. These are discussed briefly below.

Safety Requirements

The safety of operating personnel in plants handling toxic or flammable substances is of the greatest importance.

Certain types of plant items which it is thought constitute a hazard may be isolated. Units such as storage tanks or direct fired units such as furnaces or vaporisers are commonly positioned in this way. This can result in increased piping length and therefore cost, and also in increased plot area. The degree of isolation and the associated increase in the margin of safety must therefore be weighed against these cost increases.

Operation

Although chemical plants are becoming increasingly automated, some degree of manual operation and supervision is required. An important part of an operators duties is to walk round the plant, visually and aurally monitoring the equipment. Plant items such as reactors with catalytic packing require an adjacent clear area for the storage of catalyst drums and for access by cranage or other equipment necessary to load and remove the packing from the tower. Batch units such as some centrifuges, filters, dryers etc., may need quite large free areas surrounding them to facilitate the easy removal of internal parts during operation. Valves and instrumentation associated with a particular plant item may occupy a significant amount of space. It is of the utmost importance that the plant items be located a sufficient distance apart because inadequate access to valves or instruments can make plant operation difficult, inefficient and unsafe.

Maintenance

Maintenance requirements for chemical plant units will sometimes dictate that quite extensive free areas are available. Nearly all chemical process units must be overhauled at least once a year and many items require maintenance work in service. This may involve removing internal parts or even dismantling the equipment entirely. For example, the tube bundle of a heat exchanger has to be removed for cleaning or for the replacement of a damaged tube or baffle. This means that there must be a clear space at the channel end of the exchanger at least as long as the length of the tube bundle. If any other equipment intrudes into this space then the cost of maintaining the exchanger either by moving the offending equipment or by lifting the exchanger out completely, increases significantly. Similarly for fixed tube heat exchangers, sheet space must be left for replacing a tube. Space must also be left around equipment for the application and maintenance of paintwork and lagging. Also access space for cranage may be necessary for the maintenance servicing of some units.

Erection

Not all the plant items that constitute the chemical plant can be delivered to the site at the time when the engineer would wish. Some complex plant items may have very long delivery times. In the mean time the field fabrication and erection of other plant items is continuing. This means that a long delivery item such as a column or boiler may have to be erected when all ancillary and surrounding equipment is in place. A heavy plant unit requires large lifting equipment for its erection and sufficient space must be available for this purpose. Failure to consider the space requirements of these items can lead to long delays or greater expense at the erection stage of the plant. It should be noted, however, that many Construction Departments pride themselves on overcoming any erection difficulty arising from the selected layout. The best plan is perhaps to ignore access for erection initially but to let the Construction Department view the alyout before finalising the plot plan.

Expansion

There should be adequate space for the future expansion of both equipment and pipeways. It has been suggested that the pipetrack should provide about 30%

of its space for future lines[1]. If the plant is a new process, space left for last minute modifications should be considered. However decisions on space for expansion should be based on specific requirements.

CHAPTER 2

LAYOUT CONSIDERATIONS

2.1 Site Layout

A complex chemical plant must be divided into a number of sections or modules to limit the amount of capital at risk. The suggested figure for each section is that fixed capital plus inventory should be in the region of £1 million to £2 million. Site layout is concerned with the location of these sections or modules within the site limits. Each area should be classified according to B.S. C.P. 1003, Part 1. The sections fall into two categories. On-plot modules are the sections formed by divisions inherent in the process, e.g. - compression, crystallisation, fractionation, purification, pyrolysis, reaction, recovery, refrigeration etc. Off-plot modules are comprised of areas not directly involved in the chemical process but still vital to the effective functioning of the complex. Typical sections would comprise amenity buildings, analysis byildings, boilerhouses, control rooms, cooling towers, effluent treatment and disposal areas, feedstock and product storage areas, firestations, flarestacks and chimneys, laboratories, loading and off-loading areas, oil-water seperators, offices, power stations, railways, roads and car parks, substations and switchrooms, water treatment plant, workshops and stores.

The technique of site layout is basically very simple. It can be broadly states as follows:-

1. Position the modules on the site in process flow sequence, with due regard for the considerations listed in this section and observing the spacing requirements given in section 3.1.

2. Arrange manufacturing, service and storage modules on either side of a pipebridge in one or more of the following shapes:

3. Calculate the average diameter of the pipelines running between sections and ignore all lines below this figure. The average diameter may be found by multiplying the line size by the footage, summing for all the pipelines and dividing by the total footage.

4. Examine quatitatively the juxtaposition of various modules for minimum piping costs. Do not neglect service pipelines.

5. Examine the sections to see whether different sub-divisions of the process might not be better. For example, two modules might both contain a refrigeration unit and it mught be advantageous to group these units in a separate section.

The factors affecting the layout of the modules comprising the plant complex are discussed below.

Amenity Buildings

1. Canteens should be located in a safe area preferably with attractive surroundings and within a short distance of the main concentration of labour.

2. Amenities huts specifically designed for controlling smoking may be provided adjacent to process units. Such buildings should be located so as to avoid infringement of any hazardous area classification.

Boiler House

1. Locate in a position where it will not be affected by a major fire and keep steam mains as short as possible.

2. Consider the effect of any troublesome or prevailing wind.

Control Room

1. Locate the control room as close and as central to the plants controlled as possible.

2. Flares and most hazardous areas should be visible from the control room.

3. There should be direct access to the most vulnerable controlled areas in the plant for manual interruption when required.

4. There should be a safe escape route from the control room in case of a hazard.

5. The control room should be directly accessible from the switchroom.

Cooling Towers

1. Locate cooling towers such that mist is carried away from the plant by the prevailing wind.

2. Cooling towers should be sited so that water drift will not pass across roads, railways or public amenities.

3. They should be orientated cross-wise to the prevailing winds to minimise recycling of air from the discharge of one tower to the suction of an adjacent tower.

4. The arrangement of multiple natural-draft cooling towers should take account of resonant frequencies generated by through-wind velocities.

5. Examine the possibility of entrainment of corrosive vapour from adjacent plants, and due consideration should be given to the position of boilers and other chimneys and flarestacks.

6. Consider the merits of centralising or decentralising cooling water facilities in the light of any possible site expansion.

Effluent Treatment and Disposal

1. The location of effluent treatment facilities and drainage systems should take advantage of the terrain.

2. The condition of effluent must meet statutory requirements. These may be administered by the receiving authority, the local authority or Alkali Inspectorate.

3. Clearly the contamination of any local water or agricultural land by overflow or leakage of fluids from the plant must be avoided.

4. Gaseous effluent should be burnt or discharged at a height such that in no circumstances can offensive fumes become a public nuisance.

5. Consider the effects of any troublesome or prevailing winds on gaseous effluent discharge.

Feedstock and Product Storage Areas

1. Feedstock and product storage areas should be located as close as possible to the means of transportation.

2. Tanks are always sited in groups.

3. Tank farms should not be sited on more than two sides of the process area.

4. Access must be available on all four sides of each tank bund area, and all roads should be linked in such a way that access in either direction is possible where any road is blocked by fire.

5. Tanks should be arranged to keep the hot lines, and those carrying flammable or corrosive fluids, as short as possible.

6. The spacing and bunding of tanks should be in accordance with Part 2 or Part 3 of the Institute of Petroleum Electrical Code.

7. Main storage tanks containing highly flammable liquids and other hazardous contents should be stored in locations designated for that purpose, away from tank farms containing less flammable material.

8. Any limitation imposed by local management, local conditions or local bye-laws should be taken into account.

9. When exothermic chemical reaction is possible between stored liquids, tanks should be segregated and not bunded together.

10. Where polymensation is probable between stored liquids, the tanks should be segregated, and consideration should be given to increasing the spacing as much as possible.

11. Tanks should be grouped and bunded so that the contents of the tanks in one bund will require one type of fire-fighting equipment only. This applies particularly when both water miscible and water immisable liquids are stored in the same installation.

12. For tanks grouped together consideration should be given to a common walkway with not less than two means of escape, depending on the number of tanks served.

13. Space should be allowed for foam or drenching systems and for the withdrawal of heating coils.

14. Loading gantries should be as far away from the process plants as process conditions will allow.

15. It is not advisable to use dykes around vessels in which LPG and similar materials are stored.

16 The process units should be to one side of a rectangular tank farm, and the shipping facilities on the other side in order to minimise the length of piping runs between process units and tankage and between shipping facilities and tankage.

17. Intermmediate tank storage should be located close to the process units, blending, shipping and crude tanks progressively further out.

18. Buffer space should be provided between the tank farm and the process units to accommodate miscellaneous small tankage and other minor facilities which later may become necessary.

<u>Flarestacks and Chimneys</u>

1. Locate in a position where the concentration of hazardous materials at working levels is within the permitted range.

2. Locate in a position where radiation from the flare does not affect plant or human life.

3. Flare stacks should be visible from the control room.

4. Consider the decentralisation of the flare[10] system; each major plant having its own flare.

<u>Loading and Offloacing Areas</u>

1. Locate loading and offloading facilities at terminal points of the plant so that regular traffic through the plant is minimised.

2. Road tanker loading and unloading bays may have to be classified as danger areas. Such bays should be at the periphery of the offsites area.

3. It is desirable that loading bays have one way traffic.

<u>Oil-Water Separators</u>

1. Oil-water separators are hazards and should be treated accordingly.

<u>Offices</u>

1. Plant Offices should, if possible, be grouped in safe areas adjacent to large process plants or groups of process plants.

2. Administration buildings should be located on the public or safe side of the security point and close to the main entrance if possible.

3. The main office building should always be near the main entrance.

Power Station

1. Locate in a position where it will not be affected by a major fire and keep steam mains as short as possible.

Pumping Station

1. Consider the decentralisation of tank farm pumping equipment. This may be preferable to a central pumphouse in order to minimise damage in the event of fire.

Railways

1. Railways within works and rail links with the national network should be laid out in consultation with the local railway authorities to meet the works requirements for raw materials reception and product despatch.

2. Consideration should be given to any limitations imposed by danger area requirements.

3. Examine any special requirements for rail tanker loading and unloading, with adequate radii at curves for the largest tanker or railcar.

4. Care should be taken to avoid boxing in the plant by branch lines.

Roads and Carparks

1. Roadways should be laid out only for access to the plant and should be arranged so that road

vehicles do not pass near process areas and do not violate danger area classifications.

2. Dual through roads should allow the passage of two vehicles. Corner radii should allow the passage of the largest vehicles, and any special loads associated with construction, without maneouvring.

3. Pedestrian pathways adjacent to roads should be allowed for in areas of high personnel concentration and traffic movement,

4. Access roads in plants should be wide enough to permit easy maneouvring of vehicles and mobile cranes. Care should be taken to ensure adequate turning space for tankers and vehicles at loading and offloading bays.

5. Car parks for personnel and visitors and their access roads should be in a safe area and outside security points.

6. Dead-end roads should be avoided and convenient access from any point of the plant should be possible in at least two different directions.

Substations and Switchrooms

1. Locate substations and switch rooms as near as possible to the centre of gravity of the electrical load.

2. The switchroom must be directly accessible from the control room.

3. Consideration should be given to alternative cable routes, away from hazardous areas, and dupli-

cate routes for major cable runs.

4. Substations, transformers and switchhouses (except those forming an integral part of a plant) should be located so that they will not be affected by any plant fire. They should be sited in areas which will permit industrial switchgear to be used.

Water Treatment Plant

1. Boiler feed water treatment plant should be located as near to the boiler house as possible.

2. Boilers and water-treating facilities should be located in a position where a major plant fire would not be likely to cause interruption of the steam supply.

Workshops and Stores

1. Locate in a safe area and preferably within easy reach of the process units.

2. Direct access should be provided for traffic which should not pass through process areas.

3. Warehouses, salvage yard and workshops should be close together. These should also be adjacent to a railway siding.

In addition, the following factors must be considered:

1. Works must meet all statutory and Local Authorities requirements and also bear in mind any impending legislation with regard to noise levels, atmospheric pollution, odour, effluents etc.

2. Examine whether hazardous areas can be grouped together and segregated. They should be located in such a way that gaseous or vaporous hazard is carried away from the plant, surrounding plant or habitations by the prevailing wind.

3. Consider the following environmental factors:

(i) Topography - streams, swamps, ponds, trees.

(ii) Existing roads, railways, structures, equipment pipelines etc.

(iii) Adjoining property.

(iv) Noise, odour or light considerations.

(v) Utility or service entry points.

(vi) Geological or geographical considerations.

4. Special construction requirements and/or delivery dates may dictate the location of heavy tall or large equipment. These factors must be considered at the site layout stage.

5. Plant and offsite areas handling highly flammable or hazardous materials, should be separated from administration areas by a fence, admission to the plant and offsite areas being through gateways constituting security points.

6. The works should be provided with a security boundary fence and all entrances should be provided with a gatehouse.

7. When an extension is being built, this building site should have a separate entrance and gatehouse and should have its own boundary fence. The main works boundary fence may serve as part of

this boundary, but any gateway between the main works and the building sire must have its security control point.

8. Areas classified as hazardous should not overlap the plot limits nor extend over railways where open firebox engines are likely to be employed.

2.2 Site Layout Spacing Requirements

1. Establish spacing between modules. Typical spacing requirements are given in Table 1.

2. Establish dimensions of roads, drains, railways, pipebridges, buried pipes and cables. Include space for electrical or instrument cables instrument lines, ducts etc. Allow an adequate margin depending on the state of the design.

3. Ensure there is adequate access for

(i) Firefighting

(ii) Maintenance trolleys, cranes etc.

(iii) Construction cranes, derricks etc.

(iv) Lifting, trailing and moving of units during the construction and erection stages. Mobile cranes are normally used for equipment weighing up to 80 tons, guyed derricks are to be used for equipment weighing between 80 and 100 tons.

4. Allow adequate headroom beneath pipebridges etc., for tankers, cranes, trolleys, or fire engines.

5. Consider the possibility of future expansion.

6. Consider allowing space for last-minute modifications especially with a new process.

PLOT LAYOUT - RECOMMENDED SPACING [1,2,3]

TABLE 1

No.	Item		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	Main Sections of Plant		50'																						
2	Tanker and Drum Filling Areas																								
3	Pipe Bridges		(2)(3)																						
4	Oil Water Separator		25'																						
5	Flare Stack		4'		(2)(4)	(4)																			
6	Plant Road		25'		(2)	25'	(2)(4)																		
7	Works Fence		50'			50'		(2)																	
8	Fired Heaters				(7)																				
9	Occupied Plant Buildings				(2)																				
10	Buildings cont[n] Elect. Equipment		(5)		(8)	(5)																			
	PRESSURE STORAGE																								
11	Hydrocarbons	Ethylene	200'	50'		25'	200'		200'				(9)												
12		C_3's	150'				150'		150'																
13		C_4's	100'				100'		100'																
14	Non-hydrocarbons insoluble in water	Methyl Chloride	75'				75'		75'																
15		Vinyl Chloride																							
16		Methyl Vinyl Ether																							
17		Ethyl Chloride					50'																		
18	Non - HC's soluble in water	Methylamines																							
	LOW PRESSURE REFRIG. STORAGE																								
19		Ethylene	300'	50'			300'	50'	300'				(10)								(11)				
20		C_3's	150'				150'		150'																
21		C_4's	100'				100'		100'																
22	Non -HC's soluble in water	Ethylene Oxide					50'		50'																
23	Flammable liquid Storage Tanks (14)		50'	(6)	(2)	25'	(4)	50'				(5)	(12)								(13)				(1)

Numbers in brackets refer to the notes overleaf.

Notes on Table 1

1. The layout of liquid storage areas will be governed by the Model Code of Principles of Construction and Licencing conditions - Petroleum (consolidated) Act 1928.

2. No specific recommendation but consider the hazards involved.

3. Equipment can normally be under pipebridges but avoid large inventories of flammable material and critical items of equipment such as pumps above auto-ignition temperature, atmospheric boiling p-int or well above flash point.

4. Distance usually determined by acceptable radiation level from flare - usually considerably more than 50ft.

5. Depends on whether buildings contain ignition sources and whether they are pressurised. Consider explosion or deflagration hazards.

6. A tanker and drum filling area may be taken to be part of an adjacent storage area where appropriate. Refer to the MOdel Code of Principles of Construction and Licencing conditions - Petroleum (consolidated) Act 1928.

7. Allow 50ft. for a main and 25ft. for a minor pipebridge.

8. Normally 10ft. minimum.

9. Spacing of one quarter of the sum of the diameters of adjacent tanks, between the shells.

10. Spacing of 50ft. from the bund wall of the low pressure tank, but not less than 100 ft., from the low pressure tank shell.

11. Spacing of half the sum of the diameter of adjacent tanks, between the shells.

12. Spacing of 50 ft., from the bund wall of the flammable liquid tank.

13. Spacing of not less than 100 ft., between low pressure refrigerated L.F.G. and flammable liquid tank shells, but L.F.G. and flammable liquids must be in separate bunds.

14. Flammable liquids are those with flashpoints up to 65.5°C, (150°F).

2.3 Plant Layout

Once the project is approved the site and plant layout will be proceeding simultaneously with other actions concerned with the construction of the plant. For example, process unit and pipework preliminary design and detailing, together with civil and structural engineering design will have already begun although the data available is, at this early stage, limited. These considerations dictate that the production of the unit plot plan must be effected as quickly as is practicable. If a company has a number of different plants that it is considering building, it will need to have an idea of the capital cost of each of them. The cost of a plant will depend on its layout and therefore, this must be partially effected for each plant. In this case the technique for layout must not only be speedy but cheap as well. No company wants to spend money laying out plants it will not build. These factors indicate that the layout technique must be as simple but as effective as possible.

Certain data is required if the layout is to be accomplished effectively. These can be noted as follows:-

1. A process flow sheet showing every item of major equipment and its size.

2. The weight of every item of major equipment.

3. The materials of construction of equipment and piping.

4. Process equipment elevation requirements.

5. The nature of flowing fluids e.g. vapour - liquid, slurry, paste etc.

6. Vendors drawings for mechanical equipments showing

   (i) critical dimensions

   (ii) space required to be left clear for maintenance.

   (iii) locations of connections.

7. Atmospheric conditions with regard to extremes of weather which may make it desirable to provide special shelter or protection for equipment or operators.

8. Preferred operating and maintenance practices as they affect:

(i) Shelter that operators are accustomed to.

(ii) The need for permanent or temporary shelter when maintaining mechanical equipment such as pumps, centrifuges, compressors, dryers etc.

There is no single technique leading to the best arrangement in any problem. Several stages may be required with different techniques applied to each.

The development process is cyclic, usually repeated at successive levels of detail from the first broad assessment of a project to the final development in detail prior to erection. According to the problem, layout may be considered at several levels from the selection of a site and arrangement of the plants on it, through block layout relating plant sections to one another, to the detailed planning of working areas and the equipment in them.

To this approach may be related the three basic principles of layout planning:-

1. Plan the layout as a whole then consider the details. Sub-optimisation should be avoided.

2. Plan the ideal layout and then give consideration to the practical difficulties. The ideal layout is easier to produce and may be rapidly costed. The cost penalties of deviating from the ideal layout due to practical considerations may then be assessed.

3. Plan more than one layout. No one layout may be considered the "best" for the given process

conditions.

Planning more than one layout encourages discussion and dialogue, leading to greater confidence in the layout finally selected.

The layout of plant units will vary according to the types of plant. There are three types of plant layout.[25]

(i) Layout by fixed position - all operations performed with material in one fixed location.

(ii) Layout by process function - similar operations located together.

(iii) Layout by process flow - position in sequence of operations and material movement.

In the chemical industry the layout engineer is usually concerned with layout by process flow.

The initial development of the layout is carried out using flow sheets, supplemented where necessary by process outline, multiple activity charts etc. For many plants the initial development is possible using a layout analogue such as block models, cutouts etc. For comple situations those have physical limitations and should be used either following or in conjunction with other techniques which attempt to produce an early assessment of the problem. Such an approach attempts to establish the objectives to record the limitations and restrictions, to examine their validity, and to check their consistency. When all this has been done, the next step is to apply them to the problem, to determine what arrangements or groups of arrangements are ruled out and which remain as possibilities; and of these to select for further development any that might compare favourably with the others.

Techniques of correlation and compatability can be used formally in eliminating what is impossible or inadmissable and in listing, for evaluation arrangements compatible with the objectives and constraints as stated in cases where it is not practicable to do this mentally. The list of possible arrangements obtained in this way will usually need further reduction before it becomes practicable to begin detailed evaluation. The selection might be made on qualitative criteria or by determining the controlling costs and proceeding to minimise them. Controlling costs are those that vary most significantly from one feasible layout to another.

To determine controlling costs the linkages between all the items are recorded and a cost per distance run obtained for each.

For example, between

Buildings - pipes; material transfers; sizes of surrounds for access, aesthetics, extension or development.

Plant items - pipes; conveyors; man movements.

Components - connections; access for fitting, maintenance.

Materials - supply and delivery points; quantities and rates of movement; stores.

It is then possible to determine which of such costs are of major importance in comparing different arrangements. Some special techniques for complex problems include:

1. The Correlation Chart

The correlation chart is a diagrammatic method of determining the effect of constraints, and recording the arrangement that they allow. In some cases objectives or preferences can also be introduced as constraints so narrowing the field still further.

A grid is drawn with the rows representing possible positions - such as floor in a building, or numbered positions in an area of one plant item, and the columns representing possible positions of another. Constraints are recorded and labelled, say x,y,z ... if any combination of positions of the two items is disallowed by X then that letter is written into the corresponding square of the grid. Vacant squares thus show permitted combinations.

The sets of lines of the grid can be produced and crossed by rows or columns representing other items, and prohinitations or preferences again applied. Then the only feasible permutations are those that can be traced through the rectangular network using only vacant squares. The advantages of this method are its visual presentation and the ease with which it can be learnt and used; the disadvantages are the amount of drawing-up required, and the difficulty in correlating items that are not contigous on the chart.

2. Algebraic Method

In this method the possible positions for different items are represented by algebraic symbols: thus A_1

means item A in position 1. Ordinary rules of algebra can then be used with the special meanings

addition represents alternatives

multiplication represents co-existence

thus A_1 $(B_2 + B_3)$ means item A in position 1 and item B in either position 2 or position 3.

If a permutation is inadmissable it is put equal to zero - i.e. it is deleted. It is not necessary to multiply out brackets when applying constraints, and it may be possible to impose more than one of these at the same time. On multiplying out at any stage, the terms represent all the permutations still allowable at that stage.

This method is not as simple as the correlation chart but is more powerful in dealing with multiply permutations and easier to read, though not to visualise.

As an example of this method consider the following process, the outline process chart for which is shown in Fig. 1. The problem is one of vertical layout and to arrange the vessels for these process stages on the five floors (Ground, 1st, 2nd, 3rd, 4th) of an existing building.

The constraints can be listed as follows:

1. Reaction vessel and Pressure Filter discharge from below and should not be put on the ground floor. Thus R_G and F_G are eliminated.
2. The Pressure Filter is not to go on the 4th floor as there would not be enough room to withdraw the spindle for maintenance. Thus F_4 is eliminated.

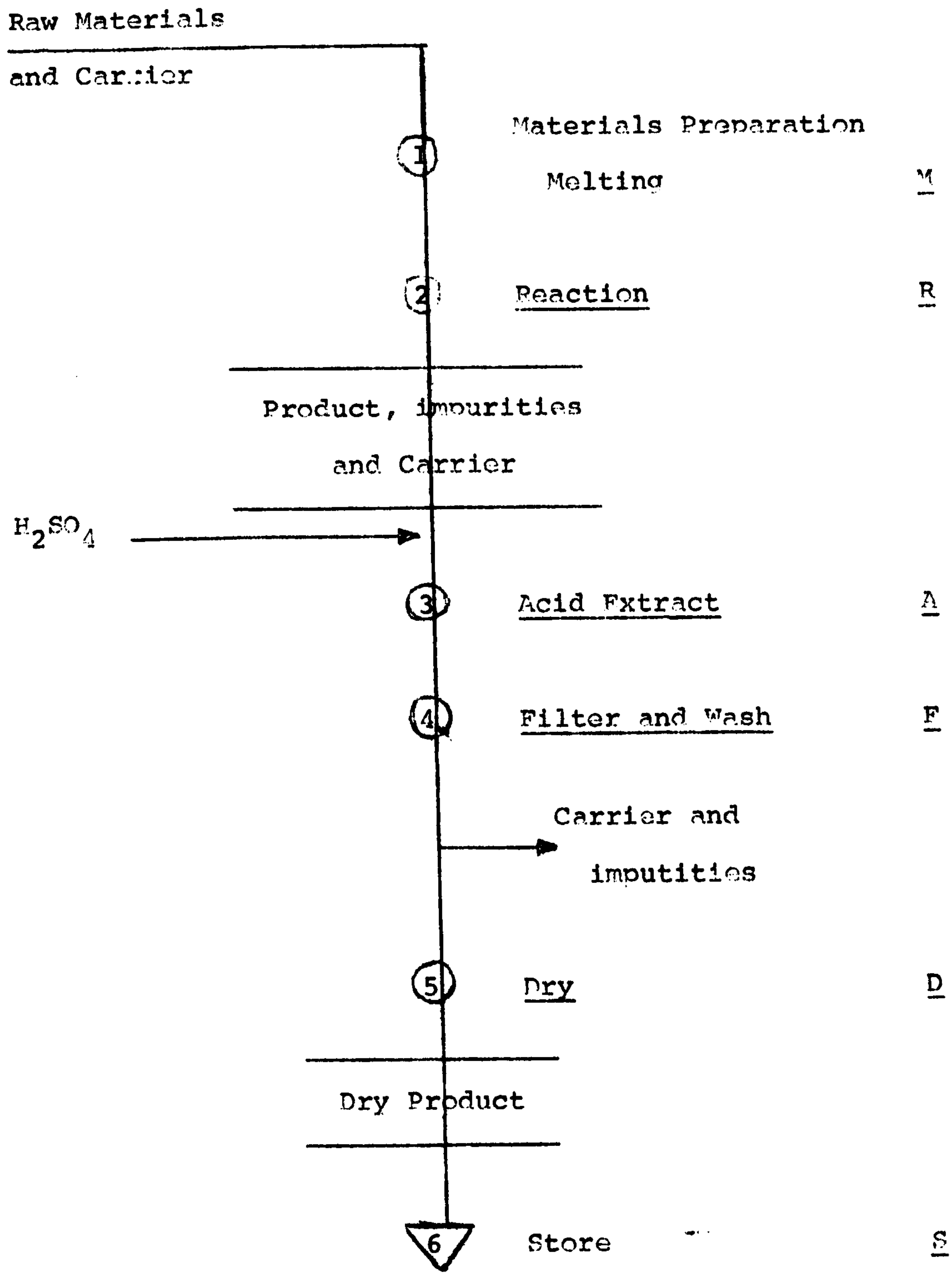


Fig. 1

Process Chart

3. The storage hopper (S) must be above a weigh hopper which is situated on the 1st floor. Thus S_G and S_1 are eliminated.

So all the feasible layouts are represented by:-

$(M_{G,1,2,3,4})\ (R_{1,2,3,4})\ (A_{G,1,2,3,4})\ (F_{1,2,3,})\ (D_{G,1,2,3,4})$

$(S_{2,3,4})$

4. It is desired to leave the Reaction and Acid Treatment vessels in their present position on the 2nd and 1st floor respectively. Thus R_2 and A_2 are to be chosen if possibbe

5. Considerations of heat transfer and gravity flow suggest that Melting should be adjacent to Reaction and above it. This means that M_3 or M_4 should be chosen; if these M_3 is nearer to R_2 and so is to be preferred.

6. For ease of handling the cake from the Pressure Filter it should be dropped directly into the Drier. This implies $F_1 D_G + F_2(D_G + D_1) + F_3(D_G + D_1 + D_2)$

7. Subject to all other conditions the cost of lifting materials is to be minimised. Lifting may be necessary for moving slurry from A to F, and/or moving powder from D to S. Thus S_3 and S_4 are ruled out as not minimal. Similarly it rules out F_2D_G, F_3D_G, F_3D_1; all of which impose more lifting than the other three relations between F and D.

The items to be evaluated in terms of the costs of moving slurry and powder are then reduced to:-

$M_3R_2A_1F_1D_GS_2$: slurry moved horizontally on 1
Powder to be raised from G to 2

$M_3R_2A_1F_3D_2S_2$: slurry to be raised from 1 to 3; powder moved horizontally on 2.

$M_3R_2A_1F_2D_1S_2$: slurry and powder each raised one floor from 1 to 2

Should it transpire that there is congestion on any of the floors then additional lifting costs may be accepted. Terms like M_4 or S_3 could be brought back into consideration.

3. Travel Charts

These were originally used for siting machines in jobbing shops, and the name related to the record of the amount of travel of different jobs between the machines.

A square grid is drawn and labelled across and down with the names of each item: every square is the meeting point of a row and a column, each representing an item and in it is inserted the total cost per unit distance of all linkages between those two items. In the case of neighbouring items this cost will lie in a square next to the main diagonal: for items that are not neighbours the cost of linking all items is found by multiplying the cost in each square by the distance of that square from the main diagonal and summing. Visual inspection of the columns then shows what interchanges (permutations of the items) would reduce this total cost by bringing certain costs nearer the main diagonal. The process is repeated until an optimum arrangement has been reached.

The method can be modified to compensate in part for different plant sizes. As a method of approximating, certain simplifications can be made: for example it can be used with only controlling costs; again, groups of items that are to be sited near to one another can be treated as a single item with 'group' linkages to other items.

The main disadvantage of the method is that it only leads to an optimum linear arrangement, which is seldom wanted: on the other hand it measures the relative importance of having different pairs of items close to one another, which is a useful first step to the two or three dimensional problems.

2.4 Models

These can be made from blocks of wood, cardboard or polystyrene to a scale of about 1:50, positioned above, but not attached to a base board faced with scale paper. A proprietory kit may be used. Standardised structural components are also available. It is claimed that such kits drastically reduce drawing time and the need for cut-outs.

These low cost models are used chiefly to develop plot and floor plans, and elevations, but should not be used for piping layout. They show only the major item of equipment and major pipe racks in very simple form and in their correct relationship to each other. The more expensive large-bore piping may be shown. Buildings, control rooms, switch rooms and roads should also be indicated.

This type of model may be used for proving the proposed plot plan and should ensure that all constraints on the layout has been satisfied. It also accomplishes the basic planning for the location of equipment to attain economical piperuns. When final agreement on the layout has been reached then the piping design model can start.

The piping model is prepared to give an accurate detailed layout of process piping, utilities, and control facilities. The scale is generally not less than ¼" = 1 foot, and the model should at least show all piping of 2" bore and above. According to policy all pipework including instrument pipe runs may be included. The cost of a design model for chemical and allied type of plants are generally in the range of ¼% to 1% of the total installed plant cost being modelled, according to the size and cost of the plant. An average figure is ahour ½%. The total design costs for the average chemical plant is in the range of 8% to 10% - or greater - of the total installed plant cost.

The main uses are as follows:-

(i) Having established the location of the main plant items, it will give optimum piping arrangements.

(ii) Saving costs in drawing time.

(iii) Eliminating site fouls.

(iv) Positioning instruments, cable racks, access platforms, lighting etc.

(v) For planning of construction.

(vi) For use on the construction site.

(vii) For the training of operating personnel.

(viii) Exhibition and advertising.

It serves as an immediate source of reference during discussions between drawing office, project engineers, construction engineers, safety officers and clients.

2.5 Plant Item Considerations

When positioning the plant units on a plot plan, certain considerations apply to each type of unit. These are listed below.

Centrifuge:

(i) If possible locate indoors in heated buildings.

(ii) Centrifuges should not be located close to vibratory equipment such as mills, crushers or reciprocating compressors.

(iii) Avoid locating under corrosive areas or piping so that spillages do not corrode safety interlocks, motors, control wiring etc.

Columns and Towers

(i) Decide whether towers shall be self-supporting or supported by structures.[29]

(ii) There must be sufficient room available for a tower to be laid down on the ground adjacent to its foundation.[29]

(iii) Columns are best located towards one end of the process area.

(iv) If frequent catalyst loading is envisaged it is advantageous to allocate an area for dumping spent catalyst, and for setting up temporary loading scaffolds or rigs.

(v) The erection procedure should be carefully considered so that special lifts or testing procedures do not hold up work on other equipment. Consider whether the tower needs to be elevated, i.e. how far above grade the base of the tower should be located.

Filters

(i) The liquid/solid separation equipment should be located at the final solids discharge point.

(ii) Obtain from the manufacturer, if possible, a typical general arrangement drawing and his recommendations for layout.

Fired Heaters

(i) Consider the relative positions of the services for the heater - such as sub-stations, fuel storage tanks, water-systems and pumphouses.

(ii) Consider the disposal of effluents and waste products.

Heat Exchangers

(i) Consider whether heat exchangers should be grouped. Most units in the same service are grouped automatically.[15]

(ii) Air-coded exchangers are usually located adjacent to the plant they serve and may be positioned on the pipebridge.

(iii) Air-cooled exchangers should not be located close to process units, instrumentation, or control rooms which could be adversely, affected by either vibration or noise.

(iv) Decide whether overhead condensers shall be located above columns or at grade.

(v) When locating air coolers above yard banks it is desirable to reserve overhead crane access to reach at least one side of each cooler.[4]

(vi) Consider whether the grouping of condensers between two towers will result in a shortening of cooling water lines.[12]

(vii) Consider whether a common steam line can be designed for grouped reboilers.[12]

(viii) Short reboiler and overhead lines to condensers are essential for both economy and reliable operation.

(ix) Product coolers or exchangers located between process equipment and the unit plot can be located at one end of the plot.

Mills and Crushers

(i) Locate at grade and away from equipment or instrumentation affected by vibration.

(ii) If the speed of the machine is in excess of 1000 r.p.m., the wind tunnel effect in the discharge chute must be considered and steps taken to avoid dust nuisance.

(iii) For multiple units, arrange them in line to simplify the feeding and delivery systems.

Pumps and Compressors

(i) Decide whether to group pumps and compressors in buildings or to arrange in process flow sequence.

(ii) Pumps should be located at grade as far as possible.

(iii) Suction lines should be kept short and uncomplicated. This will largely determine the best location for the pumphouse.

(iv) Consider the foundation block size and decide whether pipes and cables will be constructed in ground or overhead.

(v) Arrange pumps in line with drivers facing the access gangway. Double rows of pumps can be arranged with pumps back to back running up to a common pipebridge.

(vi) Pumps located below yard banks should have their drive ends lined up.[4]

(vii) Locate pumps below their point of suction and do not neglect net positive suction head requirements.

Such a list is of necessity less than exhaustive as it is not the purpose of this thesis to produce a handbook of layout practice, but rather to develop an improved technique.

CHAPTER 3

SIMULATION OF PLANT LAYOUT

3.1 Plant Layout Spacing Requirements

The first part of this study was concerned with the physical simulation fo the plant layout process. This is necessarily a physical modelling process because mathematical modelling is not possible to any large extent. The difficulty is in defining mathematically the considerations and constraints imposed on the layout. Also the large number of combinations and permutations of plant items and modules that occur for a complex plant, would make mathematical modelling prohibitively time consuming and expensive. The plant items collectively constituting a chemical plant may be simply represented by blocks of cardboard or wood fashioned approximately to scale. These blocks may be positioned on a board representing the site area and the various layout configurations may then be discussed and assessed. The simplest way of attempting to decrease plant capital cost is to position all plant items as close as possible to one another. The operating costs of such a plant would be very high as the maintenance and indeed the operation of the plant would be extremely difficult. This means that all plant items must be at a minimum distance from one another. Table 2 gives approximate clearances that should be left between chemical plant items. This table should serve only as a guide. In practice manufacturers drawings and accurate data relating to the individual items must be consulted.

PLANT LAYOUT - RECOMMENDED CLEARANCES

TABLE 2

EQUIPMENT		ELEVATION	OPERATION		MAINTENANCE/GENERAL ACCESS
			HORIZONTAL	VERTICAL	
CENTRIFUGES CLASSIFIERS			10'	6' + λ[1]	12' access corridor.
CRUSHERS MILLS			At least the width of the machine	6' + λ	12' access corridor.
CRYSTALLISERS EVAPORATORS			5'	6' + λ	
DRYERS			5' + λ	6' + λ	
COLUMNS Note 5	PLATE		5' Note 4		10' between adjacent columns.
	PACKED		5' Note 4		10' between adjacent columns. Note 2.
FIRED HEATERS			10'		See Table 1.
REACTORS STIRRED VESSELS			5'	10' + λ	Note 2.
HEAT EXCHANGERS		REBOILERS: Check column elevation	6' + at channel end 6' at shellcover end 4' at sides	4'	
TANKS					See Table 1.
PUMPS		N.P.S.H: Check back to point of suction	6' at motor end 4' at sides		3' between pumps and piping. 7' working aisles. 4' between pumps and walls. Note 3.
FILTERS		10' if solids dump	5' + λ		6' between adjacent filters.
COMPRESSORS			5' + λ	10' + λ	

Notes on Table 2

1. λ is the length of the longest internal part of the equipment that must be removed for maintenance or operation.

2. If frequent catalyst loading is envisaged it is advantageous to allocate an area for dumping spend catalyst.

3. In the open, pumps handling hot liquids (>60°C) should be at least 25 ft. from pumps handling volatile liquids (b.p.< 40°C). In pump rooms they should be separated by a tight wall.[1]

4. For extremely high columns, for which special foundation designs are necessary, the area required for the foundations may define the spacing.

5. Space adjacent vertical vessels at least 2½ standard platform widths between shells.[4]

3.2 Elevation of Plant Units

The decision whether or not to elevate or stack certain plant units will depend mainly on economic considerations. Sometimes it is less costly to stack units than to locate them at grade. The cost saving may be affected by a resulting shortening of the pipe-rack and the size reduction in plot area. In practice this last consideration is negligible. A method comparing the cost of elevated units with the cost of units at grade must therefore be devised.

The cost of a block of stacked units will consist of the sum of a number of cost terms each representing one of the following factors.

	Factor	Cost term - dollars
1.	Excavation	0.77 (W + 1)
2.	Backfill	0.7 (W + 1)
3.	Foundations and slab	5.6 (W + 1) 1.0 nWl
4.	Floor slabs	1.9 (n - 1)Wl
5.	Building Frame	1.35 (n + 1) Wl
6.	Stairways	52.6 nh
7.	Handrails	10.2 (W + 1)

Also included must be a cost term which accounts for the change in pipe length, and thus in cost, when pipework is routed to units which are elevated. It is assumed that the pipes are carried on a pipe-rack at a height equal to that of the second storey of the stacked units. This cost term is thus the sum of the costs for each unit and may be expressed by:-

$$\sum_{i=1}^{i=r} P_i h \ (n_i - 2)$$

r = number of units in the block

P_i = sum total per foot of the piping costs for the ith unit

n_i = storey on which the ith unit is located.

Thus the expression for the total erected cost for a block W by l and containing r units is

$$C_1 = 7.07(W+l) + 4.25nWl - 0.55Wl + 10.2n(W+l) + 52.6nh$$

$$+ \sum_{i=1}^{i=r} P_i h(n_i - 2) \qquad \ldots (1)$$

The arrangement of plant units is decided by using a number of cost relationships which may be derived from this equation for the cost of the erected block.

1. Vertical Increment

Given a block of (m - 1) storeys, what is the cost of one more floor?

from equation 1:- the cost of block of (m - 1) storeys is

$$C_{m-1} = 7.07(W+l) + 4.25(m-1)Wl - 0.55Wl + 10.2(m-1)(W+l)$$

$$+ 52.6(m-1)h + \sum_{i=1}^{i=r} P_i h(n_i - 2)$$

The cost of a block of m storeys with j units on the mth (added) floor is

$$C_m = 7.07(W+1) + 4.25mWl - 0.55Wl + 10.2m(W+1) + 56.2mh$$

$$\sum_{i=1}^{i=r} P_i h(n_i \sim 2) + \sum_{i=1}^{i=j} P_i h(m \sim 2)$$

therefore the cost of the mth storey $C_m - (C_m - 1)$ is

$$4.25Wl + 10.2(W+1) + 52.6h + \sum_{1}^{j} P_i h(m \sim 2) \quad \ldots (2)$$

or alternatively the cost of the first unit on the mth storey is

$$4.25Wl + 10.2(W+1) + 52.6h + Ph(m \sim 2) \quad \ldots (3)$$

2. <u>Horizontal Increment</u>

Given a block of m storeys where the mth storey contains fewer units than the others, what is the cost of adding one more unit onto the mth storey?

This follows directly from equation 2, the cost of the mth storey of width W is

$$4.25Wl + 10.2(W+1) + 52.6h + \sum_{1}^{j} P_i h(m \sim 2)$$

if another unit of width δ is added the cost is now

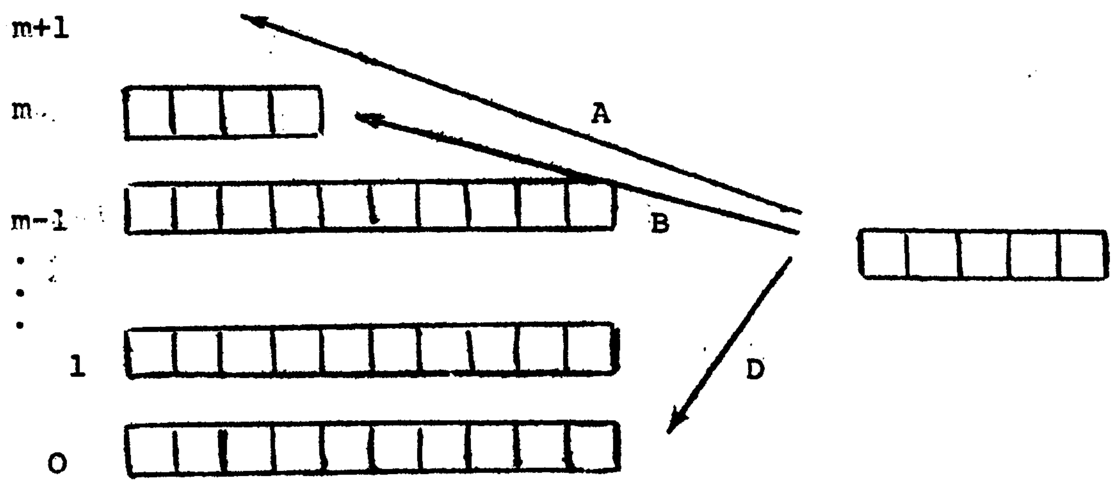
$$4.25(W+\delta)l + 10.2(W+\delta+1) + 52.6h + \sum_{1}^{j+1} P_i h(m \sim 2)$$

and the cost of the extra unit is

$$C_3 = 4.25\delta l + 10.2\delta + Ph(m \sim 2) \quad \ldots (4)$$

3. <u>Relative Positional Cost</u>

Given a block of m storeys with the mth storey incomplete how shall an extra k units be added?



shall they be added to the mth storey (2) or stacked above on the (m + 1)th storey, or at grade? Each position can be considered separately.

(A) the cost here is given by equation 2.

$$C_{(A)} = 4.25Wl + 10.2(W+1) + 52.6h + \sum_{1}^{K} P_i h\{(m-1)-2\}$$

( ) the cost here is given by equation 4.

$$C_{(B)} = 4.25Wl + 10.2W + \sum_{1}^{K} P_i h(m-2)$$

(D) the position here extends the piperack by W feet and the cost of the pipework routing must also be considered.

$$C_{(D)} = C_p W + 2h \sum_{1}^{K} P_i$$

C_p = cost of piperack per foot.

The important question here is the comparison of positions (A) and (B)

$$C_A - C_B = 10.21 + 52.6h + h \sum_{1}^{K} P_i |\{(m+1)-2\} - \{m-2\}|$$

for m > 1 this expression is always positive, i.e. $C_A > C_B$.

So it is always less costly to build on to an existing structure than above it.

However for $m = 1$

$$C_A - C_B = 10.21 + 52.6h - h \sum_{1}^{K} P_i$$

so if $\sum_{1}^{K} P_i > \frac{10.21}{h} + 52.6$

then it is less costly to start a new floor than to proceed with the first storey.

4. <u>Application of Cost Equations</u>

Consider a number of units of effective dimensions (i.e. taking into account access and maintenance).

$$W = 6.5 \text{ ft.}$$

$$l = 30 \text{ ft.}$$

it shall be assumed that the units are similar and that

$$h = 8 \text{ ft.}$$

$$P = 25 \text{ \$ / ft.}$$

$$C_P = 200 \text{ \$ / ft.}$$

Such data is specific to these examples only.

The first unit will obviously be located at grade. For the second unit there are two alternative positions

1 2

Case 1, from equation 1 as $n = 1$

$$C_1 = 7.07(W+1) + 4.25Wl - 0.55Wl + 10.2(W+1) + 52.6h + 2hP$$

$$= \$\ 2373$$

plus 6.5 ft. of piperack costing $\$\ 1300$

<u>cost = $\$\ 3673$</u>

Case 2, the cost here is $2\ (2hP + C_pW)$

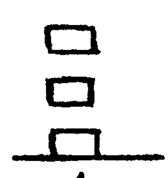
(see section 3)

$$C_2 = \$\ 3400$$

$$\therefore \quad C_1 > C_2$$

Thus both units are located at grade, as in position two. It is obvious that if a unit (or units) is placed at grade next to position 1 or 2 then the inequality $C_1 > C_2$ will still exist. Therefore for the third unit the positions available are

3.



4.

$$C_3 = 3(2hP + C_pW) = \$\ 5100$$

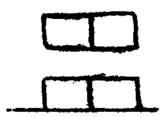
$$C_4 = 7.07(W+1) + 4.25nWl - 0.55Wl + 10.2n(W+1) + 52.6nh$$

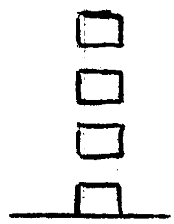
$$+ hP + 2hP = \$\ 4053 \qquad \text{where } n = 2$$

$$\text{plus } \$\ 1300 = \$\ 5353$$

$$C_4 > C_3$$

thus the three units are located at grade. As before the addition of a unit does not disturb the inequality, so for the fourth unit we must consider the positions


5.


6.


7.

$$C_5 = \$\ 6402$$

$$C_6 = \$\ 6959$$

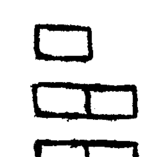
$$C_7 = 4(2hP + C_pW) = \$\ 6800$$

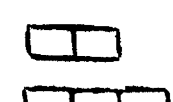
thus $C_6 > C_7 > C_5$

therefore the four units are stacked as position 5.

Now than an elevated structure has been indicated, the assessment becomes easier as decisions are based on incremental costs, units being added to the existing arrangement.

e.g. for the fifth unit


8.


9.

the increase in cost incurred by placing the unit as in position 8 is given by equation 2.

$$C_8 = 4.25Wl + 10.2(W+l) + 52.6h + Ph(n-2) \quad n = 2$$
$$= \$\, 1602 \quad w = 6.5$$

whereas the cost of placing it at grade is again $ 1700

$$C_9 > C_8$$

The process may now be continued, each time comparing the indremental cost of the added unit, horizontal or vertical to the unit grade cost which in this case is $ 1700. In this way any number of units may be positioned.

This calculation will vary for each case and in some cases maintenance must be considered. In these examples no major economic error would ensue by acopting any of the cases. As it is appreciated that such calculations are not readily practicable it might be stated as a rule of thumb that the layou engineer should stack if he feels that he will thereby achieve a layout better in terms of appearance and operability. However if maintenance problems are anticipated stacking should be avoided.

3.3 Unit Plot Plan Production

The general form of the layout must first be decided. Normally the most efficient in terms of maintenance, access and operability will be a layout where the plant units are positioned each side of a central piperack. The layout may be effected using wooden blocks and drawings, but experienced layout engineers are necessary for this method. For less experienced personnel a simple technique permits a similar layout to be achieved. Pipe runs are optimised within the layout by means of a string piping analogue; a technique originally used for minimising the total distance travelled by say a storekeeper. The string (or elastic) is used to represent pipe runs, the thickness of the string being proportional to the cost of the pipes per unit length. By reducing the overall length of string the pipeline capital costs are decreased. An alternative to thick strings is coloured strings, coded according to the unit costs of the pipelines. In this way the most expensive piperuns are immediately identifiable and consideration can be given to reducing their length.

Celluloid sheeting (paper or cardboard may also be used) is cut to scale, about 1:60, to represent the plot area required for the operation of each plant unit. This area must take into account the constraints noted earlier i.e. access for maintenance, safety etc. For example, if the unit is a heat exchanger with 16 ft. tubes, the clearance at the channel end of the exchanger, required for the removal of the tube bundle, would be 21 ft. In addition 2 ft. should be left in front of the shell cover and 3 ft. each side of the shell. Therefore the plot area required for an exchanger with a 31 inch diameter shell would be 45 ft. by 8 ft. 7 ins. A cutout should be produced for every unit that is to be located, and each unit should be numbered as for the Process Flow Diagram. The actual position and size of the unit should be marked on the cutout. The cutouts are laid on a board the surface of which should be crosshatched. The plot area is marked, to scale, together with the position of the major pipe runs and access roads if these are already fixed. When positioning plant units the maintenance areas are overlapped so that units may be compactly located without compromising on maintenance and access areas. The layout procedure can be summarised as follows:-

(i) Position the process units on the plot in process flow order, with due regard for the considerations listed in section 2.5 and observing the spacing requirements indicated in section 3.1.

(ii) Locate the process units at grade, unless process requirements dictate otherwise.

(iii) Arrange the units each side of a secondary piperack branching off the main piperack.

(iv) Where a structure is necessary because of elevation requirements, consider the location of other process units within it to reduce the plot area.

(v) If there is insufficient plot area available for grade location, some units will have to be located within a multi-storey structure.

(vi) Examine quantitatively the juxtaposition of various process units for minimum piping costs. Do not neglect service pipelines.

(vii) When positioning a major unit, locate all ancillary equipment at the same time. For example, when siting a tower the positioning of any associated condensers, reboilers, accumulators, pumps etc., must be dealt with simultaneously.

There are numerous factors to be taken into account at this stage of the layout. The more important considerations are listed below.

(i) Consider which units require shelter and whether a cost saving is realised by grouping them within the same building.

(ii) Examine the elevation requirements of all process units. In particular those due to n.p.s.h., gravity flow and process control requirements.

(iii) Decide whether gravity flow might be preferable to pumping.

(iv) Decide whether to provide a structure to support an elevated process unit, or whether to support it on columns or other tall equipment.[29]

(v) Assess alternative methods of materials transportation i.e. batch transport using drums or mechanical conveyors.

(vi) Consider whether the re-orientation of some units is advantageous i.e. horizontal drums or reboilers can be positioned vertically.[16]

(vii) Examine whether foundation costs and pipebridge costs can be reduced by stacking exchangers or drums.[16]

(viii) Thought should be given to the location of equipment requiring frequent attention by operating personnel and to the relative position of the control room to obtain the shortest and most direct routes for operators when on routine operation.[22]

(ix) Hot lines should be long enough for adequate flexibility and should have enough elbows to minimise or eliminate the need for expansion joints.[10]

(x) Operating hazards should be studied so that the safest arrangement of equipment can be devised.

3.4 Toluene Hydrocracker Layout

As an example of the procedure outlined above, the layout of a small toluene hydrocracker plant was attempted. The process flow diagram is shown overleaf.

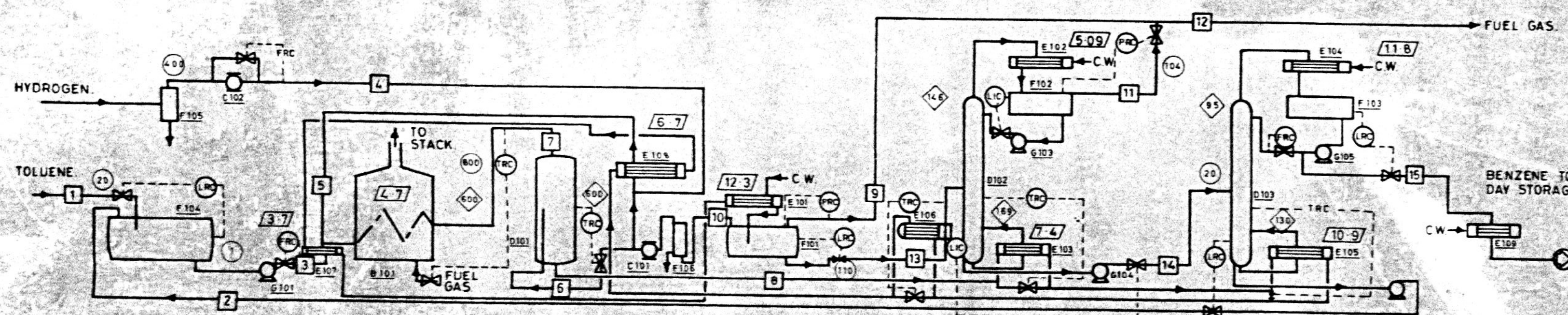
Toluene from intermediate storage (F104) is pumped through a preheater to a vapouriser (B101), and then to a vapour phase reactor (D101) where it undergoes the primary reaction.

$$C_6H_5CH_3 + H_2 \rightarrow C_6H_6 + CH_4$$

Conversion is dependent on the operating conditions but is in the region of 70%. In addition some heavy aromatics such as xylene are produced. The hot reactor products pass through the reboilers on both columns, and after passing through a number of heat exchangers, are finally cooled, by the reactor effluent condenser (E101) and passed to the high pressure flash drum (F101). Here the methane and unreacted hydrogen are taken off. The reactor products consisting mainly of benzene, toluene and xylene, are preheated and passed to the stabiliser tower (D102) where any remaining methane and hydrogen is removed. The products then pass to the benzene tower (D103) where the benzene is distilled off overhead, the remaining toluene and small amount of xylene re-cycling to the storage tank F104.

The first step in the layout procedure is to estimate the diameter of each pipeline. This may be accomplished by any standard technique. Appendix II presents an economic method of pipe size determination.

STREAM NUMBER.		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15	
STREAM NAME.		TOLUENE FEED		RECYCLE TOLUENE		VAPOURISERLIQUID		HYDROGEN FEED		VAPOURISER GAS		QUENCH GAS		REACTOR FEED		REACTOR PROD'TS.		FUEL GAS.F-101		RECYCLE GAS		STAB. FUEL GAS		FUEL GAS		STABILIZER FEED		BENZENE TOWER		BENZENE PRODUCT	
COMPONENT	Mol. wt.	lb.mols/hr.	lb/hr.	lb.mols./hr.	lbs/hr.	lb.mols./hr.	lbs./hr.	lb.mols./hr	lbs/hr.	lb.mols./hr.	lbs./hr.	lb.mols./hr	lbs/hr.	lb.mols./hr	lbs/hr.	lb.mols./hr	lbs./hr.	lb.mols/hr	lbs/hr.	lb.mols/hr	lbs/hr.	lb.mols/hr.	lbs/hr.	lb.mols/hr	lbs/hr.	lb.mols/hr	lbs./hr.	lb.mols/hr	lbs./hr.	lb.mols/hr.	lbs./hr.
BENZENE	78			0·4	28		28							0·4	28	244	19028						100				19028	243	18928		18900
TOLUENE	92	244	22400	60	5592		27992							304	27992	60·8	5600										5600	60·8	5600		8
HEAVY AVS.	110			7·6	830		830							7·2	830	7·6	830										830	7·6	830		
TOTAL LIQUID		244	22400	68	6450		28850							312	28850	312	25458										25458	311·4	25358		18908
HYDROGEN	2							635	1270		2480		560	1520	3040	1276	2552		712		1770		70		782		70				
METHANE	16							35	550		8640		1940	662	10580	906	14484		4054		10030		400		4454		400				
TOTAL GAS								670	1820		11120		2500	2182	13620	2182	17036		4766		11800		570	670	5236						
TOTAL FLOW		244	22400	68	6450		28850	670	1820		11120		2500	2494	42470	2494	42494		4766		11800		570	670	5236				25358		18908



F 104 TOLUENE STORAGE TANK

F 105 MAKE-UP GAS K.O. POT

C 102 MAKE-UP GAS COMPRESSOR

G 101 TOLUENE CHARGE PUMP

E 107 TOLUENE PREHEATER

B 101 VAPOURISER

D 101 REACTOR.

E 108 RECYCLE GAS/ REACTOR EFFLUENT EXCHANGER.

C 101 RECYCLE GAS COMPRESSOR.

F 106 RECYCLE GAS K.O. POT

E 101 REACTOR EFFLUENT FINAL CONDENSER.

F 101 HIGH PRESSURE FLASH DRUM.

E 106 STAB. FEED/ REACTOR EFFLUENT

D 102 STABILIZER

E 102 STABILIZER CONDENSER

F 102 STABILIZER ACCUMULATOR

F 103 STABILIZER REBOILER.

G 103 STAB. REFLUX PUMP.

G 105 BENZENE TOWER REFLUX PUMP.

G 104 BENZENE TOWER FEED PUMP.

E 104 BENZENE OH'D. CONDENSER.

F 103 BENZENE OH'D. ACCUMULATOR

D 103 BENZENE TOWER.

E 105 BENZENE TOWER REBOILER.

G 106 TOLUENE RECYCLE PUMP.

E 109 BENZENE PRODUCT COOLER.

(600) PRESSURE P.S.I.G.

◇95◇ TEMPERATURE °C.

/6·7/ MM. B.T.U. /hr.

PROCESS FLOW DIAGRAM TOLUENE HYDROCRACKER.

DRAWING No. 1001-1101

Knowing the operating conditions and the material being handled, the pipeline material can be decided and the cost per unit length of the pipeline then estimated.

The erected cost per foot of piping can be approximated by

$$\text{cost} = YD_i^{\,n}$$

where D_i is the diameter in inches

for stainless steel	$Y = 3.12$	$n = 1.2$
and		
for carbon steel	$Y = 0.83$	$n = 1.3$

The erected costs of the pipelines may therefore be estimated as in Table 3.

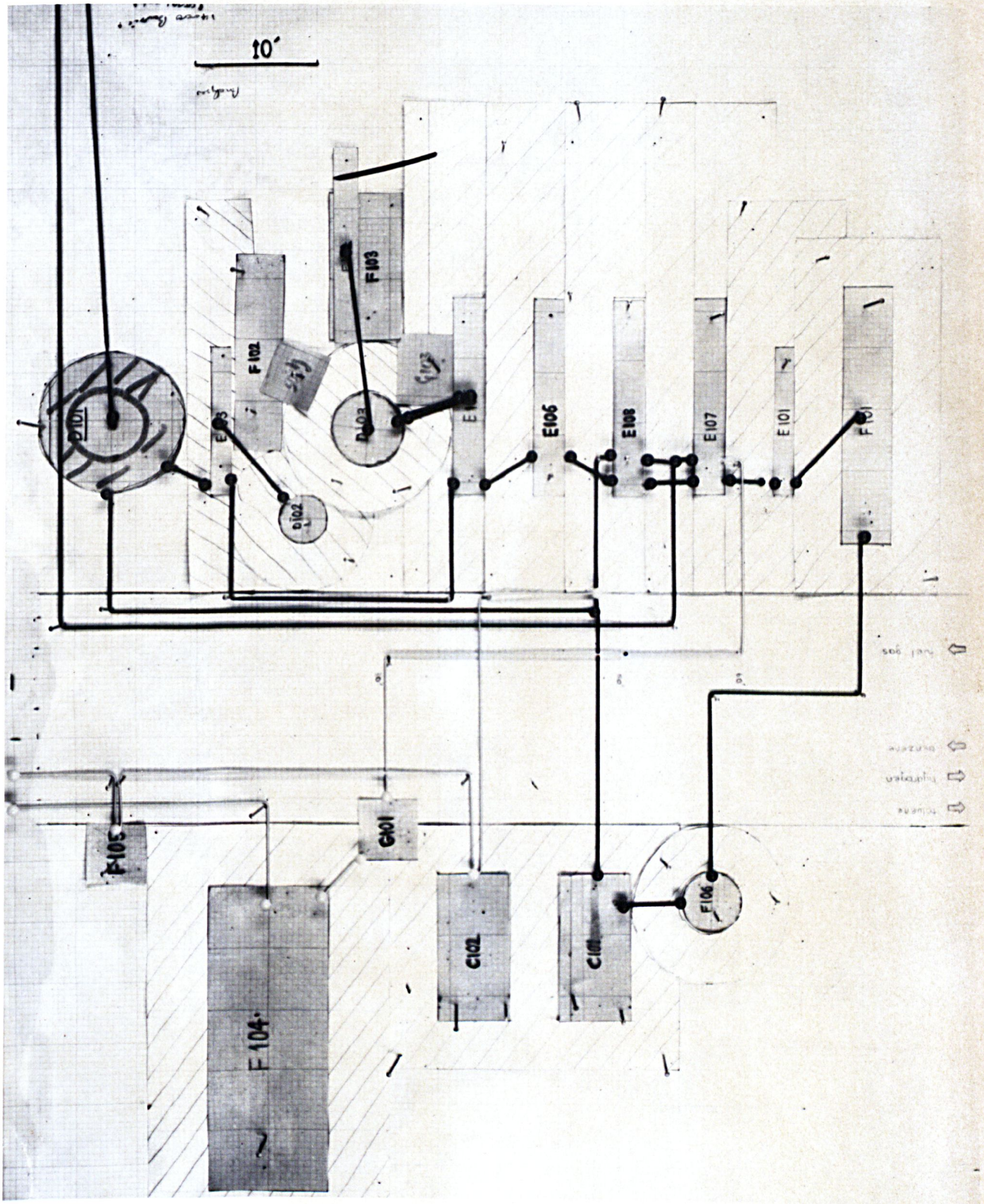
The layout may now be efected using the string diagram analogue. It can be seen that lines 7 and 8 are the most expensive and it is therefore important that these be as short as possible. The pipelines may be arbitrarily grouped according to cost and a colour code used to designate each grouping. For example, the lines may be split into three groups. The most expensive coloured black, being those lines with an erected cost greater than 20 pounds/foot. The intermediate lines, coloured red, having an erected cost between 5 and 20 pounds/foot, and lastly the least costly lines, pale blue, costing less than 5 pounds/foot.

Each unit is then sized and a cutout constructed. The access area required for each unit (Table 2) must be shown on each cutout. The layout then proceeds as in section 3.3. A photograph of this layout is shown

Stream Number	Pipe Diameter Inches	Material --Steel	Cost Pounds/Foot
1	3	Carbon	3.7
2	2	"	2.1
3	3	"	3.7
4	2	"	2.1
5	3	Stainless	12.5
6	2	"	7.2
7	6	"	27.2
8	6	"	27.2
9	2	Carbon	2.1
10	3	Stainless	12.5
11	2	Carbon	2.1
12	8	"	13.1
13	3	"	3.7
14	3	"	3.7
15	2	"	2.1

TABLE 3

overleaf in Figure 3. The first layout is accomplished with the units located at grade. The next step is to consider whether stacking is advantageous. The representation of a layout, with a large number of stacked units, using the string diagram is not good. Better visualisation is achieved by building a rough model; the model being supported on a perspex sheet over the string diagram.


10'
D101
F102
F103
D103
D102
E106
E108
E107
E101
F101
F105
G101
F104
C102
C101
F106

CHAPTER 4

COMPUTER AIDED PLANT LAYOUT

4.1 Available Programs

CRAFT

Computerised Relative Allocation of Facilities Technique. This program uses a simulation algorithm which arranges departments in a given plant area. Input into the algorithm includes the area of the individual departments or plant units, handling volume in unit loads moved between departments and handling costs per unit load per unit distance. The unit cost is established in advance and is independent of the plant arrangement; which is often not true in practice. The program determines distances between departments in any arrangement and from this using the handling cost matrix, seeks minimum cost for exchanges of department locations. The exchange is repeated until the minimum cost is found.

ALDEP

Automated Layout Design Program. This program applies a heuristic programming method to create block layouts. Input to the program includes the area requirements of each department or unit and a preference table which is a matrix of weighting factors which indicate the desirability of units to be near each other. These weighting factors can be given values by the user to represent his view of their relative importance. The program will lay out up to three floors. It performs a two step process for each floor of the building - first any non-assigned departments area is allocated to specific floors and second, those departments assigned to a floor are given a specific location on that floor. A department is randomly selected and processed, then the preference table for that department is searched to find any department with a preference of highest priority which is then selected. When all departments have been processed the resulting layout is scored and printed out. By selecting block layouts with highest scores the best layout are obtained for further analysis.

CORELAP

Computerised Relationship Layout Planning. This program requires input in the form of a Relationship chart as described by Muther,[8] the departmental area restrictions, the size of the unit square to be manipulated and the maximum ratio of building length to width. The main algorithm consists of a heuristic program which add departments in a logical fashion to generate a block plan layout. The output consists of a matrix print-out representing the block plan.

A program for the layout of plant modules with interconnecting piping has been described by Gunn[6]. The objective function considered is the cost of the interconnecting piping and the cost of the building. For each module co-ordinates of the nozzles to which piping is attached are input to the program together with the datum co-ordinates of the modules, pipe descriptions and costs per unit length of pipe. The co-ordinates of the plant modules are arranged so that the objective function is minimised. Minimum and maximum constraints are set on the co-ordinates of the modules so that they do not occupy the same volume in space. Accurate positions of the nozzles and plant modules are available at the end of the computation and may be used in further design work.

4.2 Computer Graphics

The computer graphics program about to be described was developed over a nine month period while the author was seconded to I.C.I. It will be necessary to describe the system that I.C.I. hoped to develope in the Engineering Computer Applications Group of Central Management Services in Wilmslow, Cheshire.

The long term objective of the graphics development program was to produce an integrated system of computer aided design for process plant. The design process involves a large and complex exercise in information set-up, exchange, retrieval and processing, and inevitably there is duplication, loss, late delivery and misinterpretation of information. This is one field in which a graphics system may bring about major improvements and benefits.

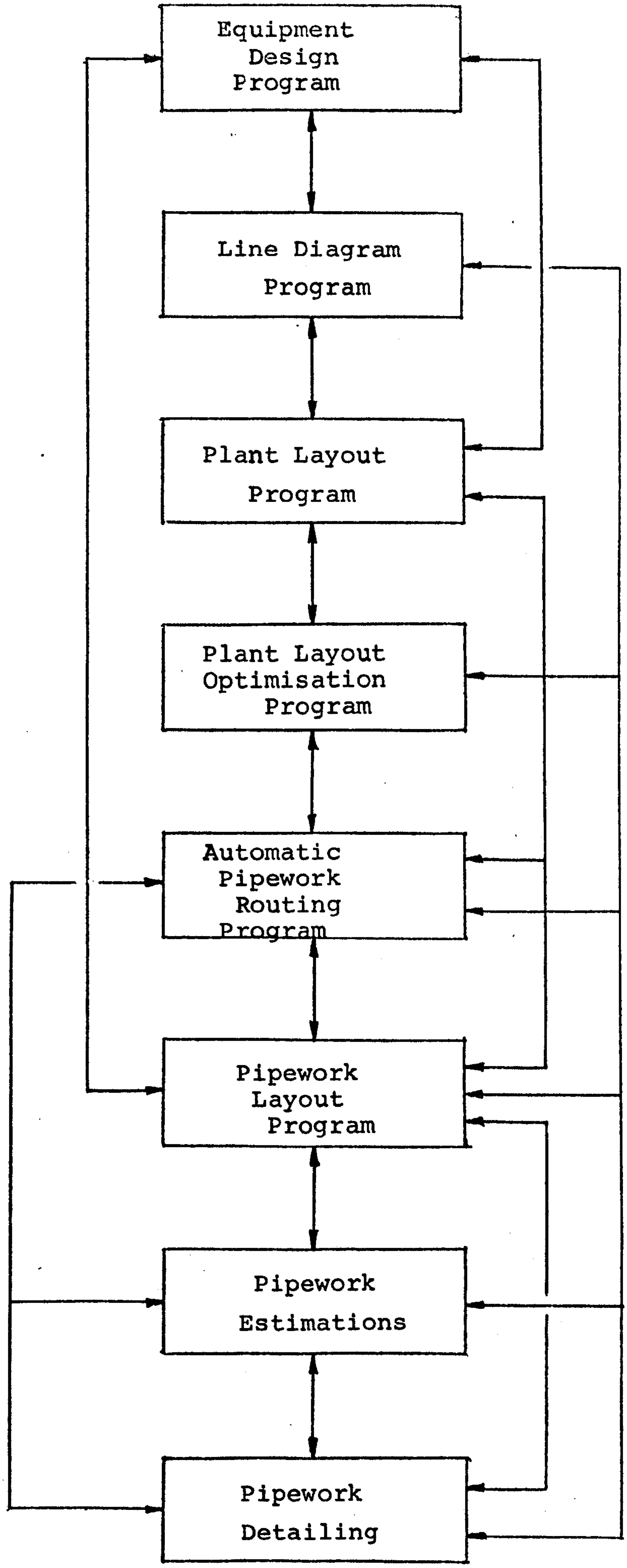
Drawings and schedules are the principal means which engineers and designers use to communicate their ideas to one another, and to contractors who are to build a chemical plant to their specification. To achieve this sort of output by normal computing methods would be a virtually impossible task in data preparation and entry in order to provide reasonable response times. The use of visual display units with suitable graphics software will provide an inter-active means for the designer to communicate with the computer, making use of its ability to handle repetitious calculations and data processing tasks while permitting the designer to use his experience and judgement to control and modify the process. Significant benefits are envisaged in terms of improved quality and accuracy of design, reduced project time span, reduction in design manpower, and more effective use of the creative abilities of designers.

To improve the methods by which chemical plants are designed one must cut down the effort required in specifying designs, and organise the information handling to arrive at a fast and integrated system. In order to make full use of the computer its efficiency must be brought right to the designer's desk and the data must be presented in a quickly assimilated manner. This can best be achieved by computer graphics.

Figure 5 shows the general flow of information between the system programs.

The plant layout has a considerable effect on the other design activities providing initial estimates of piping and structural requirements, approximate branch and support positions for equipment design, and influencing the process requirements in the line diagram. It is therefore important to arrive at a basic plant arrangement as soon as possible. Therefore, work on the overall layout is possible as soon

Fig. 5



Information flow

Between System Programs

as the process data is inserted in the system. At the same time work may go on to clarify process requirements in the engineering line diagram, so that the means of connection between vessels (piperun data) will also be accumulating. When sufficient data is available from the plant layout and line diagram activities, application programs would be accessed to provide simple automatic piperouting and hence arrive at a cost estimate of the piping requirements. Various alternative shcemes may then be tried to improve the layout, using the display aided by plant layout optimisation routines.

The work being undertaken by the author at the University of Sheffield was judged to be in accordance with that proceeding at I.C.I. Central Management Services in Wilmslow. In November, 1970 the author joined the staff at C.M.S. to work on the development of the Plant Layout program of the system. The resulting program was developed over the next nine months during which time the project was postponed by I.C.I. . Computer personnel were engaged on other work and the author continued the work alone.

4.3 The Graphics Program

It was decided that the program must satisfy two basic requirements. Firstly there must be a continuous visual indication of the state of the layout. This is necessary because although the objective function being minimised may be calculated, it cannot be defined in terms of all the constraints imposed on the system. This is what makes layout essentially a visual exercise and the operator must therefore monitor the progress of the layout. Secondly the cost of different layout variations must be instantly available. These requirements are based on the assumption that it is impossible to quantify or even define all the constraints and considerations that affect the cost of the layout. The program is therefore written in such a way that it is possible for the operator to determine completely the progression of the layout. It is possible to move scaled representations of the plant items around on the screen independently of one another. Pipework is automatically inserted orthogonally, but the user is able to change the pipeline positions and routes and to redefine the item nozzle positions at will. It is important that the user should be able to file or putput the data defining the current state of the layout so that a layout may be modified on successive runs over a period of time. This means that the data structure is such that the co-ordinates of all items and pipelines are continously filed and updated as the layout progresses.

The basic flowchart of the program developed is shown in Figure 6. A description of the program follows.

4.4 Plant Item Specification

It is necessary to categorise the plant units. The space requirements for maintenance etc., can then be defined for each type of unit, and an indication can be given of the type of duty performed by

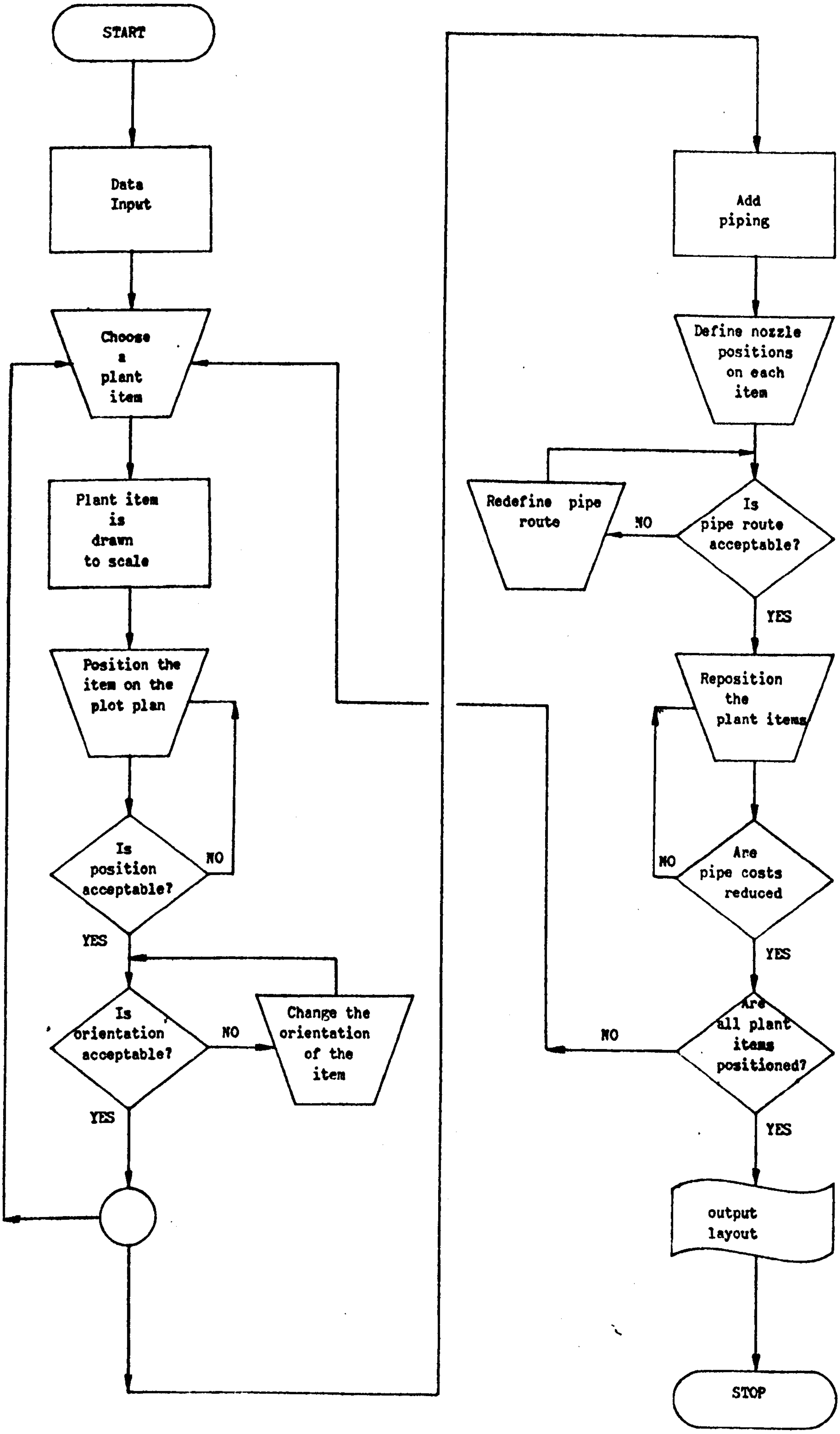


Fig. 6 - Basic Flow Chart

each unit. The two letter code specification used within the program is shown in Table 4 below.

AA	Agitators, Stirrers
BA	Blenders, Crushers, Mills, Screens, etc.
CA	Packed Columns
CB	Plate Columns
DA	Driers
EA	Evaporators, Crystallisers, Flakers
FA	Filters, Centrifuges
HA	Air Coolers
HE	Heat Exchangers
HR	Reboilers
HC	Condensers
JA	Furnaces, Vaporisers, Boilers
KA	Compressors, Fans, Blowers
PA	Pumps
RA	Reactors, Kilns
TA	Tanks, Drums, Receivers, Hoppers, Accumulators

TABLE 4

PLANT ITEM CODING

The minimum space requirements may then be defined for each item type (Table 5), using the conventional axes of Figure 7.

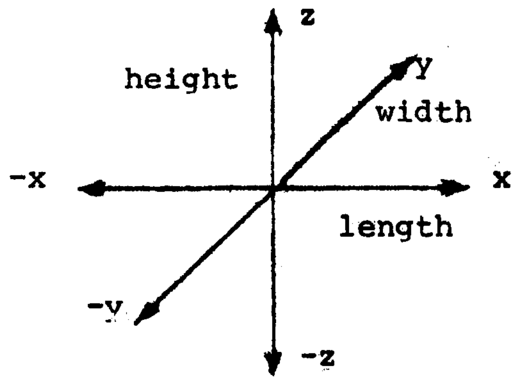


Fig. 7

AA	5	5	5	5	0	5
BA	6	6	5	5	0	6
CA	5	5	5	5	5	15
CB	5	5	5	5	5	15
DA	5	5	5	5	0	6
EA	5	5	5	5	0	6
FA	10	10	10	10	0	6
HA	6	6	4	4	3	3
HE	6	6	4	4	3	3
HB	6	6	4	4	3	3
HC	6	6	4	4	3	3
JA	10	10	10	10	0	∞
KA	5	5	5	5	0	10
PA	6	6	4	4	0	6
RA	5	5	5	5	0	10
TA	4	4	4	4	3	4

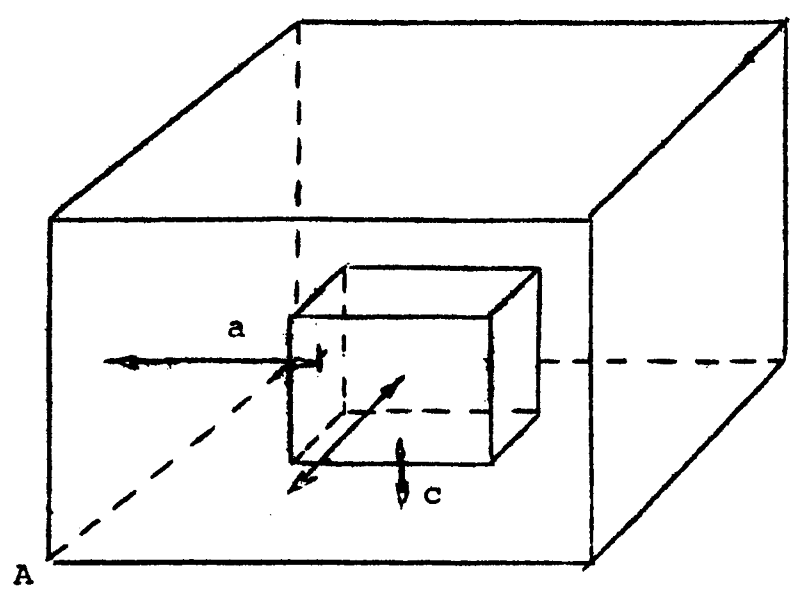
TABLE 5

ITEM SPACING REQUIREMENTS - FEET

The items are represented diagrammatically by three simple shapes, - cuboids, spheres or cylinders. These reduce to either circles or rectangles when the layout is presented in plan or elevation. The specification code of an item type together with its shape on the screen, facilitates its identification.

The size of the item is defined relative to an origin such that the unit lies along the positive axes of Figure 7. The origin of a sphere is taken to be at its centre. In representing the plant items on the screen it is of major importance to also show the space requirements of the units. These may be defined relative to the origin of each item. The co-ordinates of the origin of the cuboid defining the space requirements of the item must be calculated relative to the origin of the item itself.

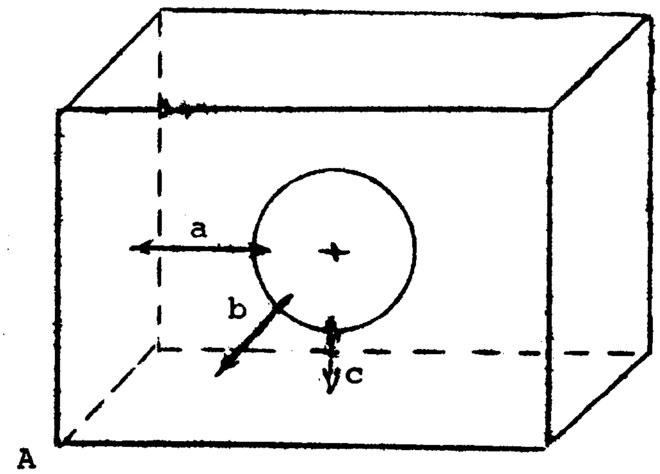
Cuboid (length, width, height)



the co-ordinates of A are

(-a, -(b + width/2), - (c + height/2))

Sphere (diameter)

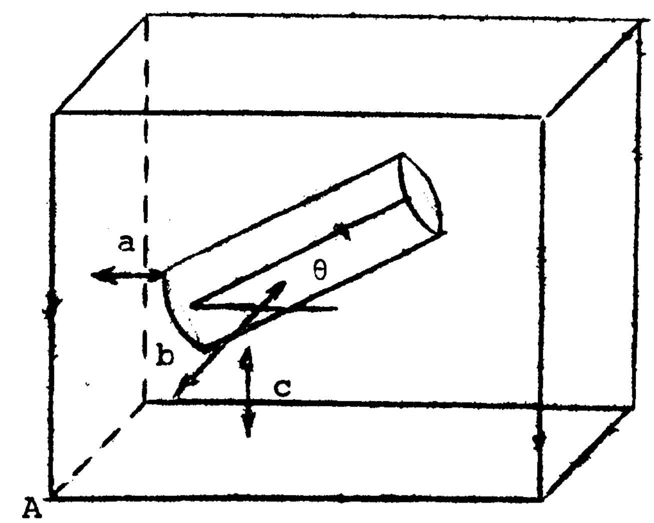


the co-ordinates of A are

(-(a + diameter/2), - (b + diameter/2),

- (c + diameter/2))

Cylinder (diameter, length, angle)



the co-ordinates of A are

(-(a + diameter x sinθ/2), - (b + diameter/2),

-(c + diameter x cosθ/2))

For a horizontal cylinder the co-ordinates are

(-a, -(b + diameter/2), - (c + diameter/2)) and for a vertical cylinder

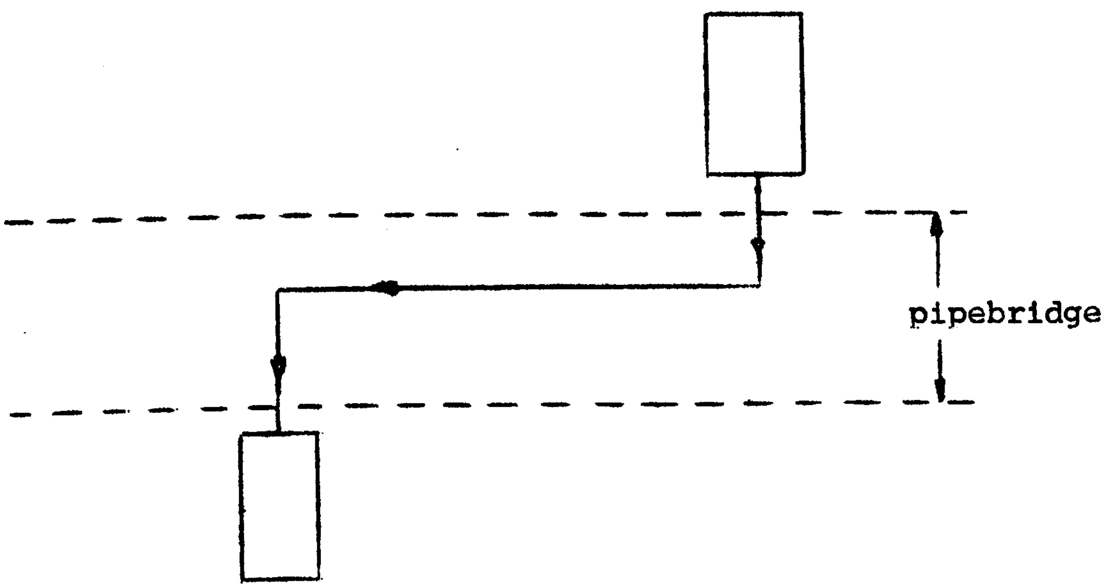
(-(a + diameter/2), - (b + diameter/2), - c)

Although the units have been defined here three-dimensionally, they are presented on the screen in plan only. The cylinder model only may be angled to the horizontal.

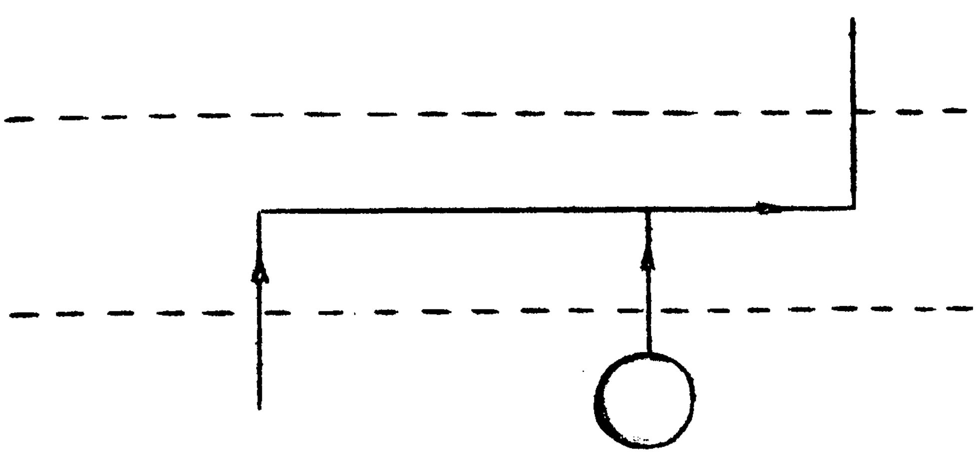
4.5 Pipework Representation

Five types of pipeline are recognised:-

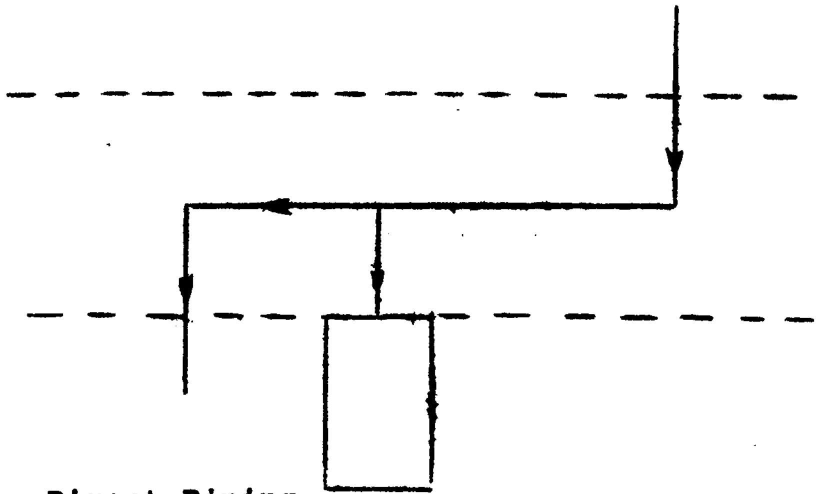
1. Vessel to Vessel Piping



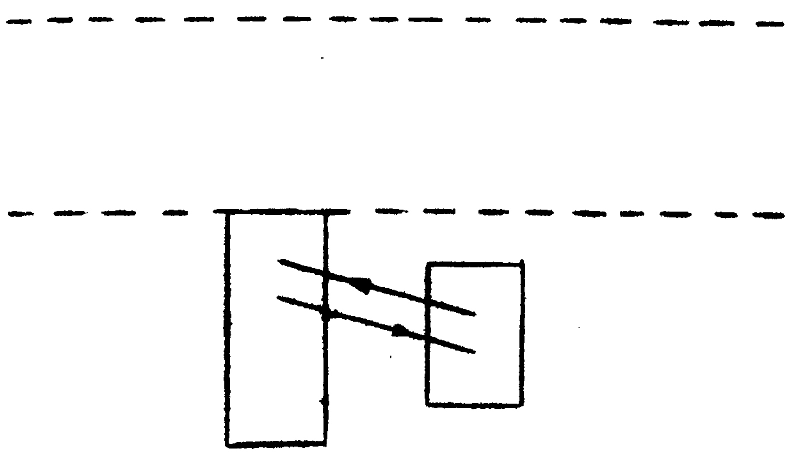
2. Vessel to Pipe Piping



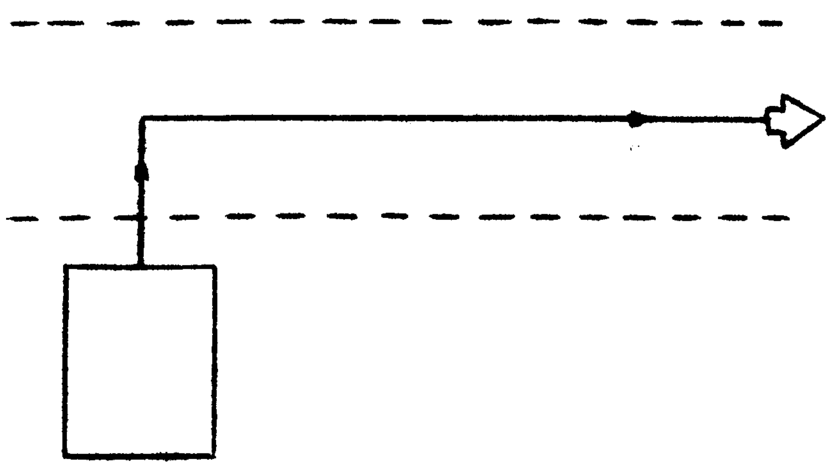
3. Pipe to Vessel Piping



4. Direct Piping



5. Offsite Piping



4.6 Input and Output Procedure

Four arrays must be filled before the program can commence. The arrays contain all information detailing plant item numbers, types, deminsions, clearances and pipeline routes, necessary to define the layout completely. These arrays are:-

DIMENS - containing the dimensions of each item.

ICODE - containing the number of each item type.

CLEAR - containing the clearances associated with each item.

NOZZLE - containing data defining the pipe routes.

When the layout has been completed, information that enables the layout to be defined absolutely is to be found in two arrays. These arrays are

UPDATE - containing the absolute positions of each item

IPIPE - containing the absolute positions of each pipeline.

These six arrays are now described in greater detail The printouts of these arrays shown in Tables 6 - 10 pertain to the petrochemical plant of section 4.8

Array DIMENS

This array consists of five columns the dimensions of each unit being located in each row.

Column	
1	The item length
2	The item width
3	The item height
4	The item diameter
5	The angle of the item to the horizontal.

The dimensions are listed in sixteenths of an inch. A printout of array DIMENS is shown in Table 6.

Array ICODE

This array consists of two columns. The reference code letters of the item type in column one and the number of items of each type in column two.

Array CLEAR

This array contains all the clearances associated with each item. There are six columns: two columns for each orthogonal axis. A printout of array CLEAR is shown in Table 7.

Array NOZZLE

This array defines the pipework completely and is taken from the Process Flow Diagram. There are four columns:

Column

1. The reference number of the item from which the pipeline runs.
2. The reference number of the item to which the pipeline runs.
3. An integer defining the brightness of the pipe when shown on the screen.
4. A relative cost figure per unit length of pipeline.

In addition the following code is used. The number one refers to offsite piping. Item reference numbers lie between 100 and 100 + the number of items. Piping reference numbers are equal to the row in array NOZZLE defining the pipe plus one thousand. Thus a number greater than one thousand in array NOZZLE means that the pipeline runs either to or from another line. A printout of array NOZZLE is shown in Table 8.

Array UPDATE

This array, which is filled and updated as the layout proceeds, contains sufficient data to enable the reconstruction of the plant items in their correct positions. It consists of fifteen columns.

Column

1. The process number of the item.
2. The specification code letters of the item.
3. The screen reference number of the item.
4. The x - co-ordinate of the item origin.
5. The y co-ordinate of the item origin.
6. The width of the item in the screen x direction (in screen increments).
7. The length of the item in the screen y direction (in screen increments).
8. The x co-ordinate of the access box origin with respect to the item origin.
9. The y co-ordinate of the access box origin with respect to the item origin.
10. The width of the access box in the screen x direction.
11. The length of the access box in the screen y direction.
12. The diameter of the item in screen increments.
13. An integer set -1,0, or 1 denoting an item represented by a cube, sphere or cylinder respectively.

14. The angle of the item to the horizontal.

15 An integer denoting whether the item is drawn with solid, dotted or dashed lines.

A printout of array UPDATE is shown in Table 9.

Array IPIPE

This array, which is filled and updated as the piperouting proceeds, contains sufficient data to enable the reconstruction of the plant pipework in the correct position. It consists of eleven columns.

Column

1. The reference number of the pipe equal to 1000 plus the row in array IPIPE defining the route.

2- These locations contain the co-ordinates of
5 the end points of the lines.

6. The level of the pipeline in the piperack.

7- The locations containing the co-ordinates of
10 the end points of the lines.

11. The pipeline type.

A printout of array IPIPE is shown in Table 10.

All six arrays can be input and output on paper tape.

4.7 Layout Routines

The program is modular in form and comprises some fifty subprograms. These routines are described in detail below

SUBROUTINE ARROW (BUP,BUF,K)

ARRAY DIMENS

			L	W	H	D	THETA
1	CB	101	11520	0	0	960	90
2	CB	102	11520	0	0	1152	90
3	HE	101	3072	0	0	576	0
4	HE	102	2304	0	0	384	0
5	HE	103	3072	0	0	576	0
6	HE	104	3072	0	0	576	0
7	HR	101	3072	0	0	576	0
8	HR	102	3072	0	0	576	0
9	HC	101	2304	0	0	384	0
10	HC	102	3072	0	0	576	0
11	HC	103	3072	0	0	576	0
12	JA	101	2304	0	0	1920	90
13	KA	101	1920	1152	1152	0	0
14	KA	102	1920	1152	1152	0	0
15	PA	101	576	384	768	0	0
16	PA	102	576	384	768	0	0
17	PA	103	576	384	768	0	0
18	PA	104	576	384	768	0	0
19	PA	105	576	384	768	0	0
20	RA	101	3072	0	0	1152	90
21	TA	101	3072	0	0	1152	0
22	TA	102	1920	0	0	768	0
23	TA	103	2304	0	0	1536	0
24	TA	104	3840	0	0	7680	90
25	TA	105	1152	0	0	768	90
26	TA	106	1152	0	0	768	90

ARRAY CLEAR

			L1	L2	W1	W2	H1	H2
1	CB	101	0	0	0	0	0	0
2	CB	102	0	0	0	0	0	0
3	HE	101	384	3072	0	0	0	0
4	HE	102	384	2304	0	0	0	0
5	HE	103	384	3072	0	0	0	0
6	HE	104	384	3072	0	0	0	0
7	HR	101	384	3072	0	0	0	0
8	HR	102	384	3072	0	0	0	0
9	HC	101	384	2304	0	0	0	0
10	HC	102	384	3072	0	0	0	0
11	HC	103	384	3072	0	0	0	0
12	JA	101	0	0	0	0	0	0
13	KA	101	0	0	0	0	0	6
14	KA	102	0	0	0	0	0	6
15	PA	101	0	0	0	0	0	0
16	PA	102	0	0	0	0	0	0
17	PA	103	0	0	0	0	0	0
18	PA	104	0	0	0	0	0	0
19	PA	105	0	0	0	0	0	0
20	RA	101	0	0	0	0	0	0
21	TA	101	0	0	0	0	0	0
22	TA	102	0	0	0	0	0	0
23	TA	103	0	0	0	0	0	0
24	TA	104	3072	3072	3072	3072	0	0
25	TA	105	0	0	0	0	0	0
26	TA	106	0	0	0	0	0	0

ARRAY NOZZLE

1	1	125	1	1
2	125	113	1	1
3	1	124	1	1
4	124	115	1	1
5	115	103	1	1
6	103	112	1	1
7	112	120	3	3
8	120	107	3	3
9	107	108	3	3
10	108	106	3	3
11	106	105	3	3
12	105	103	3	3
13	103	111	3	3
14	111	121	3	3
15	121	126	2	2
16	126	114	2	2
17	114	120	2	2
18	1017	105	2	2
19	105	1006	2	2
20	113	1018	1	1
21	121	1	2	2
22	121	106	1	1
23	106	101	1	1
24	101	109	3	3
25	109	122	1	1
26	122	116	1	1
27	116	101	1	1
28	122	1021	1	1
29	101	107	1	1
30	107	101	3	3
31	101	117	1	1
32	117	102	1	1
33	102	110	3	3
34	110	123	1	1
35	123	118	1	1
36	118	102	1	1
37	1036	104	1	1
38	104	1	1	1
39	102	108	1	1
40	108	102	3	3
41	102	119	1	1
42	119	124	1	1

2 KEY 1

1	101HR	107	1392	1203	0	160	55	-80	110	460	30	1	0	1
2	101CB	101	1472	1196	0	600	75	-75	150	150	50	1	90	1
3	101JA	112	1175	1597	0	-120	-150	150	-300	-300	-100	1	90	1
4	101RA	120	1287	1203	0	160	80	-80	160	160	60	1	90	1
5	102CB	102	1576	1202	0	600	80	-80	160	160	60	1	90	1
6	102HR	108	1672	1202	0	160	55	-80	110	460	30	1	0	1
7	104HE	106	1744	1204	0	160	55	-80	110	460	30	1	0	1
8	103HE	105	1815	1204	0	160	55	-80	110	460	30	1	0	1
9	101HE	103	1886	1204	0	160	55	-80	110	460	30	1	0	1
10	103HC	111	1957	1201	0	160	55	-80	110	460	30	1	0	1
11	101TA	121	2044	1163	0	160	70	-40	140	240	60	1	0	1
12	102TA	122	1374	1162	0	100	60	-40	120	180	40	1	0	2
13	103TA	123	1697	1167	0	120	80	-40	160	200	80	1	0	2
14	104TA	124	1424	524	0	-200	-400	400	-800	-800	-400	1	90	1
15	106TA	126	2079	864	0	-60	-60	60	-120	-120	-40	1	90	1
16	101KA	113	2146	1174	60	100	80	-50	160	200	0	-1	0	1
17	101HC	109	1384	1201	0	120	50	-80	100	380	20	1	0	4
18	102HC	110	1673	1213	0	160	55	-80	110	460	30	1	0	4
19	102KA	114	1977	874	-60	-100	-80	50	-160	-200	0	-1	0	1
20	101PA	115	1886	873	-20	-30	-60	50	-120	-130	0	-1	0	1
21	102PA	116	1427	1091	20	30	60	-50	120	130	0	-1	0	1
22	103PA	117	1500	1094	20	30	60	-50	120	130	0	-1	0	1
23	104PA	118	1640	1092	20	30	60	-50	120	130	0	-1	0	1
24	105PA	119	1570	1095	20	30	60	-50	120	130	0	-1	0	1
25	105TA	125	2247	1182	0	60	60	-60	120	120	40	1	90	1
26	102HE	104	2317	1203	0	120	50	-80	100	380	20	1	0	1

1007	1175	1597	0	0	0	0	0	1287	1203	4
1008	1317	1207	0	0	0	0	0	1377	1220	4
1009	1408	1221	1417	1221	1123	1647	1220	1657	1220	1
1029	1476	1207	0	0	0	0	0	1394	1292	4
1030	1395	1280	0	0	0	0	0	1472	1196	4
1010	1690	1219	0	0	0	0	0	1731	1237	4
1011	1760	1333	0	0	0	0	0	1802	1350	4
1023	1730	1220	1719	1220	1113	1472	1196	1472	1196	1
1039	1582	1199	0	0	0	0	0	1676	1274	4
1040	1678	1289	0	0	0	0	0	1576	1202	4
1006	1871	1234	1861	1234	1103	1175	1546	1175	1546	1
1012	1827	1219	0	0	0	0	0	1874	1217	4
1019	1831	1331	1840	1331	1103	0	0	0	0	2
1013	1901	1349	0	0	0	0	0	1945	1347	4
1014	1972	1216	0	0	0	0	0	2047	1218	4
1021	1	-5	2044	1163	1093	1073	0	0	0	5
1022	2029	1164	2029	1153	1123	1769	1351	1763	1351	1
1025	1387	1203	0	0	0	0	0	1374	1162	4
1028	1374	1162	1374	1152	1093	0	0	0	0	2
1034	1673	1213	0	0	0	0	0	1697	1167	4
1003	2	5	1424	524	923	1073	0	0	0	5
1015	2064	1167	2064	1153	993	2079	874	2079	864	1
1016	2062	863	0	0	0	0	0	2008	861	4
1024	1467	1193	0	0	0	0	0	1386	1263	4
1033	1577	1213	0	0	0	0	0	1676	1299	4
1017	1960	874	1960	884	1073	1288	1176	1288	1176	1
1018	1802	1232	1790	1232	1073	0	0	0	0	3
1020	2146	1174	2146	1164	1113	1790	0	0	0	2
1004	1589	635	0	0	0	0	0	1889	864	4
1005	1886	873	1886	883	1003	1911	1331	1900	1331	1
1026	1374	1162	0	0	0	0	0	1429	1114	4
1027	1427	1091	0	0	0	0	0	1455	1177	4
1031	1483	1178	0	0	0	0	0	1499	1117	4
1032	1500	1126	0	0	0	0	0	1561	1179	4
1035	1697	1167	0	0	0	0	0	1640	1115	4
1036	1640	1092	1640	1082	1083	1603	1183	1603	1183	1
1041	1576	1202	0	0	0	0	0	1571	1126	4
1042	1570	1095	1570	1085	1003	1424	534	1424	524	1
1001	2	-5	2247	1182	1123	3019	0	0	0	5
1002	2229	1186	0	0	0	0	0	2178	1186	4
1037	2308	1308	2297	1308	1083	1640	0	0	0	2
1038	1	5	2317	1203	1113	3019	0	0	0	5

FUNCTION:

This routine draws a solid arrow at the end of a pipeline showing that the pipeline carries a feed or product stream across the plot limits.

PARAMETERS:

BUP Brightness control.

BUF The array into which the code generating the item or the screen is stored.

K An integer which controls both the size of the arrow and its direction.

SUBROUTINE BOX (ITNAME,L,W,H,D,THETA,ECLEAR,BOXEXT, SCALE KFORM)

FUNCTION:

This routine calculates the size of the box surrounding the unit required for access, within which other plant units may not intrude. The co-ordinates of the origin of the box are calculated relative to the plant item as described in section 4.4.

PARAMETERS:

KNAME	The two letter word defining the item type.
L	The item length.
W	The item width.
H	The item height.
D	The item diameter.
THETA	The angle of the major axis of the item to the horizontal.
ECLEAR	An array containing six integers being the row of the array CLEAR containing the access requirements of the unit.

BOXEXT	An array containing the dimensions of the item box and the co-ordinates of the box origin relative to the item origin.
SCALE	An integer scale factor to convert the dimensions in array DIMENS to feet.
KFORM	An integer indicating the type of item representation cube KFORM = -1 sphere KFORM = 0 cylinder KFORM = 1

<u>SUBROUTINE BOXIN (UPDATE,XBASE,YBASE,BASE,BUF)</u>

<u>FUNCTION</u>:

To replace on the screen the boxes defining the access requirements of each item.

<u>PARAMETERS</u>:

UPDATE	An array holding the name, dimensions and position of each item. See 4.6.
XBASE, YBASE	The co-ordinate of the screen origin with respect to the origin of the area being scissored.
BASE	An array containing the code used in SUBROUTINE ENTER for instantaneous scissoring using subpictures.
BUF	see ARROW

<u>SUBROUTINE BOXOUT (UPDATE,XBASE,YBASE,BASE,BUF)</u>

<u>FUNCTION</u>:

To delete from the screen the boxes defining the access requirements of each item. The items themselves remain.

PARAMETERS:

see BOXIN

SUBROUTINE DISBOX (EXORX,EXORY,EXINCX,EXINCY,JSHINE)

FUNCTION:

To draw a rectangle surrounding a plant unit indicating the extent of its access space requirements.

PARAMETERS:

EXORX, EXORY.	The co-ordinates of the origin of the box relative to the origin of the item.
EXINCX, EXINCY	The dimensions of the box in the x and y directions.
JSHINE	Brightness control.

SUBROUTINE DISCUB (IX,IY,INCX,INCY,BUF,ITYPE)

FUNCTION:

To draw a rectangle on the screen representing a plant item.

PARAMETERS:

IX,IY	The absolute co-ordinates of the origin of the item.
INCX,INCY	The dimensions of the item in the x and y directions.
BUF	see ARROW
ITYPE	The type of line to be drawn on the screen.

SUBROUTINE DISCYL(IX,IY,L,D,THETA,BUF,ITYPE)

FUNCTION:

To draw a cylinder on the screen in plan. If theta is 90°, a circle is drawn; if theta is zero or greater, a rectangle is drawn.

PARAMETERS:

IX,IY	The absolute co-ordinates of the origin of the item.
L,D,THETA	see BOX
BUF	see ARROW
ITYPE	The type of line to be drawn on the screen.

SUBROUTINE DRAW (JX,JY,KX,KY,YLEVEL,BUP,BUF,BASE, XBASE,YBASE)

FUNCTION:

This routine draws the three arrowed lines of type 1 pipelines (q.v. 4.5.1).

PARAMETERS:

JX,JY	The co-ordinates of the beginning of the first arrowed line.
KX,KY	The co-ordinates of the end of the third arrowed line.
YLEVEL	The Y co-ordinate of the centre line. This marks its route on the piperack.
BUP	Brightness control.
BUF	see ARROW
BASE,XBASE, YBASE	see BOXIN

SUBROUTINE ELVATE (IPIPE,NOZZLE,BASE,BUF,XBASE,YBASE)

FUNCTION:

This subroutine sees a pipeline and changes its position on the piperack. It moves continously across the piperack and is stopped at the required position.

PARAMETERS:

IPIPE see 4.6

NOZZLE see 4.6

BASE,XBASE, see BOXIN
YBASE

BUF see ARROW

SUBROUTINE ERASE (IPIPE,UPDATE,NOZZLE,BUF)

FUNCTION:

This routine sees an item on the screen and deletes it. Arrays UPDATE and IPIPE are updated and the piping costs is recalculated if necessary.

PARAMETERS:

IPIPE see 4.6

UPDATE see 4.6

NOZZLE see 4.6

BUF see ARROW

SUBROUTINE FIGURE (UPDATE,UPROW,IX,IY,XBASE,BASE,BUF)

FUNCTION:

This subroutine generates the code for the plant item and also the lettering shown above the unit.

PARAMETERS:

UPDATE see 4.6

UPROW The row in array UPDATE containing the relevant data.

IX,IY The absolute co-ordinates of the item.

XBASE,YBASE,see BOXIN
BASE,

BUF see ARROW

SUBROUTINE FIRST (NOZZLE,UPDATE,IPIPE,I,J,K,IL,M,N, XBASE,YBASE,BASE,BUF,YTRAK)

FUNCTION:

This subroutine scans through array NOZZLE and checks each pipeline against the parameters in array UPDATE which contain information about each item on the screen. When it finds that a pipeline may be displayed it exits and the piping routines are entered.

PARAMETERS:

NOZZLE,UPDATE IPIPE	see 4.6
I	The row in array UPDATE containing the parameters defining the item from which the pipeline runs.
J	The row in array NOZZLE defining the pipe.
K	The row in array UPDATE containing the parameters defining the item to which the pipeline runs.
IL	The row in array IPIPE containing the parameters defining the pipeline from which the pipe runs (4.5.3).
M	The row in array IPIPE containing the parameters defining the pipeline to which the pipe runs. (4.5.2).
N	The row in array IPIPE in which will be entered the parameters defining the pipeline to be displayed.
XBASE,YBASE, BASE	see BOXIN
BUF	see ARROW

SUBROUTINE FOX (FACTOR,ESCALE,IFACTO)

FUNCTION:

The subroutine converts the real variables contained in array BXEXT to integers.

PARAMETERS:

FACTOR	A real variable whose dimensions is feet.
ESCALE	A scale factor converting from feet to the dimensions of the variables stored in array DIMENS.
IFACTO	The value of the FACTOR after conversion.

SUBROUTINE INCOST (IPIPE,NOZZLE,BUF)

FUNCTION:

This subroutine calculates the cost of the piping. The cost is calculated each time a new pipe is displayed or a pipeline route altered. Three unit costs are available. The cost is the length of the line in screen increments multiplied by 1, 2 or 3. With a scale (IRATIO) of 10 screen increments per foot, this means the cost is calculated at 10, 20 or 30 units per foot of pipe.

PARAMETERS:

IPIPE,NOZZLE	see 4.6
BUF	see ARROW

SUBROUTINE INITAL (BUF)

FUNCTION:

This routine initialises the piping cost which is continuously replaced by SUBROUTINE INCOST.

PARAMETERS:

BUF see ARROW

SUBROUTINE INPIPE (NOZZLE,UPDATE,IPIPE,YLEVEL,YTRAK,
BUF,BASE,BUP,XBASE,YBASE)

FUNCTION:

This routine is the base routine which handles the piping maneouvres. Together with SUBROUTINE SCREEN it forms the basis of the program.

PARAMETERS:

NOZZLE,UPDATE,see 4.6

YLEVEL	The Y co-ordinate of a pipeline as it runs along the piperack.
YTRAK	The level of the upper boundary of the piperack.
BUF	see ARROW
BASE	see BOXIN
BUP	Brightness control
XBASE,YBASE	see BOXIN

SUBROUTINE INSET (ICODE,XBASE,YBASE,BASE,BUF,ITYPES,
IROW,CODE,NUMBER)

FUNCTION:

This routine displays a list of item types on the screen, allowing the user to pick one with the light pen. It then displays a list of the numbers of the item. When the item number has been chosen in the same way, the row in array DIMENS containing the dimensions of item is calculated.

PARAMETERS:

ICODE	see 4.6
XBASE,YBASE, BASE,	see BOXIN

ITYPES	The number of item types.
IROW	The row in array DIMENS containing the dimensions of the item chosen.
CODE	The item type chosen.
NUMBER	The item number chosen.

SUBROUTINE INTAPE (UPDATE,IPIPE)

FUNCTION:

This routine writes the arrays UPDATE and IPIPE to paper tape. The two arrays hold all the information necessary to regenerate the layout on the screen. They are also used by the plotter program.

PARAMETERS:

UPDATE,IPIPE see 4.6

SUBROUTINE INVERT (UPDATE,UPROW,XBASE,YBASE,BUF)

FUNCTION:

This routine swings an item on the screen through 180 degrees.

PARAMETERS:

UPDATE	see 4.6
UPROW	The row in array UPDATE containing the parameters defining the item being inverted.
XBASE,YBASE	see BOXIN
BUF	see ARROW

SUBROUTINE ITRASE (UPDATE,XBASE,YBASE,BASE,BUF)

FUNCTION:

This subroutine substitutes dotted or dashed lines in the code for the plant items so that the user may emphasise a number of items according to his own convention.

PARAMETERS:

UPDATE	see 4.6
XBASE,YBASE, BASE	see BOXIN
BUF	see ARROW

SUBROUTINE LETTER (UPDATE,I,IX,IY,XBASE,YBASE,BASE,BUF)

FUNCTION:

This routine writes the code letters and item number just above or below the item depending on which side of the piperack the item is located.

PARAMETERS:

UPDATE	see 4.6
I	The row in array UPDATE containing the parameters defining the item.
IX,IY	The absolute co-ordinates of the item origin.
XBASE,YBASE BASE	see BOXIN
BUF	see ARROW

SUBROUTINE LEVEL (YLEVEL,YTRAK,IPIPE,JX,JY,KX,KY,BUF)

FUNCTION:

This routine decides where a pipeline will be positioned on the piperack. The piperack is taken as 20 ft. wide and pipelines are positioned at one foot intervals across the bridge. The routine scans through array IPIPE and checks that two or more pipes do not use the same position.

PARAMETERS:

YLEVEL — The Y co-ordinate of a pipeline as it runs along the pipebridge.

YTRAK — The Y co-ordinate (absolute) of the upper edge of the piperack.

IPIPE — see 4.6

JX,JY — The co-ordinates of the starting point of the pipelin .

KX,KY — The co-ordinates of the finishing point of the pipeline.

BUF — see ARROW.

SUBROUTINE LVEC (X,Y)

FUNCTION:

This routine positions the light beam at a point on the screen. By calling NEWBUF (BASE 8), the beam is positioned at point (0,0). To move the beam to another point on the screen an invisible line is drawn to that point. LVEC breaks down the line into lenghts of 1023 increments which is the longest line the display will draw.

PARAMETERS:

X,Y — The absolute co-ordinates of the point where the light beam is to be positioned.

SUBROUTINE NOPIPE (IPIPE)

FUNCTION:

To delete all piping from the screen and to set array IPIPE to zero.

PARAMETERS:

IPIPE — see 4.6

SUBROUTINE OFSITE (NOZZLE,UPDATE,IPIPE,J,YTRAK,XBASE,
YBASE,BASE,BUF)

FUNCTION:

To display on the screen a pipeline of type 4.5.5

PARAMETERS:

NOZZLE,UPDATE IPIPE	see 4.6
J	The row in array NOZZLE containing the parameters defining the pipeline.
YTRAK	The Y co-ordinate (absolute) of the upper edge of the piperack.
XBASE,YBASE, BASE	see BOXIN
BUF	see ARROW

SUBROUTINE ORDER (KPIPE)

FUNCTION:

To place the elements of array KPIPE in ascending order.

PARAMETERS:

KPIPE	An integer array of length eight.

SUBROUTINE PIMOVE (UPDATE,M,IPIPE,NOZZLE,XBASE,YBASE,
BASE,BUF,YTRAK,KPIPE,XDIF,YDIF,IXX,IYY,IFTURN)

FUNCTION:

This subroutine partners SUBROUTINE REMOVE. The routine calculates the co-ordinates of the new piping nozzles and replaces the relevant pipelines as the item is moved around the screen.

PARAMETERS:

UPDATE	see 4.6
M	The row in array UPDATE containing the parameters defining the item being moved.
IPIPE,NOZZLE	see 4.6
XBASE,YBASE BASE	see BOXIN
BUF	see ARROW
YTRAK	The Y co-ordinate (absolute) of the upper edge of the piperack.
KPIPE	An integer array containing the reference number of all pipelines that need to be replaced.
XDIF,YDIF	The difference between the old and new origin co-ordinates of the item.
IXX,IYY	The original co-ordinates of the item origin.
IFTURN	The value of this variable indicates whether the item has been inverted.

SUBROUTINE PIPE1 (UPDATE,UPROW,IXX,IYY,XO,YO,BUP,BUF, BASE)

FUNCTION:

This subroutine indicates the nozzle positions of the items. A line is drawn a scale one foot perpendicular to the side of the unit. This routine is not called when the item is represented in plan by a circle. The co-ordinates of the end of the nozzle are calculated and the pipeline is drawn from this point.

PARAMETERS:

UPDATE	see 4.6
UPROW	The row in array UPDATE containing the parameters defining the item.
IXX,IYY	The co-ordinates of the light beam; used when the nozzle position is being redefined.
XO,YO	The co-ordinates of the end of the nozzle at which point the pipeline begins.
BUP	Brightness parameter.
BUF	see ARROW
BASE	see BOXIN

SUBROUTINE PIPEX (IPIPE,NOZZLE,BUF,BASE,XBASE,YBASE)

FUNCTION:

To replace an existing orthogonal pipeline with direct nozzle to nozzle piping.

PARAMETERS:

IPIPE,NOZZLE	see 4.6
BUF	see ARROW
BASE,XBASE, YBASE	see BOXIN

SUBROUTINE PIRACK (YTRAK,IWIDTH,IRATIO,BUP,BUF,BASE, XBASE,YBASE)

FUNCTION:

To display a representation of the main piperack.

PARAMETERS:

YTRAK	The Y co-ordinate (absolute) of the upper edge of the piperack.
IWIDTH	The piperack width.
IRATIO	The number of screen increments representing one foot.
BUP	Brightness parameter.
BUF	see ARROW
BASE,XBASE, YBASE	see BOXIN

SUBROUTINE PRINT1 (IPIPE)

FUNCTION:

This subroutine writes out the contents of array IPIPE.

PARAMETERS:

IPIPE see 4.6

SUBROUTINE PRINT2 (NOZZLE)

FUNCTION:

This subroutine writes out the contents of array NOZZLE.

PARAMETERS:

NOZZLE see 4.6

SUBROUTINE PRINT3 (UPDATE)

FUNCTION:

To write out array UPDATE

PARAMETERS:

UPDATE see 4.6

SUBROUTINE PRINT4 (DIMENS,ICODE,ITNUMB)

FUNCTION:

This routine writes out the contents of array DIMENS, indicating which item the dimensions refer to.

PARAMETERS:

DIMENS,ICODE	see 4.6
ITNUMB	The number of plant items being displayed.

SUBROUTINE PRINT5 (CLEAR,ICODE,ITNUMB)

FUNCTION:

This writes out array CLEAR, indicating which item the clearances refer to.

PARAMETERS:

CLEAR,ICODE	see 4.6
ITNUMB	The number of plant items being displayed.

SUBROUTINE PRINTX (IPIPE,UPDATE,NOZZLE,DIMENS,ICODE, ITNUMB,CLEAR)

FUNCTION:

This routine calls the above PRINT routines.

PARAMETERS:

see above.

SUBROUTINE PTOV (UPDATE IPIPE,K,IL,N,KIX,KIY,NUP,BUF, BASE)

FUNCTION:

This routine displays pipework of type 4.5.3

PARAMETERS:

UPDATE,IPIPE	see 4.6
K	The row in array UPDATE containing the parameters defining the item.

IL	The row in array IPIPE containing the parameters defining the pipe.
N	The row in array IPIPE in which will be entered the parameters defining the pipeline to be displayed.
KIX,KIY	The co-ordinates of the nozzle position on the item perimeter.
BUP	A brightness parameter.
BUF	see ARROW
BASE	see BOXIN

<u>SUBROUTINE PTOVX (UPDATE,IPIPE,NOZZLE,N,BUF,NASE, XBASE,YBASE,YTRAK)</u>

<u>FUNCTION</u>:

This subroutine allows the user to redefine pipework of type 4.5.3, by calling PTOV for different NOZZLE co-ordinates.

<u>PARAMETERS</u>:

UPDATE,IPIPE, NOZZLE	see 4.6
N	The row in array IPIPE containing the parameters defining the pipeline to be redisplayed.
BUF	see ARROW
BASE,XBASE, YBASE	see BOXIN
YTRAK	The Y co-ordinate of the upper edge of the piperack.

<u>SUBROUTINE REMOVE (UPDATE,IPIPE,NOZZLE,XBASE,YBASE, BASE,BUF,YTRAK)</u>

FUNCTION:

This routine allows the operator to move a plant item around the screen. Any pipework associated with the item will be redisplayed at the new item position. The pipework costs are recalculated thus allowing the user to position the item in the optimum position.

PARAMETERS:

UPDATE,IPIPE, NOZZLE	see 4.6
XBASE,YBASE BASE	see ARROW
BUF	see BOXIN
YTRAK	The y co-ordinate of the upper edge of the piperack.

SUBROUTINE REPUT (UPDATE,IPIPE,NOZZLE,BASE,BUF,XBASE, YBASE)

FUNCTION:

This routine reads in the paper tape output by INTAPE. This allows the user to work on a number of layouts modifying each in turn.

PARAMETERS:

see PTOVX

SUBROUTINE SETUP (YTRAK,IWIDTH,IRATIO,BUF,BASE,XBASE, YBASE)

FUNCTION:

This routine sets up the base co-ordinates for the scissoring routine.

PARAMETERS:

YTRAK	The Y co-ordinate of the upper edge of the piperack.
IWIDTH	The piperack width.
IRATIO	The number of screen increments representing one foot.
BUF	see ARROW
BASE, XBASE, YBASE	see BOXIN

SUBROUTINE SWOPP (UPDATE,I,BUF)

FUNCTION:

To alter the orientation of an item, swinging it through 180 degrees.

PARAMETERS:

UPDATE	see 4.6
I	The row in array UPDATE containing the parameters defining the item.
BUF	see ARROW

SUBROUTINE VIEW (XBASE,YBASE,BASE,BUF)

FUNCTION:

This routine allows the user to scan the picture area available for display.

SUBROUTINE VTOP (UPDATE,IPIPE,I,M,N,JIX,JIY,BUP,BUF BASE)

FUNCTION:

This routine displays pipework of type 4.5.2

PARAMETERS:

UPDATE,IPIPE	see 4.6
I	The row in array UPDATE containing the parameters defining the item.
M	The row in array IPIPE containing the parameters defining the pipe.
N	The row in array IPIPE in which will be entered the parameters defining the pipeline to be displayed.
JIX,JIY	The co-ordinates of the nozzle position on the item parimeter.
BUP	Brightness parameter.
BUF	see ARROW
BASE	see BOXIN

VTOPX (UPDATE,IPIPE,NOZZLE,N,BUF,BASE,XBASE,YBASE,YTPAK)

FUNCTION:

This subroutine allows the user to redefine pipework of type 4.3.2. by calling VTOP for different nozzle co-ordinates.

PARAMETERS:

UPDATE,IPIPE, NOZZLE	see 4.6
N	The row in array IPIPE containing the parameters defining the pipeline to be redisplayed.

BUF	see ARROW
BASE,XBASE, YBASE	see BOXIN
YTRAK	The Y co-ordinate of the upper edge of the piperack.

<u>VTOV (UPDATE,IPIPE,I,K,N,JIX,JIY,KIX,KIY,NUP,BUF,BASE, XBASE,YBASE,YTRAK)</u>

<u>FUNCTION</u>:

This routine displays pipework of type 4.5.1.

<u>PARAMETERS</u>:

UPDATE,IPIPE	see 4.6
I,K	The rows in array UPDATE containing the defining items.
KIX,JIY	The co-ordinates of the nozzle position on one item perimeter.
KIX,KIY	The co-ordinates of the nozzle position on one item perimeter.
BUP	Brightness parameter.
BUF	see ARROW
BASE,XBASE, YBASE	see BOXIN
YTRAK	The Y co-ordinates of the upper edge of the piperack.

VTOVX (UPDATE,IPIPE,NOZZLE,N,BUF,BASE,XBASE,YBASE,YTRAK)

FUNCTION:

This subroutine allows the user to redefine pipework of type 4.3.1 by calling VTOV for different nozzle co ordinates.

PARAMETERS:

UPDATE,IPIPE, NOZZLE	see 4.6
N	The row in array IPIPE containing the parameters defining the pipeline to be redisplayed.
BUF	see ARROW
BASE,XBASE, YBASE	see BOXIN
YTRAK	The Y co-ordinates of the upper edge of the piperack.

SUBROUTINE WAIT (KEY,BUF,NUMBER)

FUNCTION:

The routine halts the program until a key is depressed. An action then occurs depending on which key is pressed.

PARAMETERS:

KEY	This integer takes the value of the key depressed.
BUF	see ARROW
NUMBER	This integer is written on the screen so that the user knows with which set of decisions he is concerned.

SUBROUTINE ZERO (IPIPE,UPDATE)

FUNCTION:

This routine initialises the IPIPE and UPDATE arrays. Each element of the arrays is set to zero except for the fifteenth column of array UPDATE which is set to unity.

PARAMETERS:

IPIPE,UPDATE	see 4.6

SUBROUTINE ZONAL (UPDATE,DIMENS,ICODE,XBASE,YBASE,BASE, BUF,ITYPES,ITNUMB,CLEAR,SCALE,UPROW)

FUNCTION:

This routine handles the operation whereby the user chooses an item to be displayed. The screen size of the unit is calculated and the parameters are entered in array UPDATE.

PARAMETERS:

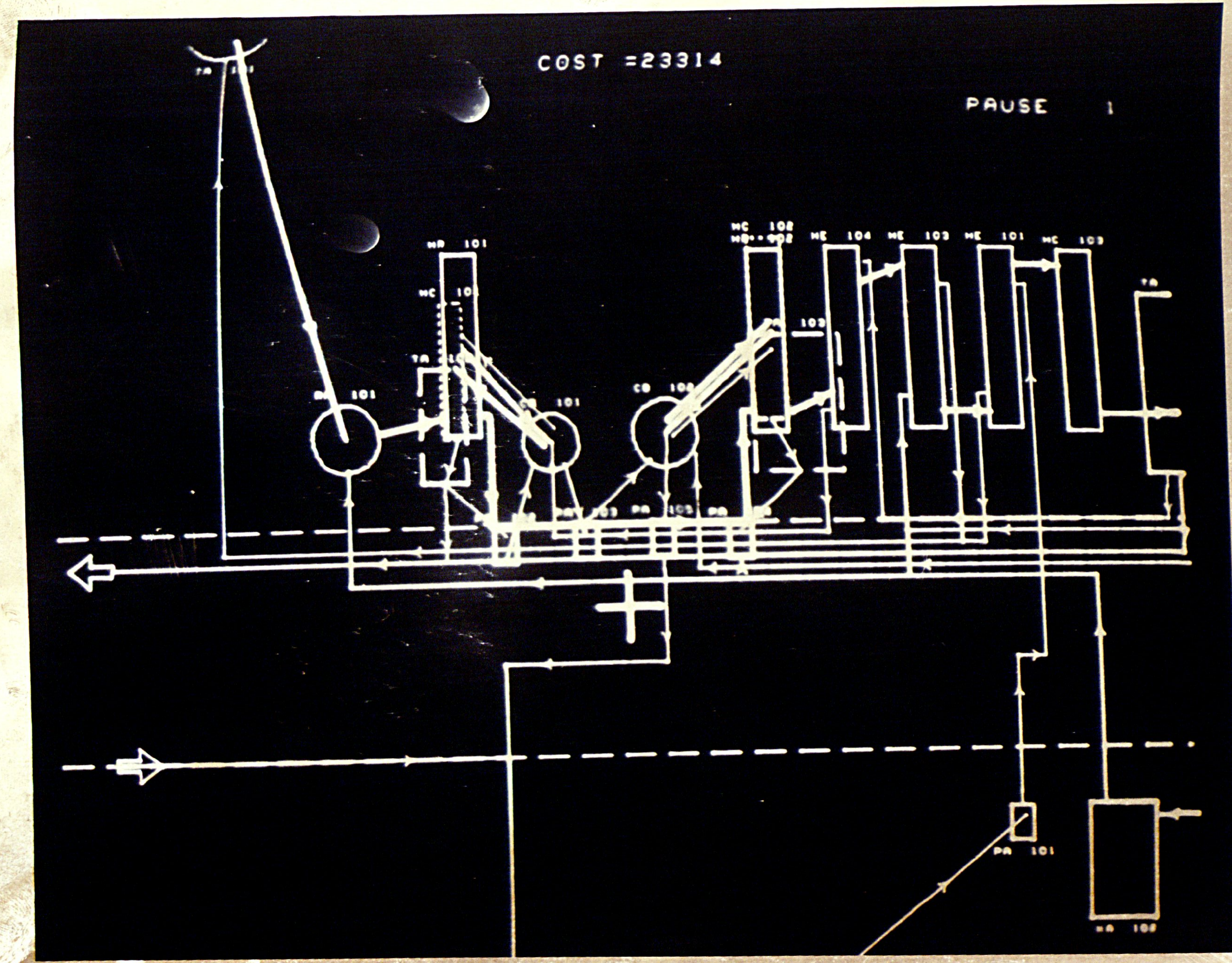
UPDATE,DIMENS, ICODE	see 4.6
XBASE,YBASE, BASE	see BOXIN
BUF	see ARROW
ITYPES	The number of item types.
ITNUMB	The total number of items.
CLEAR	see 4.6
SCALE	An integer scale factor to convert the dimension in array DIMENS to feet.
UPROW	The row in array UPDATE where the parameters defining the item chosen are to be located.

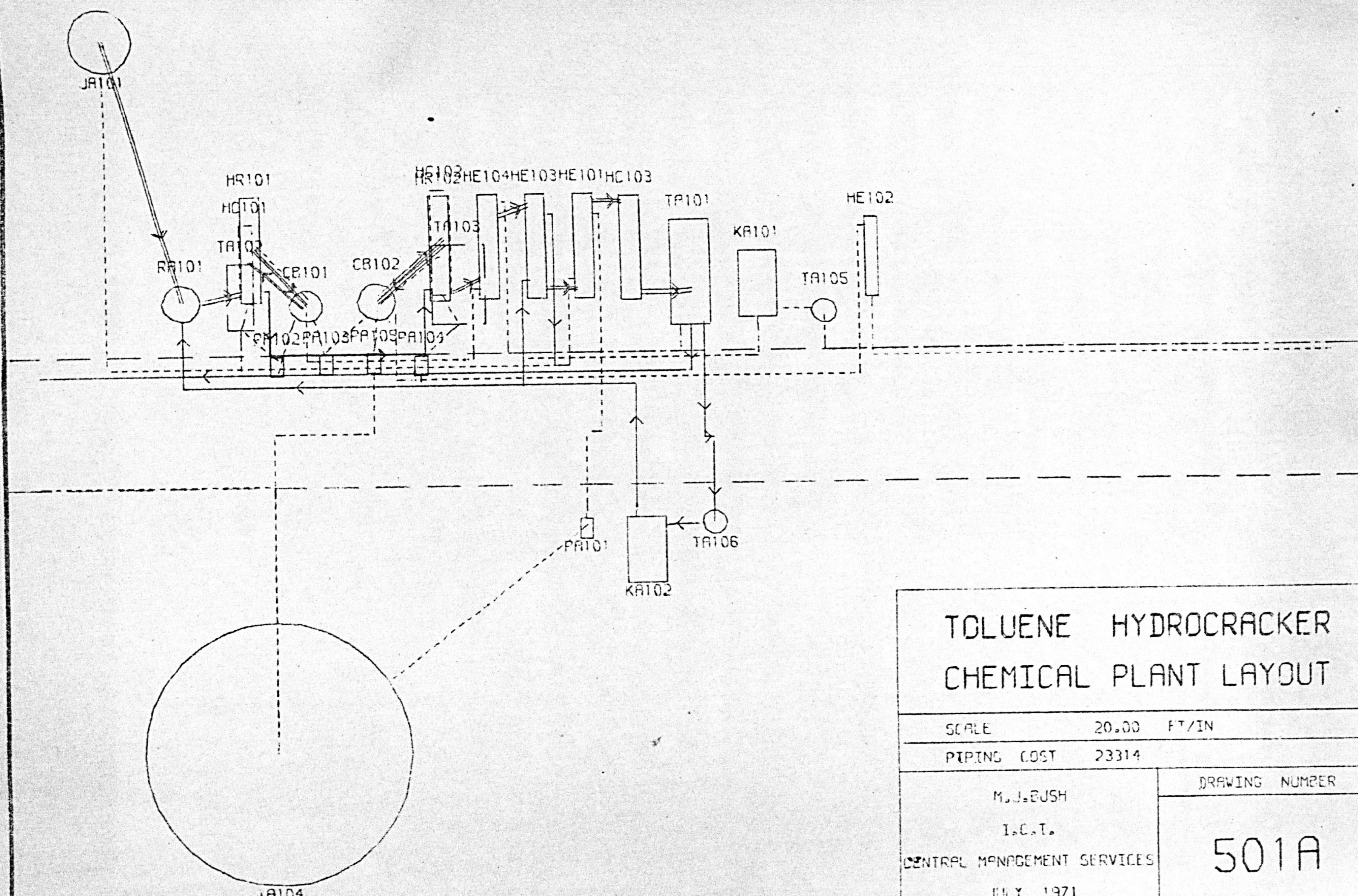
4.8 Toluene Hydrocracker Layout

The petrochemical plant laid out manually in section 3.4, the process flow diagram of which is shown in Fig. 2., was laid out using the computer graphics program developed. The item code specification was altered in accordance with section 4.4 and this specification occurrs on the plot of the display shown in Fig. 8. Table 11, indicates the specification changes. The Elliott display allows three levels of brightness and this enabled three pipe cost groupings to be presented. The plot (Fig. 9) of the display shows a double line for group one pipelines (most expensive); a single line for group two pipelines; and a dashed line for group three pipelines.

Old Specification as Fig. 2.	New Specification as Fig. 8.
D 102	CB 101
D 103	CB 102
E 107	HE 101
E 109	HE 102
E 108	HE 103
E 106	HE 104
E 103	HR 101
E 105	HR 102
E 102	HC 101
E 104	HC 102
E 101	HC 103
B 101	JA 101
C 102	KA 101
C 101	KA 102
G 101	PA 101
G 103	PA 102
G 104	PA 103
G 105	PA 104
G 106	PA 105
D 101	RA 101
F 101	TA 101
F 102	TA 102
F 103	TA 103
F 104	TA 104
F 105	TA 105
F 106	TA 106

TABLE 11


COST =23314
PAUSE 1
MA 101
CB 103
ME 104
ME 103
ME 101
MC 103
PA 101
PA 103



TOLUENE HYDROCRACKER
CHEMICAL PLANT LAYOUT

SCALE	20.00	FT/IN
PIPING COST	23314	

M.J.BUSH I.C.I. CENTRAL MANAGEMENT SERVICES JULY 1971	DRAWING NUMBER 501A

Conclusion

Some indication has already been given of the scope of the graphics development program envisaged by I.C.I. The philosophy of their approach is, I believe, still valid although work on this project is at present curtailed.

It was estimated that it would take about six years to develope and implement the fully integrated system. Obviously in nine months only a limited amount of work was possible. The graphics program developed is constricted insofar as it gives a two dimensional projection only. Scissoring capability also leaves much to be desired. Extension of the Program to enable three dimensional layout with isometric projection and full scissoring facilities are logical improvements and would greatly increase the potential of the program. The size of the plant that the program will handle is limited only by the size of the computer. The program developed handles twenty six plant process items on a 32k machine.

APPENDIX I

Plant Layout Program

This program is written in Fortran IV for an Elliott 41/30 machine.

```
C
C          INDEX OF SUBROUTINES

C   SUBROUTINE          PAGE

C     ARROW              11
C     BOX                70
C     BOXIN              35
C     BOXOUT             34
C     DISBOX             67
C     DISCUB             48
C     DISCYL             59
C     DRAW               62
C     ELVATE             52
C     ERASE              17
C     FIGURE             14
C     FIRST              15
C     FOX                69
C     INCOST             18
C     INITAL             24
C     INPIPE              7
C     INSET              22
C     INTAPE             47
C     INVERT             37
C     ITRASE             66
C     LETTER             20
C     LEVEL              60
C     LVEC               68
C     NOPIPE             19
C     OFSITE             12
C     ORDER              58
C     PIMOVE             56
C     PIPE1              64
C     PIPEX              38
C     PIRACK             21
C     PRINT1             42
C     PRINT2             41
C     PRINT3             43
C     PRINT4             44
C     PRINT5             45
C     PRINTX             46
C     PTOV               27
C     PTOVX              32
C     REMOVE             54
C     REPUT              49
C     SCREEN              3
C     SETUP              36
C     SWOPP              63
C     VIEW               10
C     VTOP               26
C     VTOPX              30
C     VTOV               25
C     VTOVX              28
C     WAIT               40
C     ZERO               53
C     ZONAL              51
```

```
C
      FETCH AROVEC
      SEGMENT
      FETCH ARC
      SEGMENT
      FETCH LINE
      SEGMENT
```

```
C
      SUBROUTINE SCREEN(DIMENS,ICODE,CLEAR,
     *BUF,NOZZLE,ITNUMB,ITYPES,BASE)
      INTEGER DIMENS(ITNUMB,5),CLEAR(ITNUMB,6),UPDATE(30,15),
     *ACTION,BUF(200),ICODE(ITYPES,2),FOUND,
     *TYPE,CODE,YPOINT,NOZZLE(60,4),UPROW,IPIPE(60,11),
     *SCALE,BASE(8),XBASE,YBASE,
     *YTRAK,BUP
      LOGICAL SWITCH

C  INITIALISE IPIPE AND UPDATE ARRAYS

      CALL ZERO(IPIPE,UPDATE)
      IWIDTH=20
      XBASE=-1023
      YBASE=-1023
      YTRAK=100-YBASE
      IRATIO=10
      LSCALE=1
      SCALE=192
      JSHINE=1
      KMODE=1
      ITYPE=1

C  INSERT COST STATEMENT AND PIPERACK REPRESENTATION

      CALL SETUP(YTRAK,IWIDTH,IRATIO,BUF,BASE,XBASE,YBASE)
      GO TO 106
    3 CALL ZONAL(UPDATE,DIMENS,ICODE,XBASE,YBASE,BASE,BUF,ITYPES,
     *ITNUMB,CLEAR,SCALE,UPROW)
      CALL PIRACK(YTRAK,IWIDTH,IRATIO,BUP,BUF,BASE,XBASE,
     *YBASE)
      MODE=1
      KMODE=1
      CALL CRPOS(850,850)

C  THE ITEM AND ACCESS AREA ARE GENERATED AND POSITIONED ON THE SCREEN
C  USING THE LIGHT-PEN.

 8999 CALL TRAK(IX,IY)
      GO TO(31,41),KMODE

C  IF THE ITEM IS PULLED BELOW THE PIPERACK IT IS INVERTED BY CALLING
C  SWOPP.

   31 IF(IY-YBASE.LT.1023)GO TO 32
      GO TO 9000
   32 KMODE=2
      CALL SWOPP(UPDATE,UPROW,BUF)
      GO TO 9000
   41 IF(IY-YBASE.GT.1023)GO TO 42
      GO TO 9000
   42 KMODE=1
      CALL SWOPP(UPDATE,UPROW,BUF)

C  GENERATE THE ITEM CODE

 9000 CALL FIGURE(UPDATE,UPROW,IX,IY,XBASE,YBASE,
     *BASE,BUF)
```

```
C
      CALL DISBOX(UPDATE(UPROW,8),UPDATE(UPROW,9),UPDATE(UPROW,10),
     *UPDATE(UPROW,11),JSHINE)

C  THE ITEM IS INSERTED ON THE FIRST PASS WHEN MODE = 1. SUBSEQUENTLY
C  MODE IS SET TO 2, AND THE ITEM IS CONTINUOUSLY REPLACED.

      GO TO(9001,9002),MODE
 9001 CALL INSERT(UPDATE(UPROW,3),0)
      CALL BOXIN(UPDATE,XBASE,YBASE,BASE,BUF)
      MODE=2
      GO TO 8999
 9002 CALL REPL(UPDATE(UPROW,3))

C  THE CYCLE IS LEFT WHEN KEY 1 IS DEPRESSED.

      IF(SWITCH(1))GO TO 105
      GO TO 8999

C  THE ABSOLUTE CO-ORDINATES OF THE ITEM ARE ENTERED IN THE UPDATE
C  ARRAY. THE ACCESS AREA IS DELETED FROM THE SCREEN.

  105 UPDATE(UPROW,4)=IX-XBASE
      UPDATE(UPROW,5)=IY-YBASE
      CALL BOXOUT(UPDATE,XBASE,YBASE,BASE,BUF)

C  SECTION ONE:- THE USER MAY DEFINE AN ACTION BY DEPRESSING A KEY
C  AS LISTED BELOW.

  106 CALL WAIT(KEY,BUF,1)

C
C         1 - DISPLAY LIST OF ITEM TYPES
C         2 - DELETE AN ITEM FROM THE SCREEN
C         3 - MOVE AN ITEM
C         4 - DELETE ALL PIPING
C         5 - WRITE ARRAY NOZZLE
C         6 - INVERT AN ITEM
C         7 - INPUT ITEMS BY PAPER TAPE
C         8 - SWITCH TO SECTION TWO
C
      GO TO (3,200,300,198,197,4001,199,1200),KEY

C  WRITE THE CONTENTS OF ARRAY NOZZLE.

  197 CALL NZPRNT(NOZZLE)
      GO TO 106

C  DELETE ALL PIPING AND SET PIPING COST TO ZERO.

  198 CALL NOPIPE(IPIPE)
      CALL INCOST(IPIPE,NOZZLE,BUF)
      GO TO 106

C  INPUT ON PAPER TAPE A PREDEFINED LAYOUT.

  199 CALL REPUT(UPDATE,IPIPE,NOZZLE,BASE,BUF,XBASE,YBASE)
      GO TO 106
```

```
C
C  DELETE AN ITEM FROM THE SCREEN.
  200 CALL ERASF(IPIPF,UPDATE,NOZZLF,BUF)
      GO TO 106

C  AN ITEM IS SPECIFIED WITH THE LIGHT-PEN. ITS POSITION IN ARRAY
C  UPDATE IS DETERMINED. THE REGENERATION CYCLE IS ENTERED SO THAT
C  THE ITEM MAY BE REPOSITIONED ON THE SCREEN. THE ACCESS AREA IS
C  REDISPLAYED.

  300 JAB=ACTION(4,FOUND)
      DO 301 IS=1,30
      IF(UPDATE(IS,3).EQ.JAB)GO TO 302
  301 CONTINUE
  302 IX=UPDATE(IS,4)+XBASE
      IY=UPDATE(IS,5)+YBASE
      KMODE=1
      IF(IY-YBASE.LT.1023)KMODE=2
      CALL CRPOS(IX,IY)
      CALL BOXIN(UPDATE,XBASE,YBASE,BASE,BUF)
      UPROW=IS
      MODE=2
      GO TO 8999

C  SECTION TWO:- THE USER MAY DEFINE AN ACTION BY DEPRESSING ONE OF
C  THE SENSE KEYS.

 1200 CALL WAIT(KEY,BUF,2)

C        KEY 1 - PRINT ARRAYS
C        KEY 2 - MOVE LEFT
C        KEY 3 - MOVE RIGHT
C        KEY 4 - MOVE UP
C        KEY 5 - MOVE DOWN
C        KEY 6 - INDICATE ELEVATED ITEM
C        KEY 7 - OUTPUT PUNCH
C        KEY 8 - SWITCH TO SECTION THREE

      GO TO(2999,3001,3001,3001,3001,2998,3000,4000),KEY

C  AN ITEM IS SPECIFIED WITH THE LIGHT-PEN. IT CAN BE SHOWN WITH
C  DOTTED OR DASHED LINES TO INDICATE THAT THE ITEM IS LOCATED
C  ABOVE GROUND LEVEL.

 2998 CALL ITRASE(UPDATE,XBASE,YBASE,BASE,BUF)
      GO TO 1200

C  ARRAYS UPDATE, IPIPE, NOZZLE, DIMENS, AND CLEAR ARE WRITTEN OUT.

 2999 CALL PRINTX(IPIPE,UPDATE,NOZZLE,DIMENS,ICODE,ITNUMB,CLEAR)
      GO TO 1200

C  ARRAYS UPDATE AND IPIPE ARE OUTPUT TO PAPER TAPE. THESE ARRAYS
C  DEFINE THE LAYOUT.

 3000 CALL INTAPE(UPDATE,IPIPE)
      GO TO 1200
```

```
C
C  SCISSORING ROUTINE.

 3001 CALL VIEW(XBASE,YBASE,BASE,BUF)
      GO TO 1200

C  BASIC SUBROUTINE HANDLING ALL PIPING FACILITIES.

 4000 CALL INPIPE(NOZZLE,UPDATE,IPIPE,YLEVEL,YTRAK,BUF,BASE,BUP,
     *XBASE,YBASE)
      GO TO 106

C  AN ITEM IS SPECIFIED WITH THE LIGHT-PEN. IT IS INVERTED AND THE
C  ACCESS AREA IS REDISPLAYED. THE REGENERATION CYCLE IS ENTERED SO
C  THAT THE ITEM MAY BE REPOSITIONED.

 4001 CALL INVERT(UPDATE,UPROW,XBASE,YBASE,BUF)
      CALL BOXIN(UPDATE,XBASE,YBASE,BASE,BUF)
      MODE=2
      GO TO 8999
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE INPIPE(NOZZLE,UPDATE,IPIPE,YLEVEL,YTRAK,
     *BUF,BASE,RUP,XBASE,YBASE)
      INTEGER NOZZLE(60,4),UPDATE(30,15),IPIPE(60,11),BUF(200),
     *BASE(8),BUP,YLEVEL,YTRAK,
     *ACTION,XBASE,YBASE
      LOGICAL SWITCH

C  SECTION THREE:- THE USER MAY DEFINE AN ACTION BY DEPRESSING
C  ONE OF THE SENSE KEYS.

 5000 CALL WAIT(KEY,BUF,3)

C       KEY 1 - DISPLAY PIPING
C       KEY 2 - NOZZLE REDEFINITION
C       KEY 3 - DELETE PIPING
C       KEY 4 - CHANGE PIPELINE LEVEL
C       KEY 5 - MOVE AN ITEM
C       KEY 6 - DELETE AN OBJECT
C       KEY 7 - DIRECT PIPING
C       KEY 8 - SWITCH TO SECTION ONE

      GO TO(500,5010,2001,1999,1998,1000,2000,106),KEY

C  AN ITEM IS SPECIFIED WITH THE LIGHT-PEN AND MAY BE REPOSITIONED
C  THE PIPELINES ARE REROUTED AND THE NEW PIPING COST IS COMPUTED.

 1998 CALL REMOVE(UPDATE,IPIPE,NOZZLE,XBASE,YBASE,BASE,
     *BUF,YTRAK)
      GO TO 5000

C  THIS ROUTINE ALLOWS THE USER TO REPOSITION THE PIPELINE ON
C  THE PIPERACK.

 1999 CALL ELVATE(IPIPE,NOZZLE,BASE,BUF,XBASE,YBASE)
      GO TO 5000

C A PIPELINE IS DRAWN DIRECTLY BETWEEN THE NOZZLES.

 2000 CALL PIPEX(IPIPE,NOZZLE,BUF,BASE,XBASE,YBASE)
      GO TO 5000

C  ALL PIPING IS DELETED FROM THE SCREEN.

 2001 CALL NOPIPE(IPIPE)
      CALL INCOST(IPIPE,NOZZLE,BUF)
      GO TO 5000

C  A PIPELINE IS SPECIFIED WITH THE LIGHT-PEN AND ITS POSITION
C  IN ARRAY IPIPE IS IDENTIFIED.

 5010 KPIPE=ACTION(4,FOUND)
      DO 5011 N=1,60
      IF(KPIPE.EQ.IPIPE(N,1))GO TO 5014
 5011 CONTINUE
      GO TO 5000
```

```
C
C  THE TRACKING CROSS IS PLACED AT THE BEGINNING OF THE
C  PIPELINE. IZ IS THE PIPELINE TYPE.

 5014 CALL CRPOS(IPIPE(N,2)+XBASE,IPIPE(N,3)+YBASE)
      IZ=IPIPE(N,11)
      GO TO(5020,5030,5040,2000,5000),IZ

C  VESSEL TO VESSEL, PIPEWORK REROUTING. IZ=1.

 5020 CALL VTOVX(UPDATE,IPIPE,NOZZLE,N,BUF,
     *BASE,XBASE,YBASE,YTRAK)
      GO TO 5000

C  VESSEL TO PIPE, PIPEWORK REROUTING. IZ=2.

 5030 CALL VTOPX(UPDATE,IPIPE,NOZZLE,N,BUF,
     *BASE,XBASE,YBASE,YTRAK)
      GO TO 5000

C  PIPE TO VESSEL, PIPEWORK REROUTING. IZ=3.

 5040 CALL PTOVX(UPDATE,IPIPE,NOZZLE,N,BUF,
     *BASE,XBASE,YBASE,YTRAK)
      GO TO 5000

C  PIPING IS DISPLAYED.

  500 J=0
  501 CALL FIRST(NOZZLE,UPDATE,IPIPE,I,J,K,IL,M,N,
     *XBASE,YBASE,BASE,BUF,YTRAK)
      IF(J.GT.60)GO TO 5000
      IPIPE(N,1)=J+1000
      MODE=1

C  THE CO-ORDINATES OF THE ORIGINS OF THE ITEMS ARE FOUND.
C  BUP IS A BRIGHTNESS PARAMETER.

      JIX=UPDATE(I,4)
      JIY=UPDATE(I,5)
      KIX=UPDATE(K,4)
      KIY=UPDATE(K,5)
      BUP=NOZZLE(J,3)

C  SET UP EMPTY CODE BUFFER AND SET BEAM POSITION TO DISPLAY
C  AREA ORIGIN CO-ORDINATES.

      CALL NEWBUF(BUF,200)
      CALL ENTER(BASE,8)

C  DETERMINE THE TYPE OF PIPELINE BY EXAMINATION OF ARRAY
C  NOZZLE.

      IF(NOZZLE(J,1).LT.1000.AND.NOZZLE(J,2).LT.1000)GO TO 630
      IF(NOZZLE(J,1).LT.1000.AND.NOZZLE(J,2).GT.1000)GO TO 640
      IF(NOZZLE(J,1).GT.1000.AND.NOZZLE(J,2).LT.1000)GO TO 650
```

```
C
C  VESSEL TO VESSEL, PIPEWORK DISPLAY.

  630 CALL VTOV(UPDATE,IPIPE,I,K,N,JIX,JIY,KIX,KIY,
     *BUP,BUF,BASE,XBASE,YBASE,YTRAK)
  631 CALL INSERT(J+1000,0)
      CALL INCOST(IPIPE,NOZZLE,BUF)
      GO TO 501

C  VESSEL TO PIPE, PIPEWORK DISPLAY.

  640 CALL VTOP(UPDATE,IPIPE,I,M,N,JIX,JIY,BUP,BUF,BASE)
  646 CALL INSERT(J+1000,0)
      CALL INCOST(IPIPE,NOZZLE,BUF)
      GO TO 501

C  PIPE TO VESSEL, PIPEWORK DISPLAY.

  650 CALL PTOV(UPDATE,IPIPE,K,IL,N,KIX,KIY,BUP,BUF,BASE)
  656 CALL INSERT(J+1000,0)
      CALL INCOST(IPIPE,NOZZLE,BUF)
      GO TO 501

C  DELETE AN OBJECT FROM THE SCREEN.

 1000 CALL ERASE(IPIPE,UPDATE,NOZZLE,BUF)
      GO TO 5000
  106 RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE VIEW(XBASE,YBASE,BASE,BUF)
      INTEGER VALUE,XBASE,YBASE,BASE(8),BUF(200),KEYS
      LOGICAL SWITCH

C   THIS SUBROUTINE SCISSORS THE DISPLAY. THE AREA AVAILABLE
C   FOR DISPLAY IS 2046 X 2046 SCREEN INCREMENTS. THE SCREEN
C   SIZE IS 1023 X 1023 INCREMENTS.
C   KEYS IS AN INTEGER FUNCTION WHICH BECOMES THE BINARY SETTING
C   OF KEYS 2-5 INCLUSIVE.

    1 VALUE=KEYS(2,5)
      IF(VALUE.EQ.0)GO TO 100
      IF(VALUE.GT.8.OR.VALUE.LT.1)GO TO 1
    5 GO TO (10,11,1,12,1,1,1,13),VALUE

C   MOVE PICTURE TO RIGHT.

   10 IF(XBASE.GT.-1023)GO TO 100
      IF(.NOT.SWITCH(2))GO TO 100
      XBASE=XBASE+1
      GO TO 50

C   MOVE PICTURE TO LEFT.

   11 IF(XBASE.LT.-2046)GO TO 100
      IF(.NOT.SWITCH(3))GO TO 100
      XBASE=XBASE-1
      GO TO 50

C   MOVE PICTURE UP.

   12 IF(YBASE.GT.0)GO TO 100
      IF(.NOT.SWITCH(4))GO TO 100
      YBASE=YBASE+1
      GO TO 50

C   MOVE PICTURE DOWN.

   13 IF(YBASE.LT.-1023)GO TO 100
      IF(.NOT.SWITCH(5))GO TO 100
      YBASE=YBASE-1

C   THE DISPLAY ORIGIN CO-ORDINATES, XBASE, YBASE, ARE
C   INCREMENTED THUS SHIFTING THE SCREEN 'WINDOW' ACROSS THE
C   DISPLAY AREA.

   50 CALL NEWBUF(BASE,8)
      CALL POINT(0,0,0)
      CALL VECTOR(XBASE/2,YBASE/2,0)
      CALL VECTOR(XBASE/2,YBASE/2,0)
      GO TO 1
  100 RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINF ARROW(RUP,RUF,K)
      INTEGER BUP,BUF(200)
C  GENERATE CODF TO FORM ARROW.
      CALL LINE(0,0,0,K,BUP,1)
      CALL LINE(0,0,4*K,0,BUP,1)
      CALL LINE(0,0,0,2*K,BUP,1)
      CALL LINE(0,0,4*K,-3*K,RUP,1)
      CALL LINE(0,0,-4*K,-3*K,BUP,1)
      CALL LINE(0,0,0,2*K,BUP,1)
      CALL LINE(0,0,-4*K,0,RUP,1)
      CALL LINE(0,0,0,K,BUP,1)
      RETURN
      END
      SEGMENT
```

```
      SUBROUTINE OFSITE(NOZZLE,UPDATE,IPIPE,J,YTRAK,XBASE,YBASE,
     *BASE,BUF)
      INTEGER NOZZLE(60,4),UPDATE(30,15),IPIPE(60,11),XBASE,
     *YBASE,BASE(8),BUF(200),YLEVEL,YTRAK,YMIN,BUP
      YMIN=YTRAK-200
      M=5
      K=1
      MODE=1
      IF(J.GT.1000)MODE=2
      IF(J.GT.1000)J=J-1000
C
C  IF MODE = 1, THE PIPE IS INSERTED.
C  IF MODE = 2, THE PIPE IS REPLACED.
C
      DO 5 JA=1,60
      IF(IPIPE(JA,1).EQ.J+1000)GO TO(50,60),MODE
    5 CONTINUE
C
C  SET BRIGHTNESS PARAMETER BUP, AND DETERMINE IF THE PIPE
C  CARRIES A FEED OR PRODUCT STREAM.
C
    6 BUP=NOZZLE(J,3)
      IF(NOZZLE(J,1).EQ.1)K=2
C
C  SET UP CODE BUFFER.
C
      CALL NEWBUF(BUF,200)
      CALL ENTER(BASE,8)
C
C  DETERMINE PIPE TYPE.
C
      DO 20 L=1,60
      IF(NOZZLE(J,K).EQ.UPDATE(L,3))GO TO 25
      IF(NOZZLE(J,K).EQ.IPIPE(L,1))GO TO 50
   20 CONTINUE
      RETURN
C
C  CALCULATE WHETHER UNIT IS CLOSER TO THE RIGHT OR LEFT SITE
C  BOUNDARY AND WHETHER THE UNIT IS ABOVE OR BELOW THE
C  PIPERACK.
C
   25 IX=UPDATE(L,4)
      IY=UPDATE(L,5)
      JY=YTRAK
      JX=3019
      IF(IX.LT.2046)JX=1073
      IF(IY.LT.YTRAK)JY=YMIN
C
C  SET ARROW SCALE FACTOR M AND FIND A LEVEL AVAILABLE ON
C  THE PIPERACK.
C
      IF(K.EQ.2.AND.IX.GE.2046.OR.K.EQ.1.AND.
     *IX.LT.2046)M=-5
      CALL LEVEL(YLEVEL,YTRAK,IPIPE,IX,IY,JX,JY,BUF)
```

```
C
C   GENERATE CODE FOR PIPELINE DISPLAY.
      GO TO(30,35),K
   30 CALL LVEC(IX,IY)
      CALL AROVEC(0,IY,0,YLEVEL,BUP,1)
      CALL AROVEC(IX,0,JX,0,BUP,1)
      CALL ARROW(BUP,BUF,M)
      GO TO(40,70),MODE
   35 CALL LVEC(JX,YLEVEL)
      CALL ARROW(BUP,BUF,M)
      CALL AROVEC(JX,0,IX,0,BUP,1)
      CALL AROVEC(0,YLEVEL,0,IY,BUP,1)
      GO TO(40,70),MODE
C   INSERT THE ITEM. FIND FIRST VACANT ROW IN ARRAY IPIPE AND
C   INSERT THE PARAMETERS DEFINING THE LINE.
   40 CALL INSERT(1000+J,0)
      DO 1 IS=1,100
      IF(IPIPE(IS,1).EQ.0)GO TO 2
    1 CONTINUE
    2 IPIPE(IS,1)=J+1000
      IPIPE(IS,2)=K
      IPIPE(IS,3)=M
      IPIPE(IS,4)=IX
      IPIPE(IS,5)=IY
      IPIPE(IS,6)=YLEVEL
      IPIPE(IS,7)=JX
      IPIPE(IS,11)=5
   50 RETURN
   60 IPIPE(JA,6)=0
      GO TO 6
   70 CALL REPL(J+1000)
      IS=JA
      GO TO 2
      END
      SEGMENT
```

```
C
      SUBROUTINE FIGURE(UPDATE,UPROW,IX,IY,XBASE,YBASE,
     *BASE,BUF)
      INTEGER UPDATE(30,15),UPROW,XBASE,YBASE,BASE(8),
     *BUF(200)
C  ITYPE INDICATES IF THE ITEM IS ELEVATED.
      ITYPE=UPDATE(UPROW,15)
C  SET UP CODE BUFFER.
      CALL NEWBUF(BUF,200)
      CALL ENTER(BASE,8)
      CALL LVEC(IX-XBASE,IY-YBASE)
   41 IF(UPDATE(UPROW,13))45,50,55
C  CUBE.
   45 CALL DISCUB(IX,IY,UPDATE(UPROW,6),UPDATE(UPROW,7),BUF,ITYPE)
      CALL VECTOR(0,UPDATE(UPROW,7),0)
      GO TO 60
C  SPHERE.
   50 CALL VECTOR(-UPDATE(UPROW,12)/2,0,0)
      CALL ARC(-UPDATE(UPROW,12)/2,0,-UPDATE(UPROW,12)/2,0,0)
      CALL VECTOR(UPDATE(UPROW,12)/2,0,0)
      CALL VECTOR(0,UPDATE(UPROW,12)/2,0)
      GO TO 60
C  CYLINDER.
   55 CALL DISCYL(IX,IY,UPDATE(UPROW,7),UPDATE(UPROW,12),
     *UPDATE(UPROW,14),BUF,ITYPE)
C  LETTER THE ITEM WITH THE IDENTIFICATION CODE AND ITS SITE
C  NUMBER.
   60 CALL LETTER(UPDATE,UPROW,IX,IY,XBASE,YBASE,BASE,BUF)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINF FIRST(NOZZLE,UPDATF,IPIPE,I,J,K,IL,M,N,
     *XBASE,YBASE,BASE,BUF,YTRAK)
      INTEGER NOZZLE(60,4),UPDATE(30,15),IPIPE(60,11),
     *XBASE,YBASE,BASE(8),BUF(200),YTRAK
      IL=0
    1 FORMAT(1X,16HIPIPF ARRAY FULL)
      M=0

C  DETERMINE WHETHER THE ITEM IN COLUMN ONE OF ARRAY NOZZLE
C  IS ON THE SCREEN.

  501 J=J+1
      IF(J.GT.60)GO TO 629
      I=1
      IF(NOZZLE(J,1).GT.1000)GO TO 600
      IF(NOZZLE(J,1).EQ.0)GO TO 700
      IF(NOZZLE(J,1).EQ.1)CALL OFSITE(NOZZLE,UPDATE,IPIPE,J,
     *YTRAK,XBASE,YBASE,BASE,BUF)
  505 IF(NOZZLE(J,1).EQ.UPDATE(I,3))GO TO 520
      I=I+1
      IF(I.GT.30)GO TO 501
      GO TO 505

C  DETERMINE WHETHER THE ITEM IN COLUMN TWO OF ARRAY NOZZLE
C  IS ON THE SCREEN.

  520 K=1
      IF(NOZZLE(J,2).GT.1000)GO TO 604
      IF(NOZZLE(J,2).EQ.1)CALL OFSITE(NOZZLE,UPDATE,IPIPE,J,
     *YTRAK,XBASE,YBASE,BASE,BUF)
  530 IF(NOZZLE(J,2).EQ.UPDATE(K,3))GO TO 620
      K=K+1
      IF(K.GT.50)GO TO 501
      GO TO 530

C  DETERMINE WHETHER THE PIPE IN COLUMN ONE OF ARRAY NOZZLE
C  IS ON THE SCREEN.

  600 IL=1
  601 IF(NOZZLE(J,1).EQ.IPIPE(IL,1))GO TO 520
      IL=IL+1
      IF(IL.GT.60)GO TO 501
      GO TO 601

C  DETERMINE WHETHER THE PIPE IN COLUMN TWO OF ARRAY NOZZLE
C  IS ON THE SCREEN.

  604 M=1
  610 IF(NOZZLE(J,2).EQ.IPIPE(M,1))GO TO 620
      M=M+1
      IF(M.GT.60)GO TO 501
      GO TO 610

C  FIND VACANT ROW IN ARRAY IPIPE.

  620 DO 622 N=1,60
      IF(IPIPE(N,1).EQ.0)GO TO 624
  622 CONTINUE
```

```
C
      WRITE(2,1)
C  CHECK THAT THE LINE IS NOT ALREADY ON DISPLAY.
  624 DO 628 LUMP=1,60
      IF(IPIPE(LUMP,1).EQ.J+1000)GO TO 501
  628 CONTINUE
  629 RETURN
  700 J=101
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE ERASE(IPIPE,UPDATE,NOZZLE,BUF)
      INTEGER IPIPE(60,11),UPDATE(30,15),BUF(200),
     *ACTION,FOUND,NOZZLE(60,4)

C  SEE AN ITEM WITH THE LIGHT-PEN AND DELETE IT.

      JIB=ACTION(4,FOUND)
      CALL DELETE(JIB)

C  FIND ITS POSITION IN EITHER ARRAY IPIPE OR ARRAY
C  UPDATE AND SET THE ROW TO ZERO.

      DO 2 J=1,30
      IF(JIB.EQ.UPDATE(J,3))GO TO 10
    2 CONTINUE
      DO 1 J=1,60
      IF(JIB.EQ.IPIPE(J,1))GO TO 20
    1 CONTINUE
      RETURN
   10 DO 11 K=1,14
   11 UPDATE(J,K)=0
      RETURN
   20 DO 21 L=1,11
   21 IPIPE(J,L)=0

C  RECALCULATE THE PIPING COST.

      CALL INCOST(IPIPE,NOZZLE,BUF)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE INCOST(IPIPE,NOZZLE,BUF)
      INTEGER IPIPE(60,11),SUM,BUF(200),NOZZLE(60,4)
      REAL M1,M2
      KSUM=0
      I=0
    1 I=I+1
      J=IPIPE(I,1)-1000

C  K INDICATES THE PIPELINE TYPE. PIPELINES TO OFFSITE
C  ARE IGNORED (TYPE 5).

      K=IPIPE(I,11)
      IF(K.EQ.0)GO TO 51

C  CALCULATE THE LENGTH OF THE LINE IN SCREEN CO-ORDINATES.

      GO TO (10,10,10,40,51),K
   10 SUM=IABS(IPIPE(I,2)-IPIPE(I,4))+IABS(IPIPE(I,3)-IPIPE(I,5))+
     *IABS(IPIPE(I,5)-IPIPE(I,6))+IABS(IPIPE(I,4)-IPIPE(I,7))
      GO TO(20,50,50,40),K
   20 SUM=SUM+IABS(IPIPE(I,6)-IPIPE(I,8))+
     *IABS(IPIPE(I,7)-IPIPE(I,9))+IABS(IPIPE(I,8)-IPIPE(I,10))
      GO TO 50

C  DIRECT PIPING.

   40 M1=IPIPE(I,2)-IPIPE(I,9)
      M2=IPIPE(I,3)-IPIPE(I,10)
      SUM=SQRT(M1*M1+M2*M2)

C  MULTIPLY LENGTH BY COST FACTOR.

   50 KSUM=SUM*NOZZLE(J,4)+KSUM
   51 IF(I.EQ.60)GO TO 100
      GO TO 1

C  REPLACE THE PIPING COST FIGURE.

  100 CALL NEWBUF(BUF,200)
      CALL POINT(450,990,0)
      WRITE(12,101)KSUM
  101 FORMAT(6HCOST =,I5)
      CALL REPL(500)
      RETURN
      END
      SEGMENT
```

```
      SUBROUTINE NOPIPE(IPIPE)
      INTEGER IPIPE(60,11)
C  DELETE ALL PIPING FROM THE SCREEN AND RESET ARRAY IPIPE.
      DO 10 I=1,60
      IF(IPIPE(I,1).EQ.0)GO TO 10
      CALL DELETE(IPIPE(I,1))
      DO 10 J=1,11
      IPIPE(I,J)=0
   10 CONTINUE
   11 RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE LETTER(UPDATE,I,IX,IY,XBASE,YBASE,BASE,BUF)
      INTEGER UPDATE(30,15),BASE(8),BUF(200),XBASE,YBASE
  100 FORMAT(A4,I3)
C
C  SET LETTER SIZE AND POSITION CHARACTER STRING ABOVE OR BELOW
C  UNIT DEPENDING ON ITS POSITION WITH RESPECT TO THE PIPERACK.
C
      CALL SETP(0,1,0,0)
      CALL VECTOR(-23,10,0)
      IF(UPDATE(I,7).LT.0)CALL VECTOR(0,-20,0)
   20 WRITE(12,100)UPDATE(I,2),UPDATE(I,1)
C
C  RESET SIZE PARAMETER.
C
       CALL SETP(0,2,0,0)
      CALL ENTER(BASE,8)
      CALL LVEC(IX-XBASE,IY-YBASE)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE PIRACK(YTRAK,IWIDTH,IRATIO,BUP,BUF,
     *BASE,XBASE,YBASE)
      INTEGER YTRAK,BUP,BUF(300),BASE(8),XBASE,YBASE
C   GENERATE CODE TO DRAW PIPERACK ACROSS CENTRE OF PICTURE AREA.
      CALL NEWBUF(BUF,300)
      CALL ENTER(BASE,8)
      CALL LVEC(1023,1123)
      CALL LINE(0,0,1023,0,BUP,2)
      CALL LINE(0,0,1023,0,BUP,2)
      CALL ENTER(BASE,8)
      CALL LVEC(1023,923)
      CALL LINE(0,0,1023,0,BUP,2)
      CALL LINE(0,0,1023,0,BUP,2)
      CALL INSERT(98,0)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE INSET(ICODE,XBASE,YBASE,BASE,BUF,
     *ITYPES,IROW,CODE,NUMBER)
      INTEGER ICODE(ITYPES,2),XBASE,YBASE,BASE(8),
     *BUF(200),YPOINT,ACTION,FOUND,CODE

C  SET PICTURE ORIGIN CO-ORDINATES AND DELETE PIPERACK.

      JTEMP=XBASE
      KTEMP=YBASE
      XBASE=0
      YBASE=-2046
      CALL DELETE(98)
      CALL NEWBUF(BASE,8)
      CALL POINT(0,0,0)
      CALL VECTOR(XBASE/2,YBASE/2,0)
      CALL VECTOR(XBASE/2,YBASE/2,0)

C  DISPLAY LIST OF ITEM TYPES.

      YPOINT=1000
      DO 5 JACK=1,ITYPES
      CALL NEWBUF(BUF,200)
      CALL POINT(0,YPOINT,0)
      WRITE(12,4)ICODE(JACK,1)
    4 FORMAT(A4)
      CALL INSERT(JACK,0)
    5 YPOINT=YPOINT-40

C  THE USER SPECIFIES AN ITEM TYPE WITH THE LIGHT-PEN AND THE
C  LIST IS THEN DELETED.

      ILINE=ACTION(4,FOUND)
      CODE=ICODE(ILINE,1)
      DO 10 JOHN=1,ITYPES
   10 CALL DELETE(JOHN)

C  DISPLAY LIST OF THE PROCESS NUMBERS OF THE ITEM CHOSEN.

      ITOTAL=ICODE(ILINE,2)
   13 YPOINT=1000
      DO 15 JILL=1,ITOTAL
      CALL NEWBUF(BUF,200)
      CALL POINT(100,YPOINT,0)
      WRITE(12,14)JILL
   14 FORMAT(2H10,I1)
      CALL INSERT(JILL,0)
   15 YPOINT=YPOINT-40

C  THE USER SPECIFIES THE PROCESS NUMBER WITH THE LIGHT-PEN
C  AND THE LIST IS DELETED.

      NUMBER=ACTION(4,FOUND)
      DO 20 JANE=1,ITOTAL
   20 CALL DELETE(JANE)
```

```
C
C  RESET THE PICTURE ORIGIN CO-ORDINATES.

      XBASE=JTEMP
      YBASE=KTEMP
      CALL NEWBUF(BASE,8)
      CALL POINT(0,0,0)
      CALL VECTOR(XBASE/2,YBASE/2,0)
      CALL VECTOR(XBASE/2,YBASE/2,0)

C  CALCULATE THE ROW IN ARRAY DIMENS CONTAINING THE DIMENSIONS
C  OF THE PROCESS UNIT CHOSEN.

      IF(ILINE.EQ.1)GO TO 25
      ISUM=0
      K=ILINE-1
      DO 21 JSUM=1,K
   21 ISUM=ISUM+ICODE(JSUM,2)
      IROW=ISUM+NUMBER
      GO TO 2003
   25 IROW=NUMBER
 2003 RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINF INITAL(BUF)
      INTEGER BUF(200)
C
C  INSERT COST FIGURE AT ZERO.
C
      KSUM=0
      CALL NEWBUF(BUF,200)
      CALL POINT(450,990,0)
      WRITE(12,1)KSUM
    1 FORMAT(6HCOST =,I5)
      CALL INSERT(500,0)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE VTOV(UPDATE,IPIPE,I,K,N,JIX,JIY,
     *KIX,KIY,BUP,BUF,BASE,XBASE,YBASE,YTRAK)
      INTEGER UPDATE(30,15),IPIPE(60,11),BUP,
     *BUF(200),BASE(8),XBASE,YBASE,YTRAK,YLEVEL

C  VESSEL TO VESSEL PIPING. FIND A VACANT POSITION IN THE
C  PIPERACK AND GENERATE THE CODE FOR THE PIPELINE.

      CALL PIPE1(UPDATE,I,JIX,JIY,JXO,JYO,BUP,BUF,BASE)
      CALL PIPE1(UPDATE,K,KIX,KIY,KXO,KYO,BUP,BUF,BASE)
      CALL LEVEL(YLEVEL,YTRAK,IPIPE,JXO,JYO,KXO,KYO,BUF)
      CALL DRAW(JXO,JYO,KXO,KYO,YLEVEL,BUP,BUF,BASE,XBASE,YBASE)

C  ENTER PARAMETERS IN ARRAY IPIPE.

      IPIPE(N,2)=JIX
      IPIPE(N,3)=JIY
      IPIPE(N,4)=JXO
      IPIPE(N,5)=JYO
      IPIPE(N,6)=YLEVEL
      IPIPE(N,7)=KXO
      IPIPE(N,8)=KYO
      IPIPE(N,9)=KIX
      IPIPE(N,10)=KIY
      IPIPE(N,11)=1
      RETURN
      END
      SEGMENT
```

```
      SUBROUTINE VTOP(UPDATE,IPIPE,I,M,N,JIX,JIY,BUP,
     *BUF,BASE)
      INTEGER UPDATE(30,15),IPIPE(60,11),BUP,
     *BUF(200),BASE(8)
C  VESSEL TO PIPE PIPING. GENERATE THE CODE FOR THE PIPELINE.
      CALL PIPE1(UPDATE,I,JIX,JIY,JXO,JYO,BUP,BUF,BASE)
      CALL AROVEC(JXO,JYO,JXO,IPIPE(M,6),BUP,1)
      JXOZ=IPIPE(M,4)
      KXOZ=IPIPE(M,7)
      IF(JXOZ-KXOZ)641,642,642
  641 JXOZ=IPIPE(M,7)
      KXOZ=IPIPE(M,4)
  642 IF(JXO.LT.KXOZ)GO TO 643
      IF(JXO.GT.JXOZ)GO TO 644
      GO TO 645
  643 CALL AROVEC(JXO,0,KXOZ,0,BUP,1)
      IPIPE(N,7)=KXOZ
      GO TO 645
  644 CALL AROVEC(JXO,0,JXOZ,0,BUP,1)
C  ENTER THE PARAMETERS IN ARRAY IPIPE.
      IPIPE(N,7)=JXOZ
  645 IPIPE(N,2)=JIX
      IPIPE(N,3)=JIY
      IPIPE(N,4)=JXO
      IPIPE(N,5)=JYO
      IPIPE(N,6)=IPIPE(M,6)
      IPIPE(N,11)=2
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE PTOV(UPDATE,IPIPE,K,IL,N,KIX,KIY,BUP,
     *BUF,BASE)
      INTEGER UPDATE(30,15),IPIPE(60,11),BUP,
     *BUF(200),BASE(8)

C  PIPE TO VESSEL PIPING. GENERATE THE CODE FOR THE PIPELINE.

      CALL PIPE1(UPDATE,K,KIX,KIY,KXO,KYO,BUP,BUF,BASE)
      CALL ENTER(BASE,8)
      JXOZ=IPIPE(IL,4)
      KXOZ=IPIPE(IL,7)
      IF(JXOZ-KXOZ)651,652,652
  651 JXOZ=IPIPE(IL,7)
      KXOZ=IPIPE(IL,4)
  652 IF(KXO.LT.KXOZ)GO TO 653
      IF(KXO.GT.JXOZ)GO TO 654
      GO TO 655
  653 CALL LVEC(KXOZ,IPIPE(IL,6))
      CALL AROVEC(KXOZ,0,KXO,0,BUP,1)
      IPIPE(N,7)=KXOZ
      GO TO 655
  654 CALL LVEC(JXOZ,IPIPE(IL,6))
      CALL AROVEC(JXOZ,0,KXO,0,BUP,1)
      IPIPE(N,7)=JXOZ
  655 CALL ENTER(BASE,8)
      CALL LVEC(KXO,IPIPE(IL,6))
      CALL AROVEC(0,IPIPE(IL,6),0,KYO,BUP,1)

C  ENTER THE PARAMETERS IN ARRAY IPIPE.

      IPIPE(N,2)=KIX
      IPIPE(N,3)=KIY
      IPIPE(N,4)=KXO
      IPIPE(N,5)=KYO
      IPIPE(N,6)=IPIPE(IL,6)
      IPIPE(N,11)=3
      RETURN
      END
      SEGMENT
```

```
      SUBROUTINE VTOVX(UPDATE,IPIPE,NOZZLE,N,BUF,
     *BASE,XBASE,YBASE,YTRAK)
      INTEGER UPDATE(30,15),IPIPE(60,11),NOZZLE(60,4),
     *BUF(200),BASE(8),XBASE,YBASE,YTRAK,BUP,ACTION,FOUND
      LOGICAL SWITCH

C  VESSEL TO VESSEL PIPEWORK REPLACEMENT.

      J=IPIPE(N,1)-1000
      BUP=NOZZLE(J,3)

C  FIND THE ROWS IN ARRAY UPDATE OF THE PARAMETERS DEFINING
C  THE ITEMS.

      DO 5021 I=1,30
      IF(NOZZLE(J,1).EQ.UPDATE(I,3))GO TO 5022
 5021 CONTINUE
 5022 DO 5023 K=1,30
      IF(NOZZLE(J,2).EQ.UPDATE(K,3))GO TO 5024
 5023 CONTINUE

C  FIND THE CO-ORDINATES OF EACH END OF THE PIPELINE.

 5024 JIX=IPIPE(N,2)
      JIY=IPIPE(N,3)
      KIX=IPIPE(N,9)
      KIY=IPIPE(N,10)

C  START REPLACEMENT CYCLE.

    1 MODE=1
      CALL CRPOS(JIX+XBASE,JIY+YBASE)

C  FRAME INTERRUPT.

      FOUND=9
      ID=ACTION(2,FOUND)

C  DRAW THE PIPELINE FROM THE TRACKING CROSS.

    2 CALL TRAK(IX,IY)
      JIX=IX-XBASE
      JIY=IY-YBASE

C  IF KEY 4 IS PRESSED THE ROUTINE IS ENDED. KEY 2 CAUSES THE
C  THE TRACKING CROSS TO MOVE TO THE OTHER END OF THE PIPELINE.

    3 IF(SWITCH(4))GO TO 5000
      IF(SWITCH(2))GO TO(4,1),MODE
      IPIPE(N,6)=0

C  REPLACE THE PIPELINE AND CALCULATE NEW COST.

      CALL NEWBUF(BUF,200)
      CALL VTOV(UPDATE,IPIPE,I,K,N,JIX,JIY,KIX,KIY,
     *BUP,BUF,BASE,XBASE,YBASE,YTRAK)
      CALL REPL(IPIPE(N,1))
      CALL INCOST(IPIPE,NOZZLE,BUF)
```

```
C
      GO TO(2,5),MODE
C  POSITION CROSS AT OTHER END OF THE LINE.
    4 MODE=2
      CALL CRPOS(KIX+XBASE,KIY+YBASE)
C  FRAME INTERRUPT.
      FOUND=9
      ID=ACTION(2,FOUND)
C  DRAW THE PIPELINE TO THE TRACKING CROSS.
    5 CALL TRAK(IX,IY)
      KIX=IX-XBASE
      KIY=IY-YBASE
      GO TO 3
 5000 RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE VTOPX(UPDATE,IPIPE,NOZZLE,N,BUF,
     *BASE,XBASE,YBASE,YTRAK)
      INTEGER UPDATE(30,15),IPIPE(60,11),NOZZLE(60,4),
     *BUF(200),BASE(8),XBASE,YBASE,YTRAK,BUP
      LOGICAL SWITCH

C  VESSEL TO PIPE, PIPEWORK REPLACEMENT.

      J=IPIPE(N,1)-1000
      BUP=NOZZLE(J,3)
      IPIPE(N,6)=0

C  FIND ROWS IN ARRAYS UPDATE AND IPIPE CONTAINING THE
C  PARAMETERS DEFINING THE ITEM AND PIPE.

      DO 5031 I=1,30
      IF(NOZZLE(J,1).EQ.UPDATE(I,3))GO TO 5032
 5031 CONTINUE
 5032 DO 5033 M=1,60
      IF(NOZZLE(J,2).EQ.IPIPE(M,1))GO TO 5034
 5033 CONTINUE

C  IF KEY 3 IS PRESSED THE TRACKING CROSS IS POSITIONED AT
C  THE OTHER END OF THE PIPELINE.

 5034 IF(SWITCH(3))GO TO 6034
 5035 CALL TRAK(IX,IY)
      JIX=IX-XBASE
      JIY=IY-YBASE

C  REPLACE THE PIPELINE AND DISPLAY NEW COST.

      CALL NEWBUF(BUF,200)
      CALL VTOP(UPDATE,IPIPE,I,M,N,JIX,JIY,BUP,
     *BUF,BASE)
 5036 CALL REPL(IPIPE(N,1))
      CALL INCOST(IPIPE,NOZZLE,BUF)

C  PRESS KEY 4 TO EXIT FROM THE ROUTINE.

      IF(SWITCH(4))GO TO 5000
      GO TO 5035
 6034 CALL CRPOS(IPIPE(N,7)+XBASE,IPIPE(N,6)+YBASE)
      JIX=IPIPE(N,2)
      JIY=IPIPE(N,3)
      MODE=3
 6035 CALL TRAK(IX,IY)
      KIX=IX-XBASE
      KIY=IY-YBASE

C  REPLACE THE PIPELINE AND DISPLAY NEW COST.

      CALL NEWBUF(BUF,200)
      CALL VTOP(UPDATE,IPIPE,I,M,N,JIX,JIY,BUP,
     *BUF,BASE)
 6036 CALL REPL(IPIPE(N,1))
      CALL INCOST(IPIPE,NOZZLE,BUF)
      IF(SWITCH(4))GO TO 5000
```

```
C
      GO TO 6035
 5000 RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINF PTOVX(UPDATE,IPIPE,NOZZLE,N,BUF,
     *BASE,XBASF,YBASE,YTRAK)
      INTEGER UPDATF(30,15),IPIPE(60,11),NOZZLE(60,4),
     *BUF(200),BASE(8),XBASF,YBASE,YTRAK,BUP
      LOGICAL SWITCH

C  PIPE TO VESSFL, PIPEWORK REPLACEMENT.

      J=IPIPE(N,1)-1000
      BUP=NOZZLF(J,3)
      IPIPE(N,6)=0

C  FIND ROWS IN ARRAYS IPIPF AND UPDATE CONTAINING
C  THE PARAMETERS DFFINING THE PIPE AND ITEM.

      DO 5041 IL=1,60
      IF(NOZZLE(J,1).EQ.IPIPE(IL,1))GO TO 5042
 5041 CONTINUE
 5042 DO 5043 K=1,30
      IF(NOZZLE(J,2).EQ.UPDATF(K,3))GO TO 5044
 5043 CONTINUE

C  IF KEY 3 IS PRESSED THE TRACKING CROSS IS POSITIONED
C  AT THE OTHER END OF THE PIPELINE.

 5044 IF(SWITCH(3))GO TO 6044
      CALL VALUF(IA,IB,5044,N,IL,K,J,0,0,0)
 5045 CALL TRAK(IX,IY)
      KIX=IX-XBASE
      KIY=IY-YBASE

C  REPLACE THE PIPELINE AND DISPLAY THE NEW COST.

      CALL NEWBUF(BUF,200)
      CALL PTOV(UPDATE,IPIPF,K,IL,N,KIX,KIY,BUP,
     *BUF,BASE)
 5046 CALL REPL(IPIPE(N,1))
      CALL INCOST(IPIPE,NOZZLE,BUF)

C  PRESS KEY 4 TO EXIT FROM THE ROUTINE.

      IF(SWITCH(4))GO TO 5000
      GO TO 5045
 6044 CALL CRPOS(IPIPE(N,7),IPIPE(N,6))
      KIX=IPIPE(N,2)
      KIY=IPIPE(N,3)
 6045 CALL TRAK(IX,IY)
      JIX=IX-XBASE
      JIY=IY-YBASE

C  REPLACE THE PIPELINE AND DISPLAY NEW COST.

      CALL NEWBUF(BUF,200)
      CALL PTOV(UPDATE,IPIPF,K,IL,N,KIX,KIY,BUP,
     *BUF,BASE)
 6046 CALL REPL(IPIPE(N,1))
      CALL INCOST(IPIPE,NOZZLE,BUF)
      IF(SWITCH(4))GO TO 5000
```

```
C
      GO TO 6045
 5000 RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINF BOXOUT(UPDATE,XBASE,YBASE,BASE,BUF)
      INTEGER UPDATF(30,15),BASF(8),BUF(200),XBASE,
     *YBASE
C  REPLACE THE PLANT ITEMS WITHOUT GENERATING THE CODE FOR
C  THE ACCESS BOXES. IX,IY ARE THE CO-ORDINATES OF THE ORIGIN
C  OF EACH UNIT.
      DO 1 I=1,30
      IF(UPDATE(I,3).EQ.0)GO TO 3
      IX=UPDATE(I,4)+XBASE
      IY=UPDATE(I,5)+YBASE
      CALL FIGURE(UPDATF,I,IX,IY,XBASE,YBASE,BASE,BUF)
      CALL REPL(UPDATF(I,3))
    3 CONTINUE
    1 CONTINUE
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE BOXIN(UPDATE,XBASE,YBASE,BASE,BUF)
      INTEGER UPDATE(30,15),BASE(8),BUF(200),XBASE,
     *YBASE

C  REPLACE THE PLANT ITEMS GENERATING THE CODE FOR THE ACCESS
C  BOXES. IX,IY ARE THE CO-ORDINATES OF THE ORIGIN OF EACH UNIT.

      DO 1 I=1,30
      IF(UPDATE(I,3).EQ.0)GO TO 3
      IX=UPDATE(I,4)+XBASE
      IY=UPDATE(I,5)+YBASE
      CALL FIGURE(UPDATE,I,IX,IY,XBASE,YBASE,BASE,BUF)
      CALL DISBOX(UPDATE(I,8),UPDATE(I,9),UPDATE(I,10),
     *UPDATE(I,11),1)
      CALL REPL(UPDATE(I,3))
    3 CONTINUE
    1 CONTINUE
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE SETUP(YTRAK,IWIDTH,IRATIO,BUF,BASE,
     *XBASE,YBASE)
      INTEGER YTRAK,BUF(200),BASE(8),XBASE,YBASE
C  INITIALISATION ROUTINE. THE PIPERACK IS SET UP.
      CALL NEWBUF(BASE,8)
      CALL POINT(0,0,0)
      CALL VECTOR(XBASE/2,YBASE/2,0)
      CALL VECTOR(XBASE/2,YBASE/2,0)
      CALL SETRAK(.TRUE.)
      CALL INITAL(BUF)
      CALL PIRACK(YTRAK,IWIDTH,IRATIO,1,BUF,BASE,XBASE,YBASE)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE INVERT(UPDATE,UPROW,XBASE,YBASE,BUF)
      INTEGER ACTION,ITEM,FOUND,UPROW,XBASE,YBASE,UPDATE(30,14),
     *BUF(200)
C  AN ITEM IS SPECIFIED WITH THE LIGHT-PEN AND ITS POSITION IN
C  ARRAY UPDATE IS FOUND.
      ITEM=ACTION(4,FOUND)
      DO 1 I=1,30
      IF(ITEM.EQ.UPDATE(I,3))GO TO 2
    1 CONTINUE
      RETURN
C  THE PARAMETERS DEFINING THE DIMENSIONS OF THE  ITEM ARE
C  MULTIPLIED BY -1.
    2 DO 3 J=6,12
    3 UPDATE(I,J)=-UPDATE(I,J)
      LX=UPDATE(I,4)+XBASE
      LY=UPDATE(I,5)+YBASE
C  THE TRACKING CROSS IS POSITIONED A THE ITEM ORIGIN READY
C  FOR REPOSITIONING.
      CALL CRPOS(LX,LY)
      UPROW=I
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE PIPEX(IPIPE,NOZZLE,BUF,BASE,XBASE,YBASE)
      INTEGER IPIPE(60,11),NOZZLE(60,4),BUF(200),BASE(8),
     *XBASE,YBASE,ACTION,FOUND,BUP
      LOGICAL SWITCH
C
C  SPECIFY A PIPE WITH THE LIGHT-PEN AND FIND ITS POSITION IN
C  ARRAY IPIPE.
C
      K=ACTION(4,FOUND)
      DO 1 I=1,60
      IF(K.EQ.IPIPE(I,1))GO TO 2
    1 CONTINUE
      GO TO 100
C
C  SET PIPELINE TYPE TO 4 AND RESET PARAMETERS.
C
    2 IPIPE(I,11)=4
      DO 3 J=4,8
    3 IPIPE(I,J)=0
      J=K-1000
      BUP=NOZZLE(J,3)
      MODE=1
C
C  POSITION TRACKING CROSS AT ONE END OF THE LINE.
C
      JX=IPIPE(I,2)
      JY=IPIPE(I,3)
      KX=IPIPE(I,9)
      KY=IPIPE(I,10)
    5 CALL CRPOS(JX+XBASE,JY+YBASE)
C
C  PAUSE (FRAME INTERRUPT).
C
      FOUND=5
      ID=ACTION(2,FOUND)
    8 CALL TRAK(IX,IY)
      JX=IX-XBASE
      JY=IY-YBASE
      IPIPE(I,2)=JX
      IPIPE(I,3)=JY
C
C  IF SWITCH 2 IS PRESSED THE TRACKING CROSS MOVES TO THE
C  OTHER END OF THE LINE.
C
    9 IF(SWITCH(4))GO TO 50
      IF(SWITCH(2))GO TO(10,12),MODE
C
C  SET UP CODE BUFFER AND REPLACE PIPELINE WITH A SINGLE
C  ARROWED LINE.
C
      CALL NEWBUF(BUF,200)
      CALL ENTER(BASE,8)
      CALL LVEC(JX,JY)
      CALL AROVEC(JX,JY,KX,KY,BUP,1)
      CALL REPL(IPIPE(I,1))
      GO TO(8,11),MODE
   10 CALL INCOST(IPIPE,NOZZLE,BUF)
      MODE=2
```

```
C
      CALL CRPOS(KX+XBASE,KY+YBASE)
C  PAUSE (FRAME INTERRUPT).
      FOUND=5
      ID=ACTION(2,FOUND)
   11 CALL TRAK(IX,IY)
      KX=IX-XBASE
      KY=IY-YBASE
      IPIPE(I,9)=KX
      IPIPE(I,10)=KY
      GO TO 9
   12 CALL INCOST(IPIPE,NOZZLE,BUF)
      MODE=1
      GO TO 5
  100 WRITE(2,101)IA,IB
  101 FORMAT(1X,1H&,2A4,5X,8HNO MATCH)
   50 CALL INCOST(IPIPE,NOZZLE,BUF)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINF WAIT(KEY,BUF,NUMBER)
      INTEGER ACTION,BUF(200),FOUND
      KEY=0
C  SET UP CODE PUFFER. WRITE PAUSE ON THE SCREEN.
      CALL NEWBUF(BUF,200)
      CALL POINT(850,950,0)
      WRITE(12,1)NUMBER
    1 FORMAT(5HPAUSE,I4)
      CALL INSERT(99,0)
C  TIME DELAY.
    2 FOUND=5
      K=ACTION(2,FOUND)
C  THE PROGRAM WAITS FOR A KEY DEPRESSION.
      KEY=ACTION(1,FOUND)
      CALL DELETE(99)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINF PRINT2(NOZZLE)
      INTEGER NOZZLE(60,4)

C  WRITE OUT ARRAY NOZZLE TO THE LINE PRINTER.

      WRITE(2,1)
    1 FORMAT(1H1,25X,12HARRAY NOZZLE//)
      DO 10 I=1,60
      IF(NOZZLE(I,1).EQ.0)GO TO 20
   10 WRITE(2,3)I,(NOZZLE(I,J),J=1,4)
    3 FORMAT(10X,5I8)
   20 RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINF PRINT1(IPIPE)
      INTEGER IPIPE(60,11)
C  WRITE OUT ARRAY IPIPF TO THE LINF PRINTER.
      DO 1 I=1,60
      IF(IPIPE(I,1).EQ.0)GO TO 3
    1 WRITE(2,2)(IPIPE(I,J),J=1,11)
    2 FORMAT(13X,13I8)
    3 RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE PRINT3(UPDATE)
      INTEGER UPDATE(30,15)

C  WRITE OUT ARRAY UPDATE TO THE LINE PRINTER.

      DO 1 I=1,30
      IF(UPDATE(I,3).EQ.0)GO TO 1
      WRITE(2,11)I,(UPDATE(I,J),J=1,15)
    1 CONTINUE
   11 FORMAT(1X,I3,I8,A4,13I7)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE PRINT4(DIMENS,ICODE,ITNUMB)
      INTEGER DIMENS(26,5),ICODE(9,2)
C  WRITE OUT ARRAY DIMENS TO THE LINE PRINTER.
      WRITE(2,1000)
 1000 FORMAT(1H1,40X,12HARRAY DIMENS//42X,1HL,10X,1HW,10X,1HH,
     *10X,1HD,6X,5HTHETA//)
      K=1
      L=1
      DO 1 I=1,ITNUMB
      IF(ICODE(K,2).LT.1)GO TO 1005
      WRITE(2,5000)I,ICODE(K,1),L,(DIMENS(I,J),J=1,5)
      IF(L.LT.ICODE(K,2))GO TO 1009
 1005 L=1
      K=K+1
      GO TO 1
 1009 L=L+1
    1 CONTINUE
 5000 FORMAT(19X,I2,5X,A4,4H  10,I1,5I10)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE PRINT5(CLEAR,ICODE,ITNUMB)
      INTEGER CLEAR(26,6),ICODE(9,2)

C  WRITE OUT ARRAY CLEAR TO THE LINE PRINTER.

      WRITE(2,2000)
 2000 FORMAT(1H1,50X,11HARRAY CLEAR//45X,2HL1,8X,2HL2,8X,2HW1,
     *8X,2HW2,8X,2HH1,8X,2HH2//)
      K=1
      L=1
      DO 4  I=1,ITNUMB
      WRITE(2,2001)I,ICODE(K,1),L,(CLEAR(I,J),J=1,6)
      IF(L.LT.ICODE(K,2))GO TO 1010
      L=1
      K=K+1
      GO TO 4
 1010 L=L+1
    4 CONTINUE
 2001 FORMAT(21X,I2,5X,A4,4H  10,I1,6I10)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE PRINTX(IPIPE,UPDATE,NOZZLE,DIMENS,ICODE,
     *ITNUMB,CLEAR)
      INTEGER IPIPE(60,11),UPDATE(30,14),NOZZLE(60,4),
     *DIMENS(26,5),ICODE(9,2),CLEAR(26,6)
      LOGICAL SWITCH

C  WRITES OUT ARRAYS UPDATE, IPIPE, NOZZLE, DIMENS AND CLEAR.
C  AS LONG AS KEY 1 REMAINS DEPRESSED.

      CALL PRINT3(UPDATE)
      CALL PRINT1(IPIPE)
      IF(.NOT.SWITCH(1))RETURN
      CALL PRINT2(NOZZLE)
      IF(.NOT.SWITCH(1))RETURN
      CALL PRINT4(DIMENS,ICODE,ITNUMB)
      IF(.NOT.SWITCH(1))RETURN
      CALL PRINT5(CLEAR,ICODE,ITNUMB)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE INTAPE(UPDATE,IPIPE)
      INTEGER UPDATE(30,15),IPIPE(60,11)
C  WRITES OUT ARRAY UPDATE TO PAPER TAPE.
      ICODE=99
      DO 2 J=1,30
      IF(UPDATE(J,3).EQ.0)GO TO 1
      WRITE(5,111)UPDATE(J,1)
      WRITE(5,109)UPDATE(J,2)
      WRITE(5,112)(UPDATE(J,K),K=3,12)
      WRITE(5,110)(UPDATE(J,K),K=13,15)
    1 CONTINUE
    2 CONTINUE
      WRITE(5,111)ICODE
C  WRITE OUT ARRAY IPIPE TO PAPER TAPE. THE NUMBER 99 MARKS
C  THE END OF EACH ARRAY.
      DO 4 J=1,60
      IF(IPIPE(J,1).EQ.0)GO TO 3
      WRITE(5,111)IPIPE(J,1)
      WRITE(5,112)(IPIPE(J,K),K=2,11)
    3 CONTINUE
    4 CONTINUE
      WRITE(5,111)ICODE
      RETURN
  109 FORMAT(A4)
  110 FORMAT(3I8)
  111 FORMAT(I8)
  112 FORMAT(10I8)
      END
      SEGMENT
```

```
C
      SUBROUTINE DISCUB(IX,IY,INCX,INCY,BUF,ITYPE)
      INTEGER BUF(200)
C  GENERATE CODE FOR DISPLAYING UNITS REPRESENTED BY A CUBE.
      CALL LINE(0,0,-INCX/2,0,2,ITYPE)
      CALL LINE(0,0,0,INCY,2,ITYPE)
      CALL LINE(0,0,INCX,0,2,ITYPE)
      CALL LINE(0,0,0,-INCY,2,ITYPE)
      CALL LINE(0,0,-INCX/2,0,2,ITYPE)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE REPUT(UPDATE,IPIPE,NOZZLE,BASE,BUF,XBASE,YBASE)
      INTEGER UPDATE(30,15),IPIPE(60,11),BASE(8),BUF(200),
     *XBASE,YBASE,BUP,INUM(15),NOZZLE(60,4)

C  READ A ROW OF ARRAY UPDATE FROM PAPER TAPE. DISPLAY THE ITEM
C  AND READ THE NEXT ROW.

      K=0
    1 READ(3,110)INUM(1)
      IF(INUM(1).EQ.99)GO TO 5
      READ(3,109)INUM(2)
      READ(3,112)(INUM(I),I=3,12)
      READ(3,111)(INUM(I),I=13,15)
      K=K+1
      DO 3 J=1,15
    3 UPDATE(K,J)=INUM(J)
      CALL FIGURE(UPDATE,K,INUM(4)+XBASE,INUM(5)+YBASE,
     *XBASE,YBASE,BASE,BUF)
      CALL INSERT(INUM(3),0)
      GO TO 1
    5 K=0

C  READ A ROW OF ARRAY IPIPE FROM PAPER TAPE. DISPLAY THE
C  PIPELINE AND READ THE NEXT ROW.

    6 READ(3,110)INUM(1)
      IF(INUM(1).EQ.99)GO TO 1000
      READ(3,112)(INUM(I),I=2,11)
      K=K+1
      DO 7 J=1,11
    7 IPIPE(K,J)=INUM(J)
      JA=INUM(1)-1000
      BUP=NOZZLE(JA,3)
      CALL NEWBUF(BUF,200)
      CALL ENTER(BASE,8)
      JA=INUM(11)
      GO TO(10,20,30,40,50),JA

C  VESSEL TO VESSEL, PIPEWORK.

   10 CALL LVEC(INUM(2),INUM(3))
      CALL LINE(INUM(2),INUM(3),INUM(4),INUM(5),BUP,1)
      CALL DRAW(INUM(4),INUM(5),INUM(7),INUM(8),INUM(6),
     *BUP,BUF,BASE,XBASE,YBASE)
      CALL LINE(INUM(7),INUM(8),INUM(9),INUM(10),BUP,1)
      GO TO 60

C  VESSEL TO PIPE, PIPEWORK.

   20 CALL LVEC(INUM(2),INUM(3))
      CALL LINE(INUM(2),INUM(3),INUM(4),INUM(5),BUP,1)
      CALL AROVEC(0,INUM(5),0,INUM(6),BUP,1)
      IF(INUM(7).EQ.0)GO TO 60
      CALL AROVEC(INUM(4),0,INUM(7),0,BUP,1)
      GO TO 60
```

```
C
C  PIPE TO VESSEL, PIPEWORK.

   30 IF(INUM(7).EQ.0)GO TO 31
      CALL LVEC(INUM(7),INUM(6))
      CALL AROVEC(INUM(7),0,INUM(4),0,BUP,1)
      GO TO 32
   31 CALL LVEC(INUM(4),INUM(6))
   32 CALL AROVEC(0,INUM(6),0,INUM(5),BUP,1)
      CALL LINE(INUM(4),INUM(5),INUM(2),INUM(3),BUP,1)
      GO TO 60

C  DIRECT PIPING.

   40 CALL LVEC(INUM(2),INUM(3))
      CALL AROVEC(INUM(2),INUM(3),INUM(9),INUM(10),BUP,1)
      GO TO 60

C  PIPING TO OFFSITE.

   50 KA=INUM(2)
      GO TO(51,55),KA
   51 CALL LVEC(INUM(4),INUM(5))
      CALL AROVEC(0,INUM(5),0,INUM(6),BUP,1)
      CALL AROVEC(INUM(4),0,INUM(7),0,BUP,1)
      CALL ARROW(BUP,BUF,INUM(3))
      GO TO 60
   55 CALL LVEC(INUM(7),INUM(6))
      CALL ARROW(BUP,BUF,INUM(3))
      CALL AROVEC(INUM(7),0,INUM(4),0,BUP,1)
      CALL AROVEC(0,INUM(6),0,INUM(5),BUP,1)
   60 CALL INSERT(INUM(1),0)
      GO TO 6

C  CALCULATE PIPING COST.

 1000 CALL INCOST(IPIPE,NOZZLE,BUF)
      RETURN
  109 FORMAT(A4)
  110 FORMAT(I8)
  111 FORMAT(3I8)
  112 FORMAT(10I8)
      END
      SEGMENT
```

```
C
      SUBROUTINF ZONAL(UPDATE,DIMENS,ICODE,XBASF,YBASF,BASE,
     *BUF,ITYPES,ITNUMB,CLEAR,SCALE,UPROW)
      INTEGER UPDATF(30,15),DIMENS(ITNUMB,5),XBASE,YBASE,
     *BASE(8),BUF(200),SCALF,CLEAR(ITNUMB,6),UPROW,CODE
     *,BOXEXT(6),ICODE(ITYPFS,2)
      REAL ECLEAR(6)
      ESCALE=SCALE
      IRATIO=10

C  A PLANT ITEM IS SPECIFIED. IROW IS THE ROW IN ARRAYS DIMENS
C  AND CLEAR CONTAINING THE DIMENSIONS AND CLEARANCES OF THE
C  ITEM.

    3 CALL INSET(ICODE,XBASF,YBASE,BASE,BUF,ITYPES,IROW,CODE,NUMBER)
      DO 32 KY=1,6
   32 ECLEAR(KY)=CLEAR(IROW,KY)/ESCALE

C  CALCULATE THF SIZE AND SHAPE OF THE ITEM REPRESENTATION.

      CALL BOX(CODE,DIMENS(IROW,1),DIMENS(IROW,2),DIMENS(IROW,3),
     *DIMENS(IROW,4),DIMENS(IROW,5),ECLEAR,BOXEXT,SCALE,KFORM)

C  FIND FIRST VACANT ROW IN ARRAY UPDATE.

      UPROW=1
   35 IF(UPDATE(UPROW,3).EQ.0)GO TO 40
      UPROW=UPROW+1
      GO TO 35

C  ENTER DATA IN ARRAY UPDATE.

   40 UPDATE(UPROW,1)=100+NUMBER
      UPDATE(UPROW,2)=CODE
      UPDATE(UPROW,3)=IROW+100
      UPDATE(UPROW,4)=850-XBASE
      UPDATE(UPROW,5)=850-YBASE
      UPDATE(UPROW,6)=(DIMENS(IROW,2)*IRATIO)/SCALE
      UPDATE(UPROW,7)=(DIMENS(IROW,1)*IRATIO)/SCALE
      UPDATE(UPROW,8)=(-BOXEXT(5)*IRATIO)/SCALE
      UPDATE(UPROW,9)=(BOXEXT(4)*IRATIO)/SCALE
      UPDATE(UPROW,10)=(BOXEXT(2)*IRATIO)/SCALE
      UPDATE(UPROW,11)=(BOXEXT(1)*IRATIO)/SCALE
      UPDATE(UPROW,12)=(DIMENS(IROW,4)*IRATIO)/SCALF
      UPDATE(UPROW,13)=KFORM
      UPDATE(UPROW,14)=DIMENS(IROW,5)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE ELVATE(IPIPE,NOZZLE,BASE,BUF,XBASE,YBASE)
      INTEGER IPIPE(60,11),NOZZLE(60,4),BASE(8),XBASE,
     *YBASE,ACTION,FOUND,BUP
      LOGICAL SWITCH
C
C  A PIPELINE IS SPECIFIED BY THE USER. IF KEY 4 IS HELD DOWN
C  MODE IS SET NEGATIVE.
C
      MODE=10
      K=ACTION(4,FOUND)
      IF(SWITCH(4))MODE=-10
C
C  LOCATE THE RELEVANT ROW IN ARRAY IPIPE.
C
      DO 1 I=1,60
      IF(K.EQ.IPIPE(I,1))GO TO 2
    1 CONTINUE
      GO TO 100
C
C  VALID ONLY FOR TYPE 1 PIPELINES.
C
    2 IF(IPIPE(I,11).NE.1)RETURN
      J=IPIPE(I,1)-1000
      BUP=NOZZLE(J,3)
C
C  REPLACE THE PIPELINE.
C
    3 CALL NEWBUF(BUF,200)
      CALL ENTER(BASE,8)
      CALL LVEC(IPIPE(I,2),IPIPE(I,3))
      CALL LINE(IPIPE(I,2),IPIPE(I,3),IPIPE(I,4),IPIPE(I,5),BUP,1)
      CALL DRAW(IPIPE(I,4),IPIPE(I,5),IPIPE(I,7),IPIPE(I,8),
     *IPIPE(I,6),BUP,BUF,BASE,XBASE,YBASE)
      CALL LINE(IPIPE(I,7),IPIPE(I,8),IPIPE(I,9),IPIPE(I,10),BUP,1)
      CALL REPL(IPIPE(I,1))
      CALL INCOST(IPIPE,NOZZLE,BUF)
C
C  IF ANY KEY IS PRESSED THE ROUTINE EXITS.
C
      FOUND=20
      K=ACTION(3,FOUND)
      IF(FOUND.EQ.1)GO TO 102
C
C  INCREMENT THE LEVEL OF THE PIPELINE IN THE PIPERACK BY
C  MODE. THE LINE MAY ONLY LIE BETWEEN THE PIPERACK BOUNDARIES.
C
      IF(IPIPE(I,6).EQ.923)MODE=10
      IF(IPIPE(I,6).EQ.1123)MODE=-10
      IPIPE(I,6)=IPIPE(I,6)+MODE
      GO TO 3
  100 WRITE(2,101)IA,IB
  101 FORMAT(1X,1H&,2A4,5X,8HNO MATCH)
  102 RETURN
      END
      SEGMENT
```

```
      SUBROUTINE ZERO(IPIPE,UPDATE)
      INTEGER IPIPE(60,11),UPDATE(30,15)
C  SET THE ELEMENTS OF ARRAY IPIPE TO ZERO.
      DO 1 I=1,60
      DO 1 J=1,11
    1 IPIPE(I,J)=0
C  SET THE ELEMENTS OF ARRAY UPDATE TO ZERO.
C  THE FIFTEENTH COLUMN IS SET TO ONE.
      DO 2 I=1,30
      DO 2 J=1,14
    2 UPDATE(I,J)=0
      DO 3 J=1,30
    3 UPDATE(J,15)=1
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE REMOVE(UPDATE,IPIPE,NOZZLE,XBASE,YBASE,BASE,
     *BUF,YTRAK)
      INTEGER UPDATE(30,15),IPIPE(60,11),NOZZLE(60,4),
     *ACTION,FOUND,KPIPE(8),XBASE,YBASE,BASE(8),BUF(200),
     *YLEVEL,YTRAK,XDIF,YDIF,BUP
      LOGICAL SWITCH
      L=0
      I=0
      KMODE=1
      YTRAK=1123

C  INITIALISE ARRAY KPIPE.

      IA=4HREMO
      IB=4HVE
      DO 15 N=1,8
   15 KPIPE(N)=0

C  AN ITEM IS SPECIFIED WITH THE LIGHT-PEN. ONLY PLANT UNITS
C  CAN BE SEEN WITH THE PEN. THE ROW IN ARRAY UPDATE IS FOUND.

  131 ITEM=ACTION(4,FOUND)
      IF(ITEM.GT.1000.OR.ITEM.LT.100)GO TO 131
      DO 1 M=1,30
      IF(ITEM.EQ.UPDATE(M,3))GO TO 3
    1 CONTINUE
      GO TO 1000

C  THE ROWS IN ARRAY IPIPE OF ALL PIPELINES GOING TO OR FROM
C  THE ITEM TO BE REPOSITIONED ARE STORED IN ARRAY KPIPE.

    3 I=I+1
      IF(I.GT.60)GO TO 5
      IF(IPIPE(I,1).EQ.0)GO TO 3
      J=IPIPE(I,1)-1000
      IF(ITEM.EQ.NOZZLE(J,1).OR.ITEM.EQ.NOZZLE(J,2))GO TO 4
      GO TO 3
    4 L=L+1
      IF(L.GT.8)GO TO 1001
      KPIPE(L)=I
      GO TO 3

C  REPOSITION THE UNIT ON THE SCREEN WITH THE LIGHT-PEN. IF
C  THE UNIT CROSSES THE PIPETRACK SWOPP IS CALLED.

    5 CALL ORDER(KPIPE)
   17 IXX=UPDATE(M,4)+XBASE
      IYY=UPDATE(M,5)+YBASE
      IF(IYY-YBASE.LT.1023)KMODE=2
      CALL CRPOS(IXX,IYY)
      CALL BOXIN(UPDATE,XBASE,YBASE,BASE,BUF)
      CALL NEWBUF(BUF,200)
      CALL POINT(460,950,0)
      WRITE(12,5001)
      CALL INSERT(99,0)
      IFTURN=KMODE
    6 CALL TRAK(IX,IY)
      GO TO(31,41),KMODE
```

```
C
   31 IF(IY-YBASE.LT.1023)GO TO 32
      GO TO 9000
   32 KMODE=2
      CALL SWOPP(UPDATE,M,BUF)
      GO TO 9000
   41 IF(IY-YBASE.GT.1023)GO TO 42
      GO TO 9000
   42 KMODE=1
      CALL SWOPP(UPDATE,M,BUF)
 9000 CALL FIGURE(UPDATE,M,IX,IY,XBASE,YBASE,BASE,BUF)
      CALL DISBOX(UPDATE(M,8),UPDATE(M,9),UPDATE(M,10),
     *UPDATE(M,11),1)
      CALL REPL(UPDATE(M,3))
      MODE=3

C  IF KEY 4 IS PRESSED THE ROUTINE EXITS. IF KEY 1 IS
C  PRESSED THE PIPELINES ARE REPOSITIONED.

      IF(SWITCH(4))MODE=1
      IF(SWITCH(1))MODE=2
      GO TO(12,10,6),MODE

C  CALCULATE HOW FAR THE UNIT ORIGIN HAS MOVED.

   10 XDIF=IX-XBASE-UPDATE(M,4)
      YDIF=IY-YBASE-UPDATE(M,5)
   12 UPDATE(M,4)=IX-XBASE
      UPDATE(M,5)=IY-YBASE
      CALL DELETE(99)
      GO TO(6000,20),MODE

C  REDRAW THE RELEVANT PIPELINES TO THE UNIT.

   20 IF(IFTURN.EQ.KMODE)IFTURN=0
      IF(IFTURN.EQ.2)IFTURN=1
      CALL PIMOVE(UPDATE,M,IPIPE,NOZZLE,XBASE,YBASE,BASE,
     *BUF,YTRAK,KPIPE,XDIF,YDIF,IXX,IYY,IFTURN)
      GO TO 17
 1000 WRITE(2,5000)IA,IB
 1001 WRITE(2,5000)IA,IB
 6000 CALL BOXOUT(UPDATE,XBASE,YBASE,BASE,BUF)
      RETURN
 5000 FORMAT(1X,1H&,2A4,8HNO MATCH)
 5001 FORMAT(10HREPOSITION)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINF PIMOVE(UPDATE,M,IPIPE,NOZZLE,XBASE,YBASE,BASF,
     *BUF,YTRAK,KPIPE,XDIF,YDIF,IXX,IYY,IFTURN)
      INTEGER UPDATF(30,15),IPIPE(60,11),NOZZLE(60,4),BASF(8),
     *XBASE,YBASE,ACTION,FOUND,KPIPF(8),BUF(200),
     *YLEVEL,YTRAK,XDIF,YDIF,BUP
      IA=4HPIMO
      IB=4HVE
C  REPLACE EACH PIPELINF.
      DO 700 L=1,8
      I=KPIPE(L)
      IF(I.FQ.0)GO TO 699
      IPIPE(I,6)=0
      J=IPIPE(I,1)-1000
      JA=IPIPF(I,11)
      CALL NEWBUF(BUF,200)
      CALL ENTER(BASE,8)
      GO TO(21,21,21,21,250),JA
   21 BUP=NOZZLF(J,3)
C  FIND THE NEW PIPFLINE CO-ORDINATFS.
      JIX=IPIPE(I,2)
      JIY=IPIPE(I,3)
      KIX=IPIPE(I,9)
      KIY=IPIPE(I,10)
      JYMOVE=JIY+YDIF-(2*IFTURN*(JIY-IYY+YBASE))
      KYMOVE=KIY+YDIF-(2*IFTURN*(KIY-IYY+YBASE))
      NONCOL=2
      IF(UPDATE(M,3).EQ.NOZ7LE(J,2))NONCOL=1
      GO TO(50,100,100,50),JA
   50 DO 51 N=1,30
      IF(NOZZLE(J,NONCOL).EQ.UPDATE(N,3))GO TO 52
   51 CONTINUE
      GO TO 1002
   52 GO TO(60,1002,1002,200),JA
C  VESSEL TO VESSEL PIPEWORK, JA=1.
   60 GO TO(61,70),NONCOL
   61 CALL VTOV(UPDATE,IPIPE,N,M,I,JIX,JIY,
     *KIX+XDIF,KYMOVE,BUP,BUF,BASE,XBASE,YBASE,YTRAK)
      GO TO 600
   70 CALL VTOV(UPDATE,IPIPE,M,N,I,JIX+XDIF,
     *JYMOVE,KIX,KIY,BUP,BUF,BASE,XBASE,YBASE,YTRAK)
      GO TO 600
  100 DO 101 N=1,60
      IF(NOZZLE(J,NONCOL).EQ.IPIPE(N,1))GO TO 102
  101 CONTINUE
      GO TO 1003
  102 GO TO(1002,110,150),JA
C  VESSFL TO PIPE PIPEWORK, JA=2.
  110 CALL VTOP(UPDATE,IPIPF,M,N,I,JIX+XDIF,JYMOVE,
     *BUP,BUF,BASE)
      GO TO 600
```

```
C
C  PIPE TO VESSEL PIPEWORK, JA=3.

  150 CALL VTOP(UPDATE,IPIPE,M,N,I,JIX+XDIF,JYMOVE,
     *BUP,BUF,BASE)
      GO TO 600

C  DIRECT PIPING, JA=4.

  200 CONTINUE
      GO TO(201,210),NONCOL
  201 CALL LVEC(JIX,JIY)
      CALL AROVEC(JIX,JIY,KIX+XDIF,KYMOVE,BUP,1)
      IPIPE(I,9)=KIX+XDIF
      IPIPE(I,10)=KYMOVE
      GO TO 600
  210 CALL LVEC(JIX+XDIF,JYMOVE)
      CALL AROVEC(JIX+XDIF,JYMOVE,KIX,KIY,BUP,1)
      IPIPE(I,2)=JIX+XDIF
      IPIPE(I,3)=JYMOVE
      GO TO 600

C  PIPING TO OFFSITE, JA=5.

  250 J=J+1000
      CALL OFSITE(NOZZLE,UPDATE,IPIPE,J,YTRAK,XBASE,YBASE,BASE,BUF)
      GO TO 699
  600 CALL REPL(IPIPE(I,1))
  699 CONTINUE
  700 CONTINUE

C  ADJUST PIPING COST.

      CALL INCOST(IPIPE,NOZZLE,BUF)
      RETURN
 1002 WRITE(2,5000)IA,IB
 1003 WRITE(2,5000)IA,IB
      RETURN
 5000 FORMAT(1X,1H&,2A4,8HNO MATCH)
      END
      SEGMENT
```

```
C
      SUBROUTINE ORDER(KPIPE)
      INTEGER KPIPE(8)
C  PLACE THE VARIABLES IN ARRAY KPIPE IN ASCENDING ORDER.
      DO 3 J=1,7
      M=J
      MA=J+1
      DO 2 I=MA,8
      IF(KPIPE(I).LT.KPIPE(M))M=I
    2 CONTINUE
      ITEMP=KPIPE(J)
      KPIPE(J)=KPIPE(M)
    3 KPIPE(M)=ITEMP
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE DISCYL(IX,IY,L,D,THETA,BUF,ITYPE)
      INTEGER L,D,THETA,BUF(200)
      IF(THETA.EQ.90)GO TO 100
      ETHETA=THETA
      RADIAN=(ETHETA/360.)*2.0*3.14159
      ITEMP=L*COS(RADIAN)
C  HORIZONTAL CYLINDER - DRAW A RECTANGLE.
      CALL LINE(0,0,-D/2,0,2,ITYPE)
      CALL LINE(0,0,0,ITEMP,2,ITYPE)
      CALL LINE(0,0,D,0,2,ITYPE)
      CALL LINE(0,0,0,-ITEMP,2,ITYPE)
      CALL LINE(0,0,-D/2,0,2,ITYPE)
      CALL VECTOR(0,ITEMP,0)
      RETURN
C  VERTICAL CYLINDER - DRAW A CIRCLE.
  100 CALL VECTOR(-D/2,0,0)
      CALL ARC(-D/2,0,-D/2,0,1)
      CALL VECTOR(D/2,0,0)
      CALL VECTOR(0,D/2,0)
      RETURN
      END
      SEGMENT
```

```
      SUBROUTINE LEVEL(YLEVEL,YTRAK,IPIPE,JX,JY,KX,KY,BUF)
      INTEGER YLEVEL,YTRAK,IPIPE(60,11),BUF(200),XB,XL,
     *YB,YL,XB2,XL2,YMIN
      YMIN=YTRAK-200
      MODE=3

C  FIND THE POSITIONS OF THE ITEMS WITH RESPECT TO THE
C  PIPERACK. MODE=1, BOTH UNITS ABOVE THE PIPERACK.
C  MODE=2, BOTH UNITS BELOW THE PIPERACK. MODE=3, ONE
C  UNIT EACH SIDE OF THE PIPERACK.

      XB=JX
      XL=KX
      YB=JY
      YL=KY
      IF(JX-KX)10,20,20
   10 XB=KX
      XL=JX
   20 IF(JY-KY)21,22,22
   21 YB=KY
      YL=JY
   22 IF(YL.GE.YTRAK)MODE=1
      IF(YB.LE.YMIN)MODE=2
      GO TO(30,40,50),MODE

C  CHOOSE A LEVEL.

   30 YLEVEL=YTRAK
      GO TO 60
   40 YLEVEL=YMIN
      GO TO 60
   50 YLEVEL=YMIN-30+(YTRAK-YMIN)/2
   60 I=0
   61 I=I+1
      XB2=IPIPE(I,4)
      XL2=IPIPE(I,7)
      IF(XB2-XL2)65,70,70
   65 XB2=IPIPE(I,7)
      XL2=IPIPE(I,4)

C  IF THIS LEVEL IS OCCUPIED BY ANY OTHER PIPES, CHECK
C  THAT THERE IS NO INTERFERENCE. IF SO, CHANGE THE LEVEL AND
C  TRY AGAIN.

   70 IF(YLEVEL.EQ.IPIPE(I,6))GO TO 71
      IF(I.EQ.60)GO TO 2000
      GO TO 61
   71 IF(XB2.GT.XL.AND.XB2.LT.XB.OR.
     *XL2.GT.XL.AND.XL2.LT.XB.OR.
     *XL2.LE.XL.AND.XB2.GE.XB)GO TO(1,2,3,1),MODE
      IF(I.EQ.60)GO TO 2000
      GO TO 61
    1 YLEVEL=YLEVEL-10
      IF(YLEVEL.LT.YMIN)GO TO 1000
      GO TO 60
    2 YLEVEL=YLEVEL+10
      IF(YLEVEL.GT.YTRAK)GO TO 1000
      GO TO 60
```

```
    3 YLEVEL=YLEVEL+10
      IF(YLEVEL.GT.YTRAK)GO TO 4
      GO TO 60
    4 YLEVEL=YMIN-40+(YTRAK-YMIN)/2
      MODE=4
      GO TO 60
 1000 WRITE(2,1001)
 1001 FORMAT(1X,10HTRACK FULL/)
 2000 RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE DRAW(JX,JY,KX,KY,YLEVEL,BUP,BUF,BASE,XBASE,YBASE)
      INTEGER BUP,BUF(200),YLEVEL,BASE(8)
C  GENERATE THE CODE FOR THE PIPELINES.
      CALL ENTER(BASE,8)
      CALL LVEC(JX,JY)
      CALL AROVEC(0,JY,0,YLEVEL,BUP,1)
      CALL AROVEC(JX,0,KX,0,BUP,1)
      CALL AROVEC(0,YLEVEL,0,KY,BUP,1)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINF SWOPP(UPDATE,I,BUF)
      INTEGER UPDATF(30,14),BUF(200)
C  MULTIPLY DIMENSIONS BY -1.
      DO 1 J=6,12
    1 UPDATE(I,J)=-UPDATE(I,J)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINF PIPE1(UPDATE,UPROW,IXX,IYY,XO,YO,BUP,BUF,BASF)
      INTEGER UPDATE(30,14),XL,XR,YT,YB,XO,YO,BUF(200),UPROW,
     *BASE(8)
      IRATIO=10
      CALL ENTER(BASE,8)
      CALL LVEC(IXX,IYY)
      IX=UPDATE(UPROW,4)
      IY=UPDATE(UPROW,5)
C
C  IF THE UNIT IS CYLINDRICAL, THE WIDTH IS EQUAL TO THE DIAMETFR.
C
      IWIDTH=UPDATE(UPROW,6)
      IF(UPDATE(UPROW,6).EQ.0)IWIDTH=UPDATE(UPROW,12)
      IF(UPDATE(UPROW,7).LT.0)GO TO 10
C
C  CALCULATE THF CO-ORDINATES OF THF SIDES OF THE RECTANGLE.
C
      IF(UPDATE(UPROW,14).EQ.90.OR.UPDATE(UPROW,7).FQ.0)GO TO 75
      XL=IX-IWIDTH/2
      XR=IX+IWIDTH/2
      YT=IY+UPDATE(UPROW,7)
      YB=IY
      GO TO 20
C
C  INVERTED ITEM. CALCULATE THE CO-ORDINATES OF THE SIDES OF
C  THE RECTANGLF.
C
   10 XL=IX+IWIDTH/2
      XR=IX-IWIDTH/2
      YT=IY
      YB=IY+UPDATE(UPROW,7)
C
C  CALCULATE WHICH SIDE THE POINT IXX,IYY IS CLOSEST TO, AND
C  DRAW THE NOZZLE PERPENDICULAR TO THIS SIDE.
C
   20 IF(IXX.GT.XL-5.AND.IXX.LT.XL+5)GO TO 40
      IF(IXX.GT.XR-5.AND.IXX.LT.XR+5)GO TO 50
      IF(IYY.GT.YT-5.AND.IYY.LT.YT+5)GO TO 60
      IF(IYY.GT.YB-5.AND.IYY.LT.YB+5)GO TO 70
      GO TO 75
   40 CALL VECTOR(XL-IXX,0,0)
      CALL LINE(0,0,-IRATIO,0,BUP,1)
      XO=XL-IRATIO
      YO=IYY
      GO TO 80
   50 CALL VECTOR(XR-IXX,0,0)
      CALL LINE(0,0,IRATIO,0,BUP,1)
      XO=XR+IRATIO
      YO=IYY
      GO TO 80
   60 CALL VECTOR(0,YT-IYY,0)
      CALL LINE(0,0,0,IRATIO,BUP,1)
      XO=IXX
      YO=YT+IRATIO
      GO TO 80
   70 CALL VECTOR(0,YB-IYY,0)
      CALL LINE(0,0,0,-IRATIO,BUP,1)
      XO=IXX
```

```
C
      YO=YB-IRATIO
      GO TO 80
   75 XO=IXX
      YO=IYY
   80 RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINF ITRASE(UPDATE,XBASF,YBASE,RASE,BUF)
      INTEGER UPDATF(30,15),XRASE,YRASE,BASF(8),BUF(200),
     *ACTION,FOUND
      LOGICAL SWITCH
      ITYPE=2
C
C  SPECIFY AN ITEM WITH THE LIGHT-PFN. ITYPF DEFINES WHFTHER
C  THE ITEM IS SHOWN DOTTED OR DASHFD.
C
      JAB=ACTION(4,FOUND)
      IF(SWITCH(6))ITYPF=4
      IF(SWITCH(5))ITYPF=1
C
C  FIND ROW IN ARRAY UPDATE.
C
      DO 1 J=1,30
      IF(UPDATE(J,3).FQ.JAB)GO TO 4
    1 CONTINUE
      GO TO 100
C
C  REDRAW THE ITEM.
C
    4 IX=UPDATE(J,4)+XBASE
      IY=UPDATE(J,5)+YBASE
      UPDATE(J,15)=ITYPF
      CALL FIGURE(UPDATE,J,IX,IY,XBASE,YBASF,RASE,BUF)
      CALL REPL(UPDATE(J,3))
      ITYPE=1
      RETURN
  100 WRITE(2,101)
  101 FORMAT(1X,8HNO MATCH)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE DISBOX(EXORX,EXORY,
     *EXINCX,EXINCY,JSHINE)
      INTEGER EXORX,EXORY,EXINCX,EXINCY,BUP,TYPE
      BUP=1
      TYPE=1
C
C  GENERATE CODF FOR A RECTANGLF.
C
      CALL VECTOR(EXORX,EXORY,0)
      CALL LINE(0,0,-EXINCX,0,BUP,TYPE)
      CALL LINE(0,0,0,EXINCY,BUP,TYPE)
      CALL LINE(0,0,EXINCX,0,BUP,TYPE)
      CALL LINE(0,0,0,-EXINCY,BUP,TYPE)
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE LVEC(X,Y)
      INTEGER X,Y,XX,YY
      XX=X
      YY=Y
C  THE LINE FROM POINT 0,0 TO POINT X,Y IS BROKEN DOWN INTO
C  LINES OF A LENGTH SUCH THAT THE BEAM DOES NOT MOVE MORE THAN
C  1023 INCREMENTS IN THE X OR Y DIRECTIONS.
   10 IF(XX) 153,152,153
  152 IF(YY) 153,151,153
  153 IF(XX.GT.1023)GO TO 11
      IF(XX.LT.-1023)GO TO 12
      IX=XX
      GO TO 13
   11 IX=1023
      GO TO 13
   12 IX=-1023
   13 XX=XX-IX
      IF(YY.GT.1023)GO TO 21
      IF(YY.LT.-1023)GO TO 22
      IY=YY
      GO TO 23
   21 IY=1023
      GO TO 23
   22 IY=-1023
   23 YY=YY-IY
      CALL VECTOR(IX,IY,0)
      GO TO 10
  151 RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINF FOX(FACTOR,ESCALE,IFACTO)
C  ROUND UP OR DOWN TO THE NEAREST INTEGER.
      IF(FACTOR.LT.0)IFACTO=FACTOR*FSCALE-0.1
      IF(FACTOR.GE.0)IFACTO=FACTOR*FSCALE+0.1
      RETURN
      END
      SEGMENT
```

```
C
      SUBROUTINE BOX(ITNAME,L,W,H,D,THETA,ECLEAR,BOXEXT,
     *SCALE,KFORM)
      INTEGER ITNAME,THETA,BOXEXT(6),SCALE,
     *L,H,W,D,KFORM
      REAL ECLEAR(6),BXEXT(6),BOXL,BOXW,BOXH
      DIMENSION ISPACE(16,6),KCODE(16)

      DATA ISPACE/5,6,5,5,5,5,10,6,6,6,6,10,5,5,5,4,
     *            5,6,5,5,5,5,10,6,6,6,6,10,5,5,5,4,
     *            5,5,5,5,5,5,10,4,4,4,4,10,5,5,5,4,
     *            5,5,5,5,5,5,10,4,4,4,4,10,5,5,5,4,
     *            0,0,5,5,0,0,0,3,3,3,3,0,0,0,0,3,
     *            5,6,15,15,6,6,6,3,3,3,3,50,10,6,10,4/

      DATA KCODE/2HAA,2HBA,2HCA,2HCB,2HDA,2HEA,2HFA,2HHA,
     *2HHE,2HHB,2HHC,2HJA,2HKA,2HPA,2HRA,2HTA/

      ETHETA=THETA
      RADIAN=(ETHETA/360.0)*(2.0*3.14159)
      ESCALE=SCALE

C  CONVERT THE UNIT DIMENSIONS TO FEET.

      SL=L/ESCALE
      SH=H/ESCALE
      SW=W/ESCALE
      SD=D/ESCALE

C  CALCULATE THE SIZE OF THE UNIT WITH CLEARANCES.

      IF(D.GT.0)GO TO 10

C  CUBE.

      BOXL=SL+ECLEAR(1)+ECLEAR(2)
      BOXW=SW+ECLEAR(3)+ECLEAR(4)
      BOXH=SH+ECLEAR(5)+ECLEAR(6)
      KFORM=-1
      GO TO 30
   10 IF(L.GT.0)GO TO 20

C  SPHERE.

      BOXL=SD+ECLEAR(1)+ECLEAR(2)
      BOXW=SD+ECLEAR(3)+ECLEAR(4)
      BOXH=SD+ECLEAR(5)+ECLEAR(6)
      KFORM=0
      GO TO 30

C  CYLINDER.

   20 BOXL=SL*COS(RADIAN)+SD*SIN(RADIAN)+ECLEAR(1)+ECLEAR(2)
      BOXW=SD+ECLEAR(3)+ECLEAR(4)
      BOXH=SL*SIN(RADIAN)+SD*COS(RADIAN)+ECLEAR(5)+ECLEAR(6)
      KFORM=1
```

```
C
C  DETERMINE THE ITEM TYPE.

   30 DO 35 K=1,16
      IF(ITNAME.EQ.KCODE(K))GO TO 40
   35 CONTINUE

C  ADD THE ACCESS SPACE REQUIREMENTS.

   40 BXEXT(1)=BOXL+ISPACE(K,1)+ISPACE(K,2)
      BXEXT(2)=BOXW+ISPACE(K,3)+ISPACE(K,4)
      BXEXT(3)=BOXH+ISPACE(K,5)+ISPACE(K,6)

C  CALCULATE THE CO-ORDINATES OF THE ORIGIN OF THE BOX WITH
C  RESPECT TO THE ITEM ORIGIN.

      IF(KFORM)50,60,70
   50 BXEXT(4)=-ECLFAR(1)-ISPACE(K,1)
      BXEXT(5)=-ECLFAR(3)-0.5*SW-ISPACE(K,3)
      BXEXT(6)=-ECLFAR(5)-0.5*SH-ISPACE(K,5)
      GO TO 80
   60 BXEXT(4)=-ECLFAR(1)-0.5*SD-ISPACE(K,1)
      BXEXT(5)=-ECLFAR(3)-0.5*SD-ISPACE(K,3)
      BXEXT(6)=-ECLFAR(5)-0.5*SD-ISPACE(K,5)
      GO TO 80
   70 BXEXT(4)=-ECLEAR(1)-0.5*SD*SIN(RADIAN)-ISPACE(K,1)
      BXEXT(5)=-ECLFAR(3)-0.5*SD-ISPACE(K,3)
      BXEXT(6)=-ECLEAR(5)-0.5*SD*COS(RADIAN)-ISPACE(K,5)

C  ROUND UP THE VALUES TO THE NEAREST INTEGER.

   80 DO 85 J=1,6
   85 CALL FOX(BXEXT(J),ESCALE,BOXEXT(J))
      CONTINUE
      RETURN

      END
      SEGMENT
```

```
C  MAIN PROGRAM.

      FETCH RESET

      INTEGER DIMENS(26,5),ICODE(9,2),CLEAR(26,6),
     *BUF(300),
     *NOZZLE(60,4),FILE(3000),
     *BASE(8)

C  THE DIMENSIONS OF THE UNITS IN SIXTEENTHS OF AN INCH.

      DATA DIMENS/11520,11520,3072,2304,3072,3072,3072,3072,
     *2304,3072,3072,2304,1920,1920,576,576,576,576,576,
     *3072,3072,1920,2304,3840,1152,1152,
     *0,0,0,0,0,0,0,0,0,0,0,0,1152,1152,384,384,384,384,384,
     *0,0,0,0,0,0,0,
     *0,0,0,0,0,0,0,0,0,0,0,0,1152,1152,768,768,768,768,768,
     *0,0,0,0,0,0,0,
     *960,1152,576,384,576,576,576,576,384,576,576,1920,0,0,0,
     *0,0,0,0,1152,1152,768,1536,7680,768,768,
     *90,90,0,0,0,0,0,0,0,0,0,90,0,0,0,0,0,0,0,90,0,0,0,90,90,
     *90/

C  THE UNIT TYPES AND THE NUMBER OF EACH.

      DATA ICODE/2HCB,2HHE,2HHR,2HHC,2HJA,2HKA,2HPA,2HRA,2HTA,
     *2,4,2,3,1,2,5,1,6/

C  THE CLEARANCES ASSOCIATED WITH EACH UNIT.

      DATA CLEAR/0,0,384,384,384,384,384,384,384,384,384,
     *0,0,0,0,0,0,0,0,0,0,0,0,3072,0,0,
     *0,0,3072,2304,3072,3072,3072,3072,2304,3072,3072,0,0,0,
     *0,0,0,0,0,0,0,0,0,3072,0,0,
     *0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3072,0,0,
     *0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3072,0,0,
     *0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
     *0,0,0,0,0,0,0,0,0,0,0,0,6,6,0,0,0,0,0,0,0,0,0,0,0,0/

C  THE PIPE ROUTES.

      DATA NOZZLE/1,125,1,124,115,103,112,120,107,108,106,105,
     *103,111,121,126,114,1017,105,113,121,121,106,101,109,122,
     *116,122,101,107,101,117,102,110,123,118,1036,104,102,108,
     *102,119,0,0,0,0,0,0,0,0,
     *0,0,0,0,0,0,0,0,0,0,
     *125,113,124,115,103,112,120,107,108,106,105,103,111,121,
     *126,114,120,105,1006,1018,1,106,101,109,122,116,101,1021,
     *107,101,117,102,110,123,118,102,104,1,108,102,119,124,
     *0,0,0,0,0,0,0,0,
     *0,0,0,0,0,0,0,0,0,0,
     *1,1,1,1,1,1,3,3,3,3,3,3,3,3,2,2,2,2,2,1,2,1,1,3,1,
     *1,1,1,1,3,1,1,3,1,1,1,1,1,1,3,1,1,0,0,0,0,0,0,0,0,
     *0,0,0,0,0,0,0,0,0,0,
     *1,1,1,1,1,1,3,3,3,3,3,3,3,3,2,2,2,2,2,1,2,1,1,3,1,
     *1,1,1,1,3,1,1,3,1,1,1,1,1,1,3,1,1,0,0,0,0,0,0,0,0,
     *0,0,0,0,0,0,0,0,0,0/
```

```
C
      ITYPES=9
      ITNUMB=26
C  SET UP THE DISPLAY FILE.
      CALL RESET(FILE,3000,100)
      CALL SCREEN(DIMENS,ICODE,CLEAR,
     *BUF,NOZZLE,ITNUMB,ITYPES,BASE)
      STOP
      END
```

SUBPROGS	CODE START	CODE LENGTH	CONSTANT START	CONSTANT LENGTH	LOCAL WKSPACE START	LOCAL WKSPACE LENGTH	DYNAMIC ARRAYS
AROVEC	0	226	9208	66	16290	25	0
SEGMENT LENGTH		226					
ARC	0	236	9276	40	0	0	0
SEGMENT LENGTH		236					
LINE	0	373	9316	58	16315	28	0
SEGMENT LENGTH		373					
SCREEN	0	773	9432	170	16343	54	1110
SEGMENT LENGTH		773					
INPIPE	0	860	9626	218	16397	47	0
SEGMENT LENGTH		860					
VIEW	0	193	9848	34	16444	12	0
SEGMENT LENGTH		193					
ARROW	0	144	9884	74	16456	14	0
SEGMENT LENGTH		144					
OFSITE	0	549	9964	90	16470	40	0
SEGMENT LENGTH		549					
FIGURE	0	256	10060	58	16510	22	0
SEGMENT LENGTH		256					
FIRST	0	457	10118	66	16532	32	0
SEGMENT LENGTH		457					
ERASE	0	164	10186	18	16564	21	0
SEGMENT LENGTH		164					
INCOST	0	361	10204	58	16585	33	0
SEGMENT LENGTH		361					
NOPIPE	0	66	10262	10	16618	8	0
SEGMENT LENGTH		66					
LETTER	0	121	10272	50	16626	18	0
SEGMENT LENGTH		121					
PIRACK	0	152	10322	66	16644	22	0
SEGMENT LENGTH		152					
INSET	0	317	10390	106	16666	31	0
SEGMENT LENGTH		317					
INITAL	0	48	10498	26	16697	8	0
SEGMENT LENGTH		48					
VTOV	0	232	10528	50	16705	31	0
SEGMENT LENGTH		232					
VTOP	0	248	10578	42	16736	25	0
SEGMENT LENGTH		248					
PTOV	0	296	10620	50	16761	26	0

```
SEGMENT LENGTH          296

VTOVX             0     349   10670        50    16787        38        0
SEGMENT LENGTH          349

VTOPX             0     372   10720        58    16825        36        0
SEGMENT LENGTH          372

PTOVX             0     372   10778        66    16861        35        0
SEGMENT LENGTH          372

BOXOUT            0     113   10844        18    16896        17        0
SEGMENT LENGTH          113

BOXIN             0     145   10862        26    16913        19        0
SEGMENT LENGTH          145

SETUP             0     144   10890        50    16932        21        0
SEGMENT LENGTH          144

INVERT            0     122   10940        26    16953        21        0
SEGMENT LENGTH          122

PIPEX             0     389   10966        74    16974        32        0
SEGMENT LENGTH          389

WAIT              0      72   11040        50    17006        11        0
SEGMENT LENGTH           72

VALUE             0      32   11090        26    17017        10        0
SEGMENT LENGTH           32

PRINT2            0      49   11118        34    17027         6        0
SEGMENT LENGTH           49

PRINT1            0      41   11154        26    17033         6        0
SEGMENT LENGTH           41

PRINT3            0      41   11182        26    17039         6        0
SEGMENT LENGTH           41

PRINT4            0      89   11210        58    17045        12        0
SEGMENT LENGTH           89

PRINT5            0      81   11270        58    17057        12        0
SEGMENT LENGTH           81

NZPRNT            0      33   11328        26    17069         5        0
SEGMENT LENGTH           33

PRINTX            0     168   11354        18    17074        23        0
SEGMENT LENGTH          168

INTAPE            0     114   11372       106    17097        10        0
SEGMENT LENGTH          114

DISCUB            0      88   11478        34    17107        14        0
SEGMENT LENGTH           88

REPUT             0     660   11512       226    17121        34       15
```

SEGMENT LENGTH		660					
ZONAL	0	418	11740	50	17155	44	18
SEGMENT LENGTH		418					
ELVATE	0	354	11790	74	17199	27	0
SEGMENT LENGTH		354					
ZERO	0	93	11864	10	17226	9	0
SEGMENT LENGTH		93					
REMOVE	0	654	11878	146	17235	45	8
SEGMENT LENGTH		654					
PIMOVE	0	659	12024	130	17280	50	0
SEGMENT LENGTH		659					
ORDER	0	82	12154	10	17330	9	0
SEGMENT LENGTH		82					
DISCYL	0	168	12164	66	17339	20	0
SEGMENT LENGTH		168					
LEVEL	0	330	12230	42	17359	30	0
SEGMENT LENGTH		330					
DRAW	0	88	12272	42	17389	19	0
SEGMENT LENGTH		88					
SWOPP	0	57	12314	2	17408	9	0
SEGMENT LENGTH		57					
PIPE1	0	400	12316	82	17417	30	0
SEGMENT LENGTH		400					
ITRASE	0	153	12398	34	17447	20	0
SEGMENT LENGTH		153					
DISBOX	0	64	12432	34	17467	13	0
SEGMENT LENGTH		64					
LVEC	0	81	12466	10	17480	9	0
SEGMENT LENGTH		81					
FOX	0	48	12478	10	17489	3	0
SEGMENT LENGTH		48					
BOX	0	562	12488	280	17492	158	12
SEGMENT LENGTH		562					
RESET	24079	1976	12770	638	0	0	0
MAINPROG	24023	56	13410	1135	17650	557	3308
VECTOR	24005	18	14547	10	0	0	0
SETP	23979	26	14559	14	0	0	0
INSERT	23951	28	14575	12	0	0	0
ACTION	23773	178	14589	56	0	0	0
SWITCH	23747	26	14647	8	0	0	0
CRPOS	23733	14	14657	10	0	0	0
TRAK	23719	14	14669	10	0	0	0
REPL	23689	30	14681	10	0	0	0

NEWBUF	23671	18	14693	10	0	0	0
ENTER	23627	44	14705	16	0	0	0
POINT	23607	20	14723	10	0	0	0
KEYS	23573	34	14735	10	0	0	0
DELETE	23557	16	14747	10	0	0	0
SETRAK	23543	14	14759	8	0	0	0

FREE STORE 18207 TO 23542

&RUN;

APPENDIX II

Economic Pipe Size Determination

The cost of process piping represents a large proportion of the capital cost of a chemical plant. As the diameter of a pipeline increases, the capital cost increases but pumping costs will decrease. Thus an optimum pipe diameter must exist.

1. Cost of Pumping

The basic expression for the work required to move unit mass of fluid between two points; 1,2 is given by

$$W' = \int_1^2 v\,dp + \left(\frac{u_2^{\,2} - u_1^{\,2}}{2g_c}\right) + (Z_2 - Z_1)\frac{g}{g_c} + F \quad \text{ft -lb}_f\text{/lb mass} \qquad \ldots (1)$$

the following assumptions will be made

(1) the fluid is incompressible

(2) there is no change in kinetic energy

(3) gravitational differences are negligible

then $W' = (Z_2 - Z_1) + F$ ft - lb_f/lb - mass ... (2)

Experimental evidence has yielded the Fanning equation:

$$F = \frac{2fV^2L}{g_cD} \quad \text{ft - lb}_f\text{/lb - mass} \qquad \ldots (3)$$

for turbulent flow $f = \dfrac{0.04}{(N_{R_e})^{0.16}}$ ... (4)

for viscous flow $f = \dfrac{16}{(N_{R_e})}$ ... (5)

combining these equations and taking into account the frictional effect of fittings, gives for turbulent flow:

$$W' = (Z_2 - Z_1) + \frac{0.08\mu^{0.16} \; V^{1.84} \; L_e}{D^{1.16} \; \rho^{\;0.16} \; g_c} \text{ ft-lb}_f\text{/lb-mass} \quad \ldots (6)$$

$$\text{also} \quad V = \frac{4q_f}{\pi D^2}$$

$$W' = (Z_2 - Z_1) + 0.125\mu^{0.16} \;\; q_f^{\,1.84} L_e / D^{4.84} \rho^{\;0.16} \; g_c$$

$$\text{ft - lb}_f\text{/lb - mass} \ldots \; (7)$$

or

$$W' = 0.00136 \; (Z_2 - Z_1) \; q_f \rho + 0.273 \; \frac{\mu_c^{\;0.16} \;\; q_f^{\;2.84} \;\; \rho^{\;0.84} L_e}{D_i^{\;4.84}} \text{ Kw}$$

$$\ldots \; (8)$$

thus the annual cost of pumping per unit length of pipe is given by

for turbulent flow

$$C_1 = \left\{ \frac{0.00136 (Z_2 - Z_1) \; q_f \rho}{L} + 0.273 \; \frac{\mu_c^{\;0.16} q_f^{\;2.84} \rho^{0.84} (1+J)}{D_i^{\;4.84}} \right\} \frac{KH}{E}$$

$$\ldots \; (9)$$

£/year unit length of pipe

for viscous flow

$$C_1 = \left\{ \frac{0.00136 (Z_2 - Z_1) q_f \rho}{L} + 0.024 \; \frac{q_f^{\;2} \mu_c \;\; (1+J)}{D_i^{\;4}} \right\} \frac{KH}{E}$$

$$\ldots \; (10)$$

£/year unit length of pipe

2. Installed Cost of Piping

For most types of pipe, a plot of the logarithm of the pipe diameter versus the logarithm of the purchase cost per foot of pipe is essentially a straight line. Thus

$$C_p = X D_i^n \qquad \ldots(11)$$

The installed cost of piping must be calculated. Consider a line of length 325 feet with fittings:

12 flanges
16 weld ells
1 tee
1 check valve
2 gate valves

Using the costing procedure of Rase and Barrow the cost of the line for three different diameter pipelines is calculated.

Table 3

pipe diameter	purchase pipe cost-dollars	installed pipe cost-dollars	installed cost/purchase cost
2" carbon steel	125	950	7.6
" stainless - 304	1,275	2,693	2.1
" " - 316	2,009	3,605	1.7
6" carbon steel	647	4,735	7.3
" stainless - 304	6,488	14,742	2.3
" " - 316	10,403	20,526	1.9
12"carbon steel	1,829	16,798	9.1
" stainless - 304	19,241	53,874	2.8
" " - 316	29,392	74,360	2.5

if F is defined by

$$F_C = \frac{\text{installed pipe costs}}{\text{purchased pipe costs}}$$

then $C_2 = F_C X D_i^{\,N}$ £/ft ...(12)

where

	stainless steel type 304	stainless steel type 316	carbon steel
F	2.4	2.0	8.0
X	1.25	1.56	0.1042
N	1.2	1.2	1.3

3. Cost of Pump Installation

The cost of the pumping installation may be related to the annual work done in pumping per foot of pipe. A straight line may be drawn to fit the equation

$$C_3 = \frac{69.2}{L} \left(\frac{C_1}{K}\right)^{0.37} \text{ £/ft of pipe} \qquad \text{...(13)}$$

4. Determination of J

The equivalent length of pipeline is given for each type of fitting[13] in Table 4. Thus for the pipeline considered in Section 2.

Table 4

EQUIVALENT LENGTH OF PELINE FITTINGS - FT.

diameter inches	weld-ells	tee	check valve	gate valve
1	2.7	5.7	6.5	0.6
2	5.4	12	14	1.2
3	8.1	17	19	1.8
4	11	22	25	2.3
6	17	34	40	3.5
8	21	44	51	4.5
10	26	57	66	5.6
12	31	67	77	6.9
14	37	75	90	8
16	42	89	110	9
18	48	100	120	10
20	52	110	130	12
24	63	130	160	14

$$E_L = 55.6\ D_i \quad \text{ft.}$$

or

$$E_L = \frac{55.6}{325}\ D_i\ L \quad \text{ft.} \qquad \ldots(14)$$

thus the total equivalent length

$$L_e = L + E_L$$

$$L_e = L(1 + 0.171 D_i)$$

$$\therefore\ J = 0.171\ D_i$$

5. Venture Profit

A term venture profit - V is defined as

$$V_P = P - i_m I \qquad \ldots(15)$$

also

$$P = (R - dI)(1 - t) \qquad \ldots(16)$$

and $$R = S - C_T \qquad \text{...(17)}$$

A term u will now be defined by the equation

$$V_P = (S - u)(1 - t) \qquad \text{...(18)}$$

thus $u = S - \dfrac{V_p}{1 - t}$

$$= (R + C_T) - \frac{(P - i_m I)}{1 - t}$$

$$= (R + C_T) - \left\{ \frac{(R - dI)(1 - t) - i_m I}{1 - t} \right\}$$

$$u = C_T + dI + \frac{i_m I}{1 - t} \qquad \text{...(19)}$$

u is an expression of the net annual cash flow such that a positive venture profit may be achieved, i.e. the constraint of the minimum acceptable return is met. It follows that u should be minimised.

but $C_T = C_1 + bC_2 + b'C_3$

$$I = C_2 (1 - g) + C_3 (1 - q)$$

$$dI = aC_2 (1 - q) + a'C_3 (1 - q)$$

$$u = C_1 + bC_2 + b'C_3 + aC_2 (1 - q) + a'C_3 (1 - q)$$

$$+ \frac{i_m}{1 - t} \left\{ C_2 (1 - g) + C_3 (1 - q) \right\}$$

$$\text{or } u = C_1 + C_2 \left\{ b + \left(a + \frac{i_m}{1-t} \right) (1-g) \right\} + C_3 \left\{ b' + \left(a' + \frac{i_m}{1-t} \right) (1-g) \right\}$$

... (20)

For <u>turbulent</u> flow this can be fully expanded.

$$u = \left\{ 0.00136 (Z_2 - Z_1) \frac{q_f \rho}{L} + \frac{0.273 \mu_c^{0.16} q_f^{2.84} \rho^{0.84} (1 + 0.171 D_i)}{D_i^{4.84}} \right\} \frac{KH}{E}$$

$$+ F_c X D_i^N \left\{ b + \left(a + \frac{i_m}{1-t} \right) (1-g) \right\}$$

$$+ \frac{69.2}{L} \left\{ 0.00136 (Z_2 - Z_1) \frac{q_f \rho H}{L\ E} + \frac{0.273 \mu_c^{0.16} q_f^{2.84} \rho^{0.84} (1 + 0.171 D_i) H}{D_i^{4.84}\ E} \right\}^{0.37}$$

$$X \left\{ b' + \left(a' + \frac{i_m}{1-t} \right) (1-g) \right\}$$

£/year per foot of pipe

and for <u>viscous flow</u>

$$u = \left\{ 0.00136 \frac{(Z_2 - Z_1)}{L} q_f \rho + \frac{0.024 q_f^2 \mu_c (1 + 0.171 D_i)}{D_i^4} \right\} \frac{KH}{E}$$

$$+ F_c X D_i^N \left\{ b + \left(a + \frac{i_m}{1-t} \right) (1-g) \right\}$$

$$+ \frac{69.2}{L} \left\{ 0.00136 (Z_2 - Z_1) \frac{q_f \rho H}{L\ E} + \frac{0.024 q_f^2 \mu_c (1 + 0.171 D_i)\ H}{D_i^4\ E} \right\}^{0.37}$$

$$\left\{ b' + \left(a' + \frac{i_m}{1-t} \right) (1-g) \right\}$$

£/year per foot of pipe

NOMENCLATURE

a	Fraction of cost of installed piping system for annual depreciation.
a'	Fraction of cost of pumping installation for annual depreciation.
b	Fraction of cost of installed piping system for annual maintenance.
b'	Fraction of cost of pumping installation for annual maintenance.
C_1	Annual cost of pumping. £/year ft. of pipe length.
C_2	Installed cost of piping. £/ft. of pipe length.
C_3	Cost of pump installation. £/ft. of pipe length.
C_T	Total annual running and operating costs.
d	Annual depreciation.
D	Pipe internal diameter. ft.
D_i	Pipe internal diameter, ins.
E	Efficiency of motor and pump expressed as a fraction.
E_L	Frictional effect of pipe fittings expressed as an equivalent length of pipeline. ft.
f	Fanning friction factor.
F	Frictional resistance. ft.-lbf/lb mass.
F_C	Ratio of installed pipe costs to purchased pipe costs.
g	Investment grant expressed as a fraction.
g_c	Conversion factor. 32.17 ft. - lb.mass/sec.^{2}lbf.
h	Height of storey. ft.
H	Hours of operation per year.
i_m	Minimum return required on capital investment expressed as a fraction.
I	Investment before grant.
J	Frictional loss due to pipe fittings and bends expressed ad equivalent fractional loss in a straight pipe.
K	Power costs. £/KWh.

l	Length of a block of elevated units (perpendicular to piperack). ft.
L	Pipe length. ft.
L_e	Equivalent pipe length, taking into account frictional effects of pipe fittings. ft.
n	Number of storeys above grade.
n_i	Storey on which the ith unit is located.
N	Cost factor.
P	Profit.
q_f	Mass flow. ft^3/sec.
r	Number of units in a block.
R	Gross return.
S	Total sales realisation.
t	Tax on profits expressed as a fraction.
u	Net outward cash flow. £/year. ft. of pipe length.
V	Fluid velocity. ft./sec.
∇_p	Venture Profit.
w	Width of a block of elevated units (parallel to piperack) ft.
w'	Work ft. - lbf./lb mass.
X	Cost factor.
Z	Height above datum level. ft.
μ	Viscosity lb./ft. sec.
μ_c	Viscosity centipoise.
ρ	Density lb/$ft.^3$

REFERENCES

1. Institution of Chemical Engineers, Engineering Practice Committee, Working Party Layout.

2. Perry, "Chemical Engineers Handbook", 4th ed. McGraw - Hill.

3. Peters & Timmerhaus, "Plant Design and Economics for Chemical Engineers", McGraw - Hill, (1968)

4. Kaess D., Chemical Engineering, 77 (12):(1970).

5. Krejcirik M., Computer Aided Design, Autumn (1969)

6. Gunn D.J., Computer Aided Design, Spring (1970).

7. Sachs G., The Chemical Engineer, No. 242:CE304(1970).

8. Muther R. & McPherson.K., Industrial Engineering, February:39(1970).

9. Macy F.C., Petroleum Refiner, 36 (7):162(1957).

10. McGarry J.F., Petroleum Refiner, 37 (10):-09(1958).

11. Landy J.A. Hydrocarbon.Proc. & Petro.Refin. 43 (3): 168(1964).

12. Kern R., Hydrocarb. Hydrocarb.Proc. & Petro.Refin. 45 (10):119(1966).

13. Kern R., Petroleum Refiner., 39 (12):139(1960).

14. Kern R., Petroleum Refiner, 37 (3):-36(1958)

15. KERN R., Petroleum Refiner 39 (2):137(1960)

16. Kern R., Hydrocarbon Proc. & Petrol. Refin.,40(5):195(1961)

17. Maranick J.V., Petroleum Refiner., 37 (9):339(1958).

18. Thompson D., Chemical Engineering., 66 (24):69(1959)

19. Thompson D., Chemical Engineering., 66 928):73(1959).

20. Patterson G.C., Petro/Chem. Engineer.,38(11):38(1966).

21. Patterson G.C., Petro/chem. Engineer.,38(12):48(1966).

22. Gysemans E.E., Chem & Proc.Eng., Feb.:62(1966).

23. Evans M.P., Refining Engineer., 32(3):C-14(1960).

24. House F.F., Chemical Engineering., July 28th, (1969).

25. Muther R., "Systematic Layout Planning", Industrial Education Institute, Boston(Mass.) 1961.

26. Williams W.H., Chem Eng. Prog., 47 (6):227(1951).

27. Gysemans E.E., The Chemical Engineer, Oct:CE252(1968).

28. Rase H.F. & Barrow M.H., "Project Engineering or Process Plants", Wiley (New York) 1957.

29. Landau R., "The Chemical Plant", Reinhold, 1966.

30. Bowen H.J., Chemical Engineering, 61 (8):176(1954).

31. Genereaux R.P., Ind. &Eng. Chem., 29(4):385(1937).

32. Genereaux R.P., Chem.&Met. Eng.,44(5):281(1937).

33. Sarchet B.R. & Colburn A.P., Ind.&Eng.Chem.,32(9): 1249(1940).

34. Bauman H.C., Fundamentals of Cost Engineering in the Chemical Industry", Reinhold, 1964.

35. Bush M.J. & Wells G.L. Brit. Chem. Eng., 16(4/5):325(1971).

36. Bush M.J. & Wells G.L. Brit. Chem. Eng.,16(6):514(1971).