

PUSH-TOWED OCEAN GOING TUG-BARGE OPERATION
IN THE INTEGRAL AND DROP-AND-SWAP MODES:
AN ECONOMIC COMPARISON USING A COMPUTER MODEL

by

JONATHAN DAVID KASKIN

B.S.E.E., University of Pennsylvania
(1971)

S.M., Harvard University
(1977)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREES OF

MASTER OF SCIENCE IN SHIPPING AND SHIPBUILDING MANAGEMENT

and

OCEAN ENGINEER

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

(JUNE 1979)

© M.I.T., 1979

Signature of Author.....
Department of Ocean Engineering, May 1979

Certified by.....
Thesis Supervisor

Accepted by.....
Chairman, Department Committee on Graduate Students

Archives
MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

AUG 8 1979

Report Documentation Page		Form Approved OMB No. 0704-0188
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.		
1. REPORT DATE JUN 1979	2. REPORT TYPE	3. DATES COVERED 00-00-1979 to 00-00-1979
4. TITLE AND SUBTITLE Push-Towed Ocean Going Tug-Barge Operation in the Integral and Drop-and-Swap Modes: An Economic Comparison Using a Computer Model		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, 02139-4307		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited		
13. SUPPLEMENTARY NOTES		

14. ABSTRACT

Currently, all large ocean-going, mechanically linked tug-barge units are being operated in the integral or ship like mode where the tug and barge remain together at all times. This method of operation does not take advantage of the inherent flexibility available to tug-barge units--the ability to separate the propulsive unit (tug) from the cargo unit (barge). This separability can be used to operate the tugs and barges in a drop-and-swap mode where the tug drops off a loaded barge at a port and then swaps it for one that has completed its cargo operations. This method substantially increases tug utilization since the tug is not required to await cargo operations. However, this increased tug utilization is achieved at the expense of the extra barge units which remain in port for cargo operations. The question is whether in drop-and-swap operation the increased tug utilization outweighs the capital costs of the additional barges. To help answer this question, a computer model is developed to compare the total capital and operating cost of providing transport capacity for simple port pair trades with tugs and barges operated in the integral and drop-and-swap mode. From the model's output it can be determined which mode is more economical for a given trade. The model uses exhaustive enumeration to select among all feasible alternatives the number of tugs and barges, the barge size and form, and the tug-barge system speed that provides the desired transport capacity for a given port pair trade at the lowest required freight rate. The required freight rate calculation takes in account the tug, barge storage tank, and terminal facility annual operating and annualized capital costs. Most of the capital and operating cost elements are obtained from the literature. However practically no data are available for two of the most important cost factors, barge capital construction cost and tug-barge fuel expense. Therefore, two computer subprograms a barge design model and a tug-barge powering model, are developed to estimate these costs. The barge design model subprogram uses the ABS offshore barge rules to determine the midship scantling configuration of single-skin tank barges that results in the minimum hull steel weight as a function of barge size and form. The hull weight estimate is used to predict barge cost. Tabular and graphical output from the model is presented for barges of various size and form. In addition, regression equations for the model output are provided. The tug-barge powering subprogram uses full-bodied bulbous-bowless, single-screw tank ship propulsion and resistance data as an approximation to tug-barge hydrodynamic performance. These data are used in conjunction with a propeller design program to determine the horsepower required to propel the tug-barge unit of a given size and form at a given speed. From the horsepower requirements the fuel expenses can be predicted.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT
unclassified

b. ABSTRACT
unclassified

c. THIS PAGE
unclassified

17. LIMITATION OF
ABSTRACT

**Same as
Report (SAR)**

18. NUMBER
OF PAGES

376

19a. NAME OF
RESPONSIBLE PERSON

PUSH-TOWED OCEAN GOING TUG-BARGE OPERATION
IN THE INTEGRAL AND DROP-AND-SWAP MODES:
AN ECONOMIC COMPARISON USING A COMPUTER MODEL

by

JONATHAN DAVID KASKIN

Submitted to the Department of Ocean Engineering
on May 11, 1979, in partial fulfillment of the requirements
for the Degrees of Master of Science in
Shipping and Shipbuilding Management and Ocean Engineer

ABSTRACT

Currently, all large ocean-going, mechanically linked tug-barge units are being operated in the integral or ship like mode where the tug and barge remain together at all times. This method of operation does not take advantage of the inherent flexibility available to tug-barge units--the ability to separate the propulsive unit (tug) from the cargo unit (barge). This separability can be used to operate the tugs and barges in a drop-and-swap mode where the tug drops off a loaded barge at a port and then swaps it for one that has completed its cargo operations. This method substantially increases tug utilization since the tug is not required to await cargo operations. However, this increased tug utilization is achieved at the expense of the extra barge units which remain in port for cargo operations. The question is whether in drop-and-swap operation the increased tug utilization outweighs the capital costs of the additional barges. To help answer this question, a computer model is developed to compare the total capital and operating cost of providing transport capacity for simple port pair trades with tugs and barges operated in the integral and drop-and-swap mode. From the model's output it can be determined which mode is more economical for a given trade.

The model uses exhaustive enumeration to select among all feasible alternatives the number of tugs and barges, the barge size and form, and the tug-barge system speed that provides the desired transport capacity for a given port pair trade at the lowest required freight rate. The required freight rate calculation takes in account the tug, barge, storage tank, and terminal facility annual operating and annualized capital costs. Most of the capital and operating cost elements are obtained from the literature. However, practically no data are available for two of the most important cost factors, barge capital construction cost and tug-barge fuel expense. Therefore, two computer subprograms, a barge design model and a tug-barge powering model, are developed to estimate these costs.

The barge design model subprogram uses the ABS offshore barge rules to determine the midship scantling configuration of single-skin tank barges that results in the minimum hull steel weight as a function of barge size and form. The hull weight estimate is used to predict barge cost. Tabular and graphical output from the model is presented for barges of various size and form. In addition, regression equations for the model output are provided.

The tug-barge powering subprogram uses full-bodied, bulbous-bowless, single-screw tank ship propulsion and resistance data as an approximation to tug-barge hydrodynamic performance. These data are used in conjunction with a propeller design program to determine the horsepower required to propel the tug-barge unit of a given size and form at a given speed. From the horsepower requirements the fuel expenses can be predicted. Since both subprograms take in account barge size and form, the model determines the most economical form for barges of a given size and speed with respect to capital construction and fuel costs.

The output of the model is presented for five base cases and six sensitivity cases. From these outputs, it is determined that the following trades can be more economically operated in the drop-and-swap mode than the integral mode: 1) trades with large annual cargo flow requirements compared to the ton-mile capacity of the maximum allowed size barge, 2) trades with slow loading/discharging rates that result in voyages with long port times, and 3) trades with substantial terminal storage costs.

Complete documentation for the economic model and for the barge design and power subprograms is provided. The documentation includes program listings, flowcharts, and dictionaries of program variables.

Name and Title of Thesis Supervisor: E. G. Frankel, Professor
of Marine Systems.

TABLE OF CONTENTS

ABSTRACT.	2
ACKNOWLEDGMENTS	9
1. INTRODUCTION.	10
1.1 General Remarks	10
1.2 Background Material on Ocean Going Tug-Barge Systems.	11
1.3 Modes of Operation: Drop-and-Swap Versus Integral.	17
1.4 Model Features.	19
2. DROP-AND-SWAP COMPUTER MODEL: FORMULATION AND ASSUMPTIONS.	28
2.1 Model Formulation	28
2.2 Summary of Program Logic.	31
2.3 Drop-and-Swap Program Detailed Logic and Assumptions.	32
2.3.1 Input of port pair trade and system variables	33
2.3.2 Input and modification of semi-fixed parameters	35
2.3.3 Selection of output format.	36
2.3.4 Iteration with respect to port pair trades.	37
2.3.5 Calculation of the number of terminal facilities (and barges for drop-and-swap mode) required at each port	38
2.3.6 Iterations with respect to barge size and form.	38
2.3.7 Calculation of barge hull weight and principal dimensions	39
2.3.8 Iteration with respect to tug-barge speed and calculation of tug IHP and cost	41
2.3.9 Calculation of the number of tugs required.	42
2.3.10 Calculation of annual operating costs.	45
2.3.11 Calculation of system capital costs.	49
2.3.12 Calculation of required freight rates.	50
2.3.13 Storage of system parameters for minimum RFR iterations.	51
3. BARGE DESIGN PROGRAM TO ESTIMATE HULL STEEL WEIGHT.	68
3.1 Introduction.	68
3.2 Logical Structure	71
3.2.1 Input of parameter values	72
3.2.2 Calculation of required section modulus	73
3.2.3 Calculation of the unsupported spans of transverse girders and required number of stanchions	73
3.2.4 Beginning of program iterations with respect to transverse girder and longitudinal frame spacing.	74
3.2.5 Derivation of the inputs for subroutines "smact1" and "smact2".	75

3.2.6	Calculation of hull girder section modulus.	76
3.2.7	Strategy to satisfy minimum hull girder section modulus requirement	77
3.2.8	Calculation of hull weight.	79
3.3	Calculation of Longitudinal Frame Scantlings.	81
3.4	Calculation of Transverse Girder Scantlings	84
3.5	Modification to Allow Hull Weight Calculation of Double-Bottom Tank Barges	86
3.6	Barge Design Model Results.	88
3.6.1	Input and output formats.	88
3.6.2	Barge design model output: Tabular and graphical	89
3.6.3	Regressions through the barge design model outputs. . . .	90
3.6.4	Some comments on barge design model output.	91
3.6.5	Model validation.	92
4.	TUG-BARGE POWERING PROGRAM TO ESTIMATE DHP.	152
4.1	Introduction.	152
4.2	Logical Structure	154
4.2.1	Input of parameter values	154
4.2.2	Calculation of self-propulsion factors.	155
4.2.3	Calculation of residual resistance coefficient.	157
4.2.4	Calculation of inputs for subroutine "prop"	158
5.	DROP-AND-SWAP MODEL RESULTS, OBSERVATIONS, AND SUGGESTED REFINEMENTS	185
5.1	Base Case Results	185
5.1.1	General observations pertaining to all base case runs . .	187
5.1.2	Base case results: Annual cargo flows of 100,000 LT. . .	188
5.1.3	Base case results: Annual cargo flows of 600,000 and 1,000,000 LT.	188
5.1.4	Base case results: Annual cargo flows of 6,000,000 LT. .	189
5.1.5	Base case results: Annual cargo flows of 10,000,000 LT .	190
5.1.6	Conclusions from base case runs	190
5.2	Sensitivity Runs.	193
5.2.1	Sensitivity run: Change in cargo value	194
5.2.2	Sensitivity runs: Changes to shoreside storage costs . .	194
5.2.3	Sensitivity runs: Inclusion of port draft limit.	196
5.3	Suggestions for Model Refinement and Future Research.	198
APPENDICES		
A.	DOCUMENTATION FOR THE DROP-AND-SWAP COMPUTER MODEL.	218
B.	DOCUMENTATION FOR THE BARGE DESIGN MODEL.	286
C.	DOCUMENTATION FOR THE BARGE POWERING MODEL.	338
REFERENCES.		375

LIST OF ILLUSTRATIONS

1.1	Breit/Ingram OGTB Linkage Design.	23
1.2	CATUG OGTB Linkage Design	24
1.3	ARTUBAR OGTB Linkage Design	25
1.4	Port Pair Trades: Integral and Drop-and-Swap Modes .	26
2.1	Model Formulation	53
2.2	Definition of Required Freight Rate	54
2.3	Summary Flowchart for Drop-and-Swap Computer Model. .	55
2.4	Sample Input for the Drop-and-Swap Model.	56
2.5	Base Case Values for Semi-Fixed Parametric Data . . .	57
2.6	Example of Detailed Output From Drop-and-Swap Computer Model	58
2.7	Example of Summary Output From Drop-and-Swap Computer Model	59
2.8	Equations Used in the Calculation of Barge Freeboard.	60
3.1	Scantling Arrangement for Single-Skin Tank Barge. . .	94
3.2	Scantling Arrangement for Double-Bottom Tank Barge. .	95
3.3	Summary Flowchart of Barge Design Subprogram.	96
3.4	Flowchart of Subroutine "smact1".	100
3.5	Flowchart of Subroutine "smact2".	103
3.6	Listing of Barge Design Model Input/Output Program "bargetest1"	107
3.7	Sample Input/Output of the Barge Design Model: Summary Output for the Optimum Barge	108
3.8	Example of Detailed Hull Scantling Output from Barge Design Submodel	109
3.9	Example of Complete Summary Output from Barge Design Subprogram.	110
3.10	Plots of Barge Hull Steel Weight Vs. Barge Scantling Length for $C_b = 0.75$	114
3.11	Plots of Barge Hull Steel Weight Vs. Barge Scantling Length for $C_b = 0.80$	119
3.12	Plots of Barge Hull Steel Weight Vs. Barge Scantling Length for $C_b = 0.85$	124
4.1	Tank-Ship Hull Lines Used in Obtaining Residual Resistance Coefficient and Self-Propulsion Factor Data.	161
4.2	Summary Flow Chart for Powering Subprogram.	162
4.3	Plots of C_r Vs. F_n	164

5.1	Plot of RFR Vs. Terminal L-D Rate. Base Case Run: Annual Cargo Flow of 100,000 LT.	202
5.2	Plot of RFR Vs. Terminal L-D Rate. Base Case Run: Annual Cargo Flow of 600,000 LT.	203
5.3	Plot of RFR Vs. Terminal L-D Rate. Base Case Run: Annual Cargo Flow of 1,000,000 LT.	204
5.4	Plot of RFR Vs. Terminal L-D Rate. Base Case Run: Annual Cargo Flow of 6,000,000 LT.	205
5.5	Plot of RFR Vs. Terminal L-D Rate. Base Case Run: Annual Cargo Flow of 10,000,000 LT	206
5.6	Plot of RFR Vs. Terminal L-D Rate. Sensitivity Run: Annual Cargo Flow of 1,000,000LT with Cargo Value of \$0.	207
5.7	Plot of RFR Vs. Terminal L-D Rate. Sensitivity Run: Annual Cargo Flow of 1,000,000 LT with Storage Facility Capital Cost of \$48/LT.	208
5.8	Plot of RFR Vs. Terminal L-D Rate. Sensitivity Run: Annual Cargo Flow of 6,000,000 LT with Storage Facility Capital Cost of \$48/LT.	209
5.9	Plot of RFR Vs. Terminal L-D Rate. Sensitivity Run: Annual Cargo Flow of 1,000,000 LT with Storage Facility Capital Cost of \$48/LT and with Storage Facility Operating Cost of \$1/LT	210
5.10	Plot of RFR Vs. Terminal L-D Rate. Sensitivity Run: Annual Cargo Flow of 1,000,000 LT with Barge Draft Limited to 38'	211
5.11	Plot of RFR Vs. Terminal L-D Rate. Sensitivity Run: Annual Cargo Flow of 1,000,000 LT with Barge Draft Limited to 38' and with Storage Facility Capital Cost of \$48/LT	212
A.1	Flowchart of the Drop-and-Swap Computer Model	219
A.2	Computer Program Listing of the Drop-and-Swap Computer Model	251
A.3	Dictionary of Variables Used in the Drop-and-Swap Computer Model	269
B.1	Flowchart of the Barge Design Model	287
B.2	Computer Program Listing of the Barge Design Model.	308
B.3	Dictionary of Variables Used in the Barge Design Model	325
C.1	Flowchart of the Barge Powering Model	339
C.2	Computer Program Listing of the Barge Powering Model.	354
C.3	Dictionary of Variables Used in the Barge Powering Model	367

LIST OF TABLES

1.1	U. S. Flag Third Generation OGTTBs	27
2.1	Ranges for System Parameters.	61
2.2	Formulae for Determining Number of Terminal Facilities/Barges at Each Port	62
2.3	Barge Hull Weight and Length Formulae	63
2.4	Barge Outfit Weight	63
2.5	Minimum Freeboard Values.	64
2.6	Tug Capital Costs	65
2.7	Annual Operating Costs for Supplies and Equipment, Maintenance and Repair, and Insurance.	66
2.8	Comparison of Model and MARAD Data for Annual Operating Costs	67
3.1	Values of "fact".	129
3.2	Depth of Longitudinal Bulkhead and Side Girders and Total Number of Stanchions Inserted Between Sides.	131
3.3	Minimum Plate Thickness	132
3.4	Minimum Required Section Moduli for Longitudinal Frames.	133
3.5	Minimum Required Section Moduli for Transverse Girders	134
3.6	Transverse Member Weight per Longitudinal Foot Calculations.	135
3.7	Initial Scantling Values Used in Subroutine "smact2".	136
3.8	Double Bottom Particulars	137
3.9	Single-Skin Tank Barge Hull Weights for $C_b = 0.75$	138
3.10	Single-Skin Tank Barge Hull Weights for $C_b = 0.80$	141
3.11	Single-Skin Tank Barge Hull Weights for $C_b = 0.85$	144
3.12	Double Bottom Tank Barge Hull Weights for $C_b = 0.80$	147
3.13	Regression Equations for Barge Length, Deadweight, and Hull Steel Weight.	150
3.14	Comparisons of Barge Design Model's Hull Steel Weight Estimates with Actual Barge Hull Steel Weights	151
4.1	Self Propulsion Factors	178
4.2	Full-Bodied Single-Screw Residual Resistance Coefficients.	181
5.1	Summary of Inputs Used in Base Case Runs.	213
5.2	Tabular Output Summary for Base Case Runs	214
5.3	Summary of Inputs Used in Sensitivity Case Runs	215
5.4	Tabular Summary for Sensitivity Runs.	216

ACKNOWLEDGEMENTS

I wish to take the opportunity here to express my appreciation to those individuals who have provided me guidance and assistance during the preparation of this thesis. First I would like to thank Mr. Gregory F. Doyle and Prof. J. Gomes de Oliviera for their technical assistance with the barge design model; Prof. M. Triantafyllou for his guidance on the barge powering model and for the use of his propeller design model; and Dr. J. W. Devanney, III, for his technical advice on the drop-and-swap economic model. I would also like to thank Prof. H. S. Marcus for his assistance throughout this research. Additionally, I express my grateful appreciation of the efforts of Prof. E. G. Frankel and J. A. Brogan of the Military Sealift Command for sponsoring the research presented here since without their efforts this thesis could never have been accomplished. Finally, I would like to thank Carol Bishop and my wife Min Jeung Wu for their dedicated efforts in putting the tables and figures in finished form.

CHAPTER 1

INTRODUCTION

1.1 General Remarks

All of the push-towed ocean going tug barge (OGTB) systems in current use operate in a ship-like integral mode where the tug and barge always remain together. This method of operation does not make use of the only long term advantage that these vessels have over ships, their ability to separate the tug propulsive unit from the cargo barge unit. It is in this drop-and-swap mode of operation that OGTBs should flourish. Therefore, it is the purpose of this thesis to investigate for which port pair trades that this feature of separability can be economically advantageous. This investigation is done through the use of a parametric optimization computer model that compares the cost of operating a port pair trade with tugs and barges in either mode.

The details of this model and the associated barge design and tug-barge powering subprograms are presented in the next three chapters. The reader who is not interested in the detailed assumptions used in the model may proceed to model

results presented in Chapter 5 after having read the remainder of this introductory chapter.

1.2 Background Material on Ocean Going Tug-Barge Systems

The idea of a towboat or tugboat pushing a barge or integrated group of barges is not a new one. For several decades, the river towboat operators have been taking advantage of the better control and lower drag resistance provided by push-towing over pull-towing a barge on a hawser. In the sixties coastwise tug and barge operators, seeing the obvious advantages of push towing, began constructing their new large barges (some over 10,000 DWT) with stern notches for the tug to push in. At first, the notches in these first generation ocean-going tug-barges (OGTBs) were rather shallow so that push-tow operations were restricted to the calm waters of bays, harbors, and gentle seas. Later, these notches became deeper and sophisticated cable or chain linking and chafing gear were used so that push-tow operation of these second generation OTGBs could be extended to all but severe seas. But, it was not until the early seventies (1) that the technology

(1) In the early fifties George G. Sharp, Inc., did design the Car-Port/G1, a tug-barge with a rigid wedge type linkage which was the prototype of the Breit/Ingram design now in use. This rather small system (less than 5000 DWT) ran into serious difficulty with USCG manning authorities which eventually demanded that the tug be manned the same as a ship. This forced the system into foreign-flag service. This tug and barge as well as a follow on tug and two barges operated successfully in a drop and swap mode of operation for many years. It may have been this excessive manning requirement that

was developed for the high horsepower diesel engines that could drive large barges at moderate speeds, and for the reliable linkages that would allow the tug and barge to remain linked together in all types of weather and seas.

Three types of these third generation OGTBs are currently in use in U. S. domestic and foreign trades. They are as follows:

1. Breit/Ingram. In this system the tug is wedged tightly into the barge notch walls and floor so that the linkage is rigid. See Figure 1.1 for a sketch and Hukill (1974) for specific details of this system.

2. Catug. In this system a tongue-like extension from the barge stern is wedged tightly between the catamaran hulls and connecting platform of the tug so that the linkage is rigid. See Figure 1.2 for a sketch and Seabulk (1976) for specific details of this system.

3. Artubar. In this system two large transverse pins (one port and starboard) extend from the tug's bow and fit into sockets located in wing wall skegs that make up the barge stern notch. See Figure 1.3 for a sketch and Fletcher (1974) for specific details of this system.

delayed the use of third generation OGTBs until they became large enough so that ship owners, who were not concerned about the ship size manning scale, rather than tug-barge owners began operating them.

A listing with some of the particulars of U. S. flag OGTBs in current or near future operation with the three above designs is given in Table 1.1.

Although only the third generation OGTBs are capable of all-weather push operation, many tug-barge owners have opted to continue construction of the second generation OGTBs. The reason is that these tugs and barges are usually operated on rather short coastal voyages and the fuel that could be saved by all-weather rather than just fair-weather push-tow operation does not appear to outweigh the extra manning (2) and linkage (3) costs and reduced flexibility associated with third generation OGTBs. The reduction in flexibility results from third generation OGTB hull forms being not compatible for use with standard tug and barge units. This is because their hull forms have been specially designed to absorb linkage stresses and provide a combined hull form that is ship shape,

(2) Under current USCG regulations, the third generation OGTBs which are "mechanically" linked systems must have manning scales determined in the same manner as ships. The smallest crew certificated for these systems has been about fourteen. On the other hand, second generation OGTBs which are normally chain or cable linked may be manned with far fewer men if the tugs are under 200 GRT. They run with at most ten on voyages of greater than 600 miles and with about seven on shorter voyages. Additionally, only the officer on watch need be licensed.

(3) Third generation OGTB linkages require special reinforcement of the barge notch and tug bow to absorb the stresses resulting from the linkage. This reinforcement as well as sophisticated coupling and decoupling gear can add approximately a million dollars to the cost of tug and barge. Additionally, the rigid linkage systems require heavier scantlings overall, increasing the hull steel weight and cost even further.

not to mate with other tugs and barges.

However, for tugs and barges that are to be normally used in long distance coastal or transocean voyages, the advantages of third generation OGTBs, lower fuel costs due to reduced resistance and better controllability, seem to outweigh the disadvantages; consequently, more operators are building them for these trades. Since the barges have grown to sizes of up to approximately 50,000 DWT, these third generation OGTBs are being used in trades previously served by ships.

There are basically two reasons why third generation OGTBs have been displacing ships, especially tankers and bulkers, in this deadweight size range--they are cheaper to construct and less costly to man than ships. Although it may not seem reasonable that a tug-barge system consisting of two separate vessels and an expensive linkage should be less expensive to construct than the equivalent sized ship, this has been apparently the case. The reason usually given for this anomaly is that when the barge is built in a specialized barge yard (usually with little plate forming capability for the simple barge forms) and the tug in a specialized tug yard (usually without any expensive drydock or building basin), the higher efficiencies resulting from this specialization will outweigh any additional costs due to the linkage. The reduced manning of third generation OGTBs is the result of extreme automation of the tug's diesel plant and the policy that the

barge remain unmanned at sea so that no crew is needed to perform barge maintenance (shoreside personnel perform this work).

To summarize, although it appears that the third generation OGTB is the natural successor to the second generation OGTB, this is true only with respect to its design, not its use. Third generation OGTBs have been embraced by ship operators as a less costly ship replacement and not by the tug-barge operators as a better tug-barge system. (4) The validity of this conjecture is confirmed by the fact that all of the third generation OGTBs have been operated in a ship like mode--that is, the tug and barge always remain together except for overhauls.

It should be noted that the use of third generation OGTBs in place of ships does not seem viable for the long term. If there are significant advantages in building the propulsive and cargo units in separate specialized yards, the same can be done for a ship, as was demonstrated by the construction of the Great Lakes vessel Stewart A. Cort. (5) As

(4) Again, this is primarily due to the manning penalty that results under USCG regulations for the "mechanically" linked third generation OGTBs. In fact, the tug-barge operators have been spending their time and energy trying to develop an improved second generation OGTB that will practically achieve the performance of the third generation OGTB with only cable or chain linkages since these systems do not have any manning penalty.

(5) The cargo component was built in a new modular construction yard which was ideal for the cargo unit specialization. The propulsive bow-stern component, however, was built at a

for the lower manning of OGTBs, there is no technical or regulatory reason why ship crews can not be reduced to the same size as third generation OGTBs if they are as fully automated and follow the same maintenance policy. For example, the gas turbine 35,000 DWT Chevron tankers were certificated by the Coast Guard to be manned with fourteen men, the same as most third generation OGTBs. These ships are manned with about twenty men, primarily due to union rules. But, it is expected that these rules will be relaxed when the unions fully realize that they could be forcing operators to use second generation tug-barge systems which are normally manned with fewer non union personnel. So, in the future, a third generation OGTB should not be any cheaper to construct or to run than an equivalent sized ship that is built and operated under the same conditions.

However, there is one inherent feature that third generation OGTBs have in common with all tug-barge systems that is not being taken advantage of by their current users and that cannot be provided by ships. It is the ability to separate the propulsive unit from the cargo carrying unit. This flexibility is commonly utilized in other transport modes, i.e., truck tractor with trailer and train locomotive with rail cars, to increase system efficiency through increased utiliza-

large shipyard, probably to ensure that all construction would remain within company owned facilities. So, in this case only partial specialization of facilities was achieved.

tion of expensive propulsive units and through the storage capability of the detached cargo units. It may be that current third generation OGTBs users, being previously experienced only in ship operations, have just overlooked the potential benefits of the separability of their assets, or it may be that it is not economically profitable to utilize this capability in the trades in which they operate.

It is the purpose of the economic model developed in the following chapters to investigate under just what conditions the separability feature of third generation OGTBs should be utilized. If the model indicates that there are few trades where this feature may be used, then it would be expected that ships would eventually push them all-but-out of the shipping scene. However, if the model indicates that there are many trades which could take advantage this feature, then it might be expected that more ships and possibly other transport modes could be replaced by OGTBs in the future. (6)

1.3 Modes of Operation: Drop and Swap versus Integral

As it is the separability feature of OGTBs that make them more versatile than ships, more mention should be made on how this feature can be profitably used. The major benefit is the same as that enjoyed by tractor trailers over single unit

(6) Except for long distance coastal or transocean trades, it would be expected that third generation OGTBs would not displace first or second generation OGTBs until their relative linkage and manning costs can be reduced significantly.

trucks. That is, the propulsive unit (tractor or tug) can be detached from the cargo unit (trailer or barge) while the cargo unit is used for loading, discharging, or storage. It then can be used for transporting another cargo unit that is available for movement. This method of operation, the drop-and-swap mode, obviously increases the utilization of the costly propulsive unit as compared to the integral mode of operation in which the propulsive unit always remains with the cargo unit. However, as can be seen in the simple port pair system shown in Figure 1.4, the drop-and-swap mode will require at least two more barges than tugs in a balanced trade or at least one more barge than tugs in an unbalanced trade (where the tug remains with the barge in one port). Certainly, the drop-and-swap mode would be of most benefit in those trades in which loading/discharging times make up a significant part of the voyage time. Here, then is the most potential in increasing tug utilization, especially in multi-tug fleets which can often be scheduled so that a tug arrives with a barge for discharge just at the time when a barge in port has completed its cargo operations and is available for transport. In this case tug utilization can approach 100%. Also, in trades with long port times, the barges remaining in port in the drop-and-swap mode of operation are used essentially as floating warehouses, replacing shoreside assets. Since port time is a function of the barge cargo capacity and both terminals' loading and discharging rates, and since sea

time is a function of port separation distance and OGTB speed, these four parameters are critical in determining whether the drop-and-swap mode should be used over the integral mode. Because the cost relationships that are functions of these parameters (e.g., fuel cost is a function of the tug-barge size and form, OGTB speed, and port separation distance) are rather complicated, it is not intuitively obvious when one mode is superior to another. A systematic analysis, such as that performed by a computer, is required to determine where the tradeoff point is for the modes.

1.4 Model Features

The economic model described in detail in the next chapter is used to determine for which trades OGTBs should be used in the drop-and-swap versus integral mode. It is in these trades that take advantage of OGTB separability that these systems should flourish, not in those in which they are used as pseudo ships.

The trades analyzed in the model are of the simple port pair type shown in Figure 1.4. This case was chosen since it is the simplest and is appropriate for many of the bulk trades (repetitive voyages from the same loading port to the same discharging port). This port pair trade can be defined essentially by three sets of parameters: (1) port separation distance, (2) terminal loading and discharging rates, and (3) annual cargo flows between ports.

Given the specifics of the trade, the model then determines the barge size (and form) and OGTB speed that will yield the minimum required freight rate (rfr) for both the integral and drop-and-swap modes (balanced and unbalanced). The rfr is defined as the freight rate that should be charged for a unit of cargo that will recover all capital and operating costs on a present-valued discounted cash flow basis (taxes and depreciation ignored). Although specific details on how these costs are obtained is given in the next three chapters, some mention about the general assumptions used in obtaining them is given here so that the reader can skip to Chapter 5 if he wishes to immediately read the results of the model without the details of its procedures. Specifically, with respect to capital and operating costs the following was assumed:

1. Barge capital costs were assumed to be a direct function of barge hull weight with the addition of outfit cost determined via a regression equation found in Sharp (1975). The hull weight as a function of barge size and form was obtained via regression equations developed from output of the barge design program presented in Chapter 3. This program is applicable to single-skin tank barges joined by a semi-rigid linkage to the tug. These types of barges were used since they are the simplest to model and since they are most prevalent of the large OGTBs in use. The semi-rigid rather than rigid linkage was used since it results in less stringent scantling requirements under ABS rules, and in less barge cost. This

rather complicated approach was taken since no reliable barge cost estimate could be obtained from the little available data on OGTBs. In addition, the variation in hull weight as a function of barge form parameters provided by the subprogram is needed for weighing the capital versus operating cost aspects of a barge form.

2. Tug capital costs was determined via a regression equation found in Sharp (1975) and adjusted to conform with prices reported in recent trade literature and government publications. Since many tugs have been built recently, this approach seemed reasonable. Some adjustment is made for the cost of the linkage, which is not extraordinary for semi-rigid linkages.

3. Storage capital costs were calculated for oil storage tanks. The costs were based on recent cost figures provided by a major oil company.

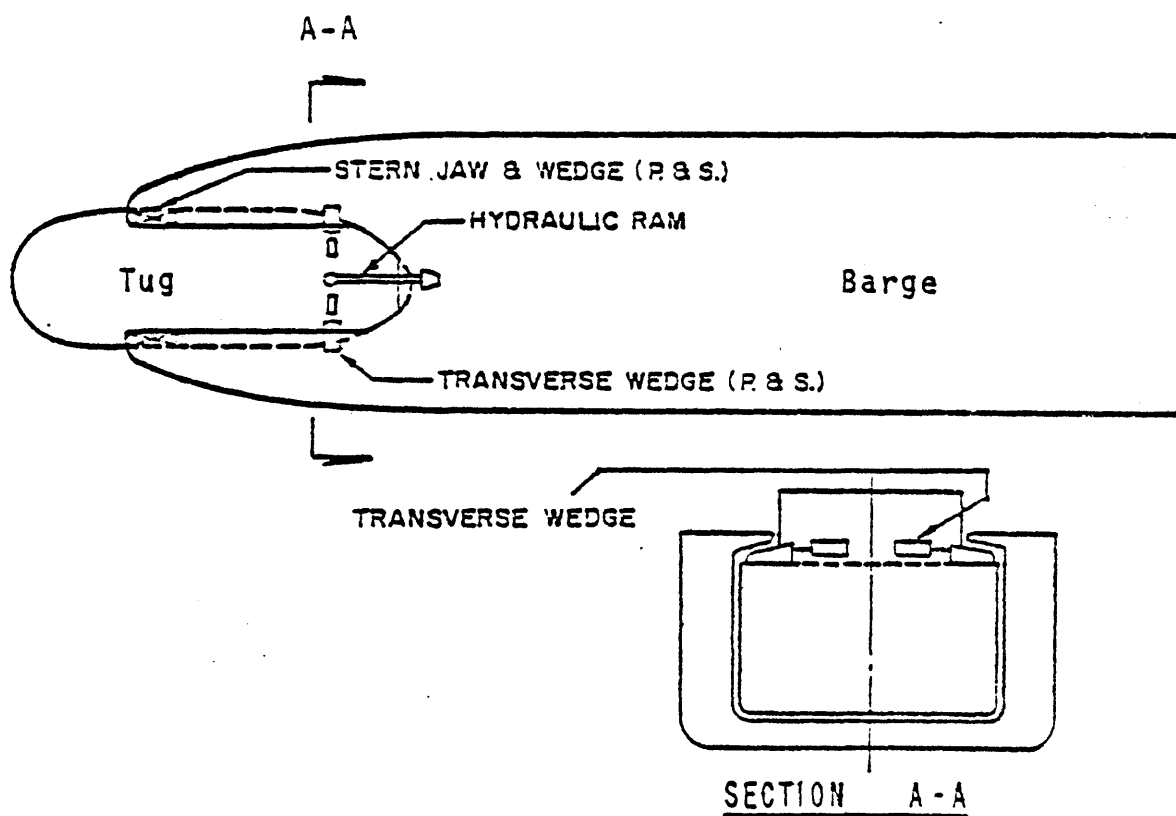
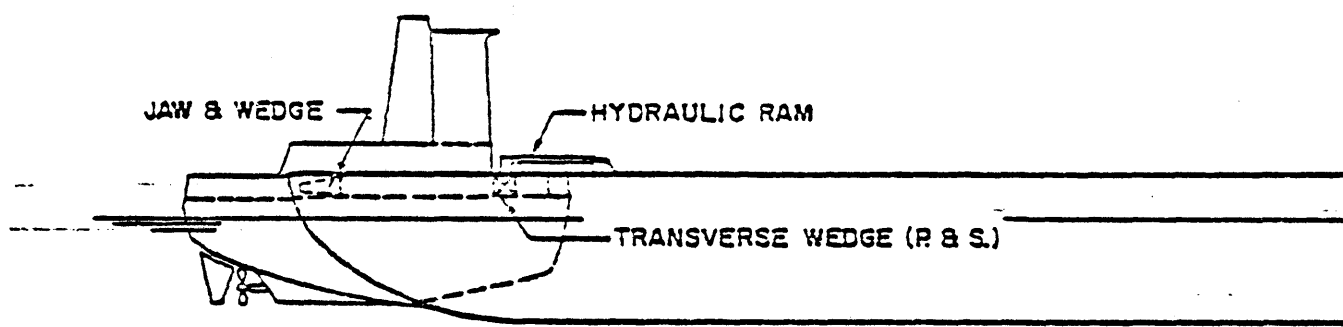
4. Tug fuel costs were determined as a function of tug-barge resistance and voyage duration. Barge resistance, a function of barge form and speed, was determined with the use of full-bodied, bulbous bowless, single-screw tank barge resistance data found in Tsuchida (1969). This was the only series resistance data that could be found to approximate OGTB hull forms. An additional 10% resistance was added to account for linkage interferences to conform with the estimates found in Robinson (1976).

5. Other operating costs were calculated by using the

equations found in Sharp (1975) and then inflating them to yield a current estimate--for 1 January 1979.

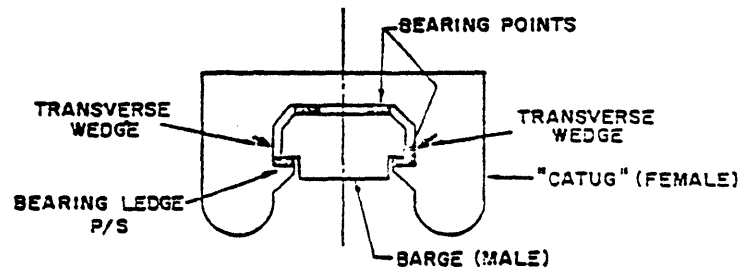
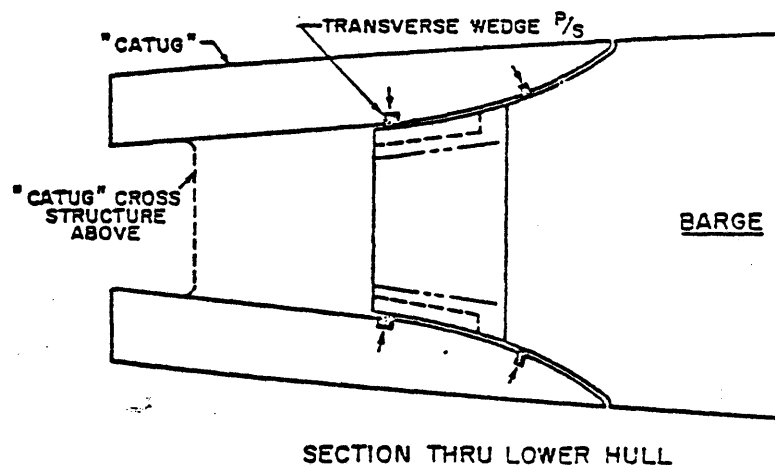
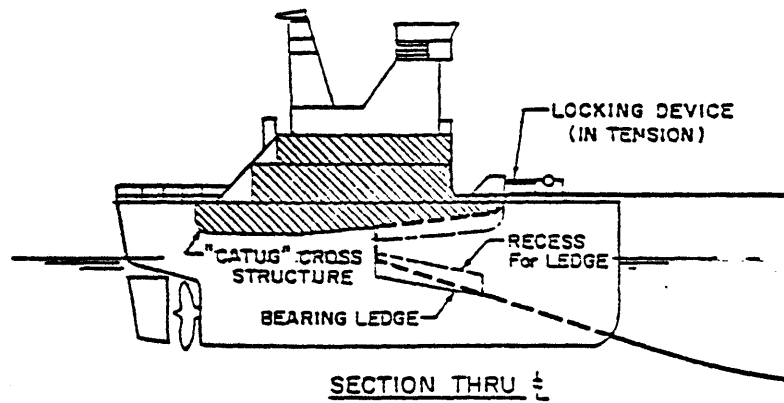
Although the economic model has been specifically developed for the tank barge case, it should still be indicative of costs for other bulk trades. Barge outfit cost would probably be the only major change for other trades. Thus, any results obtained from the tank barge model, even for trades with very long port times which are not usual for oil barge trades, should not be in great error.

In the following chapters frequent mention will be made to specific computer program variables which will be enclosed in quotation marks. Although their definition will normally be understood by context, detailed definitions of these variables are provided in the appendices.



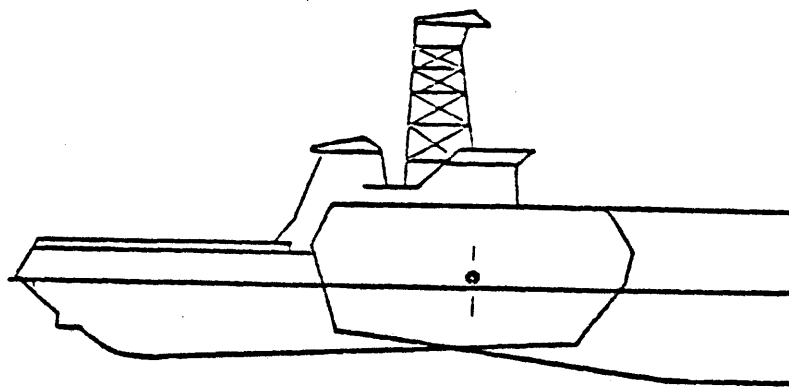
SOURCE: Waller (1972)

FIGURE 1.1
BREIT/INGRAM OGTB LINKAGE DESIGN



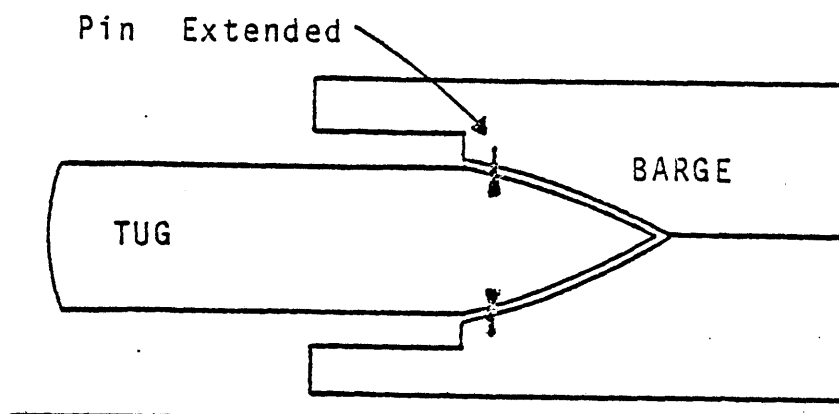
SOURCE: Waller (1972)

FIGURE 1.2
CATUG OGTB LINKAGE DESIGN



Deep Notch-Pinned "Artubar"

From the tug, port and starboard pins are extended toward the barge. The bullet shaped pins aid alignment. In the extended position, the pins fit into lubricated sockets within the barge.



SOURCE: Waller (1972)

FIGURE 1.3
ARTUBAR OGTB LINKAGE DESIGN

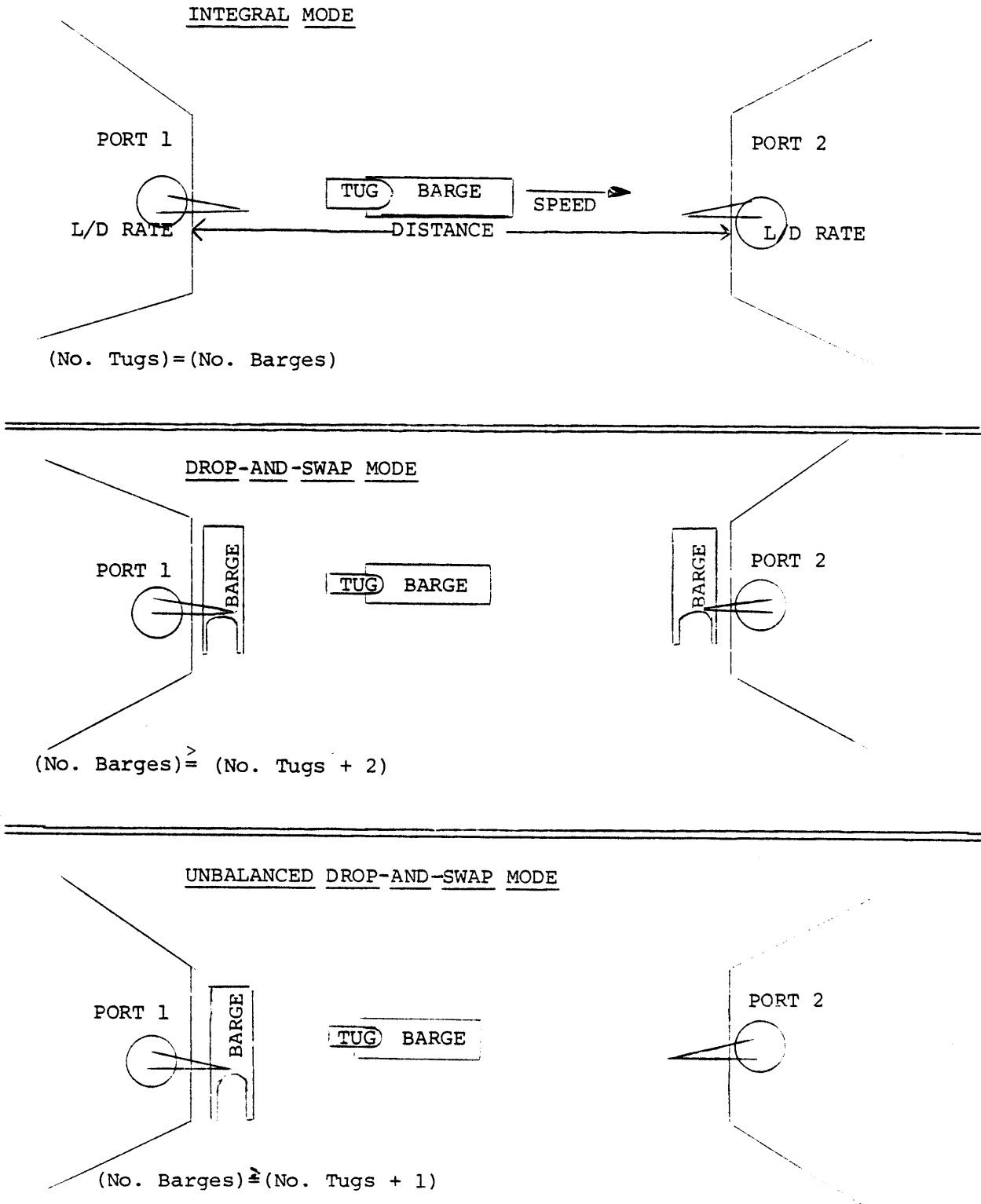


FIGURE 1.4
PORT PAIR TRADES: INTEGRAL AND DROP-AND-SWAP MODES

TABLE 1.1
U.S. FLAG THIRD GENERATION OGTB'S

Linkage Design	Barge Type/Service	Name: Tug Barge	Dimensions (LxBxD): Tug Barge	Tug-Barge: Draft Length	Tug-Barge: HP/Design Speed DWT	When Built: Tug Barge	Where Built: Tug Barge	Owner/Operator
Breit/Ingram	Taconite Dry Bulk/ Great Lakes	Presque Isle Presque Isle	140.33'x54.0'x31.25' 974.5'x104.58'x46.50'	29' 1000'	14,840/16mph 52,000	12/73 "	Halter Marine Erie Marine	Crocker National Bank/ U.S. Steel
Breit/Ingram	Clean Products Tank/Ocean	Martha R. Ingram IOS 3301	145.84'x46.0'x30.25' 584.5'x87.0'x46.33'	37'5" 620'	11,128/14.1 Kts 36,500	7/71 3/71	Southern Shipbuilding Alabama Drydock	Ingram Corp.
Breit/Ingram	Clean Products Tank/Ocean	Carole G. Ingram IOS 3302	145.84'x46.0'x30.25' 584.50'x87.0'x46.33'	37'5" 620'	11,128/14.0 Kts 37,500	3/72 "	Southern Shipbuilding Livingston Shipbuilding	Ingram Corp
Breit/Ingram	Rice Phosphate Dry-Bulk/Ocean	Valerie F Valerie F	150.67'x54.0'x34.0' 620.0'x85.0'x45.0'	30'8" 656'	16,000/15.5 Kts 25,000	12/76 "	Southern Shipbuilding Maryland Shipbuilding	
Breit/Ingram	Fertilizer Dry-Bulk/Ocean	Jamie A. Baxter CF-1	125.0'x45.0'x27.75' 500.0'x75.25'x46.5'	32' 600'	7200 /12.5 Kts 22,500	6/76 12/77	Peterson Builders Avonsdale Shipyard	C.F. Industries
CATUG	Oil Tank/ Ocean	Seabulk Challenger SSC-3901	116.08'x90.44'x38.42' 581.0'x95.0'x46.0'	37' 629'	14,000/15.5 Kts 35,000	1/75 "	Galveston Shipbuilding (Kelso Marine)	Hvide Shipping/ Shell Oil
CATUG	Chemical Tank/Ocean	Seabulk Magnachem SSC-3902	116.08'x90.44'x38.42' 582.17'x95.0'x52.0'	40'1" 615'	14,000/15.5 Kts 40,000	2/77 "	Galveston Shipbuilding (Kelso Marine)	Hvide Shipping/ Diamond Shamrock
CATUG	Superphosphoric Acid/Ocean	Two or Three Systems to be Named	126'5"x90'4"x39' 626'6"x99 x50'	36' 677'10"	18,200 /15.5 Kts 41,250	80+	Avondale Shipyard	Occidental Oil
CATUG	Oil Tank/ Ocean	Two Systems to be Named	127'7"x90'4"x39' 645'x95.0'x61.6"	40'6" 699'4"	18,200/15.5 Kts 47,075	80-81	Halter Marine Bethlehem Steel	Amerada Hess
ARTUGBAR (2-units)	Ro-Ro/ Ocean	JJ Oberdorf Other Names Unknown	140'x40'x 568'x85'x41'6"	19' 605'6"	7,500 6450 (554 vehicles)	79-80	Marinette Marine Seatrain Shipbuilding	Coordinated Caribbean Transport

CHAPTER 2

DROP-AND-SWAP COMPUTER MODEL: FORMULATION AND ASSUMPTIONS

2.1 Model Formulation

The drop-and-swap computer model has been developed to make an economic comparison of the operation of push-towed ocean going tug-barge combinations in the drop-and-swap versus integral modes. As shown in the model formulation summary in Figure 2.1, the model makes this comparison by determining for both modes the number of tugs and barges and the barge speed, size (DWT), and form (block coefficient-- C_b , length-breadth ratio-- L/B , breadth-draft ratio-- B/T) that results in the minimum required freight rate for cargo transported on a port pair trade. (1) This is subject to the conditions that the system has sufficient ton-mile capacity to carry the annual cargo flows and sufficient number of terminal facilities at each port to handle the annual port throughput.

The objective function, the required freight rate (rfr), is equal to the system capital costs (tugs, barges, terminal

(1) See Figure 1.4 and Section 1.4 for the definition of a port pair trade.

and storage facilities) divided by a present value factor (1) plus system annual operating costs (fuel costs, crewing costs, storage and terminal costs, etc.) all divided by the annual cargo flows. As shown in Figure 2.2, these capital and operating costs are nonlinear functions of the port pair trade parameters--port separation distance, terminal facility loading/discharging rates, and annual cargo flows--as well as the five continuous and three integral system variables--Barge DWT, Speed, Cb, L/B, B/T; Number of Tugs, Number of Barges--Port 1, and Number of Barges--Port 2.

Since the last three variables must be integral, this makes the model's form a mixed-integer non-linear program. Programs of this type are not amenable for solution by optimization techniques. If they can be solved at all, it is usually by some sophisticated specialized technique that transforms the model to one that is easier to solve but with many more variables. Rather than taking this approach, I decided to use the brute force method of exhaustive enumeration because it is the simplest to program and because it provides good results at reasonable cost when the ranges of the variables to be considered are chosen judiciously. With this approach, I calculated for each port pair trade and for both modes of operation the required freight rate for all possible

(1) The present value factor apportions the capital costs on an annual basis. It is a function of the capital's pre-tax discount rate or rate of return and the economic life of the system.

combinations of the discretized values (1) of the five continuous system variables. Given a specific combination of these variables, the capacity and continuity constraints determine the values of the three integer variables.

Since I used discretized values for the continuous variables, the minimum required freight rate found will most likely not be the true minimum. However, since the objective function was found to very flat near the optimum, the rfr found will be close to the true minimum even if rather large increments for the variables are used. And, of course, a more accurate estimate of the true minimum rfr can be obtained by using smaller increments, although the cost will probably not warrant the additional accuracy achieved.

Prior to proceeding to a discussion of the computer model's logic and assumptions in the next section, mention should be made of the parametric ranges of the system variables for which the model will produce valid results. These ranges, which are governed by the valid ranges of the formulae used in the model, are provided in summary form in Table 2.1.

(1) To limit the number of combinations to a finite and reasonable number, it was necessary to discretize these continuous variables by dividing their parametric ranges into equally spaced increments. The number of increments can vary from two to fifteen depending on the sensitivity of objective function to changes in the variable.

2.2 Summary of Program Logic

A summary of the logic of the drop-and-swap computer program is shown in flow chart form in Figure 2.3. A brief discussion of the overall logic in this section will be followed by a detailed discussion of each step in the next section.

The program begins by asking the user to specify the port pair trades to be considered as well as the parametric ranges and increments of the five continuous system variables. It then asks the user to specify whether he wishes to change any of the semi-fixed parameters that are used in the cost calculations. Following this, it asks the user to specify what form he wants the output. The program then begins the cost calculations for a specific port pair trade iteration defined by port separation distance and terminal loading/discharging rates. Given these port pair trade parameters, the model determines the number of terminal facilities with the specified loading/discharging rate (and the number of barges for the drop-and-swap mode) that are required at each port.

Then the program selects the next iterative values for the four barge size and form system variables--Barge DWT, Cb, L/B, and B/T. Given these values, it calculates the barge length, hull steel weight, cost, and the tug and barge principal dimensions. The program then checks to see if these system variables are feasible in that they fit within the inter-

polation table ranges for the tug-barge residual resistance coefficients. If not, the program skips to the next iteration of the four barge size and form variables. If so, the program selects the next iterative value of tug-barge speed and calculates the horsepower required to be delivered to the propeller of the tug, the tug-barge resistance, and finally the horsepower of the engine required to be installed onboard the tug. Given these values, the cost of the tug is calculated and then the number of tugs required to provide sufficient transport capacity for both drop-and-swap and integral modes. Then the operating costs and the total capital costs are determined for both modes. From these the rfr can be determined. If the rfr is less than that calculated during previous iterations of the five continuous system variables, it is saved; otherwise, it is ignored. After all iterations of the variables are examined, the minimum rfr found is stored for that port pair trade. After all of the port pair trades are examined the program can present the results in various graphical forms.

2.3 Drop-and-Swap Program Detailed Logic and Assumptions

The detailed logic and assumptions used in the drop-and-swap program are shown in flowchart form in Appendix A. It will be useful to refer to these flowcharts in the discussions to follow.

2.3.1 Input of Port Pair Trade and System Variables (Refer to Figure 2.4)

After typing the execution command "drop_and_swap", the program reads from tape into memory the values of the loadline factor, residual resistance coefficient, self-propulsion factor, and propeller design coefficient arrays. The program then asks the user to "Input via list format the following parameters:". (To input via list format, the user types in the value of the requested parameter followed by a comma.) The first request is a question on the desired form of the terminal facility loading/discharging rates: "Do you wish to specify individual L-D rates?". If the user answers negatively ("no", "n", or "0"), then the program requests values for "minrate, maxrate, delrate". This is a request for a range of loading/discharging rates in tons per day per terminal facility to be investigated, from "minrate" to "maxrate" in "delrate" increments. It is assumed that the loading and unloading rates at both terminals will be the same ("rload1"="runload1"="rload2"="runload2"). Also, if "delrate" is set to zero, then a "delrate" equal to 1000 is assumed. If, on the other hand, the user answers affirmatively ("yes", "y", or "1") to the loading/discharging rate question, then the program requests values for "rload1, runload1, rload2, runload2". This is a request for a specific set of terminal facility loading and discharging rates for each port. In

either case, any set of loading/discharging rates may be specified.

Now the program continues with a request for the values of "mindist, maxdist, deldist". This is a request for a range of port separation distances in nautical miles to be considered from "mindist" to "maxdist" by "deldist" increments. If "deldist" of zero is inputted, then a "deldist" of 1000 nautical miles is assumed. Any set of port separation distances may be considered.

Next the program requests values for "minspeed, maxspeed, delspeed". This is a request for a range of tug-barge speeds in knots to be considered from "minspeed" to "maxspeed" by "delspeed" increments. If "delspeed" of zero is inputted, then a "delspeed" of 1.0 knot is assumed. Any set of speeds can be considered. However, a minimum speed of six knots and a maximum of twelve or thirteen knots will usually cover the optimum speed range and will not exceed the boundary restrictions for Froude number and IHP shown in Table 2.1.

Next the program requests values of "mindwt, maxdwt, deldwt". This is a request for a range of barge cargo deadweights in long tons (LT) to be considered from "mindwt" to "maxdwt" by "deldwt" increments. If "deldwt" of zero is inputted, then a "deldwt" of 5000 LT is assumed. Any set of deadweights from 5000 to 80,000 LT can be used.

Next the program requests values for "aflowave1, aflowave2". This is a request for the annual average cargo

flows in long tons from Port 1 to Port 2 and from Port 2 to Port 1, respectively. Any pair of values can be specified, except that "aflowave1" must be greater than or equal to "aflowave2". For example, if a one way trade is desired, then "aflowave2" is set to zero.

Next the program requests values for the three barge form variables, "mincb, maxcb, delcb", "minlb, maxlb, dellb", and "minbt, maxbt, delbt". These three requests are for the ranges of the barge block coefficient (C_b), tug-barge length-breadth ratio (L/B), and barge breadth-draft ratio (B/T) to be considered from "mincb", "minlb", "minbt" to "maxcb", "maxlb", "maxbt" by "delcb", "dellb", "delbt" increments, respectively. If "delcb", "dellb", or "delbt" of zero is inputted, then a value of 0.1, 0.2, or 0.1 is assumed, respectively. The valid ranges for these form parameters are given in Table 2.1. If the user does not input values within these ranges (including the reduced confidence ranges) then the program will ask the user to specify a new set of form parameter ranges. After the program has accepted the form parameter ranges, input of the system variable and port pair trade ranges to be considered has been completed.

2.3.2 Input and Modification of Semi-Fixed Parameters (Refer to Figure 2.4)

Now the program outputs the statement, "Input changes to semifixed data via get data format". This is a request to the user to make any modifications to the semi-fixed parametric

data that is used in the required freight rate calculations. The base case values of these parameters that are read from tape into memory are shown in Figure 2.5. Definitions of these parameters can be found in Appendix A. The user may modify the value of any of these semi-fixed parameters by simply typing the parameter name followed by an equal sign and then followed by the desired parameter value. Each parameter that is modified should be separated by a comma and the final one should be followed by a semicolon.

2.3.3 Selection of Output Format

Next the program asks the user to specify what type of format he desires for the program output. Specifically, it asks, "Do you want printed output?:". If the user answers in the affirmative, the program responds, "Do you want detailed output?:". If the user answers this question in the affirmative, the program will print out a line of output for every single iteration with respect to port pair trade and system variables. An example of this output for the input case shown in Figure 2.4 is shown in Figure 2.6. If the user answers negatively to the detailed output question, the program will print the system data associated with the iteration that resulted in the minimum required freight rate for both modes (drop-and-swap, then integral) for each port pair trade considered. An example of this output for the input case shown in Figure 2.4 is shown in Figure 2.7.

On the other hand, if the user answers negatively to the question about printed output, or after the program has completed printing output, the program will ask, "Do you wish graphic output?". If the user answers negatively, the program will start from the beginning, asking for a new set of inputs. Otherwise, if the user answers affirmatively, the program will ask a series of questions concerning the form of the graphical output. Samples of graphical output are found in Chapter 5.

2.3.4 Iterations With Respect to Port Pair Trades

At this point the program begins the first of its iterative loops. It now selects the next incremental value for the iterative variable "distance" within the range "mindist" to "maxdist". This specifies the port separation distance of the trade under consideration. Next, the program will either use the values of "rload1", "rload2", "runload1", and "runload2" specified at the beginning of the program or will set these variables to the value of the iterative variable "rate" within the range "minrate" to "maxrate". This specifies the terminal facility loading/discharging rates of the port pair trade under consideration.

After this, the program zeroes the arrays ("best1" and "best2") that store the characteristics of the tug-barge systems that have the lowest required freight rate for the given port pair trade. Then it calculates the monthly average cargo flows ("mflowave1" and "mflowave2") which are the annual

flows apportioned on a monthly basis taking in account that the barge is available only "bargeopdays" of the year for service.

2.3.5 Calculation of the Number of Terminal Facilities (and Barges for Drop-and-Swap Mode) Required at Each Port

Now that the port pair trade characteristics have been defined, the program determines the number of terminal facilities with the specified loading and discharging rates that must be located at each port to handle the annual cargo flows. This is also the number of barges that must be handled simultaneously at each port for the drop-and-swap mode of operation. This number is determined by dividing the monthly cargo flows ("mflowave1" and "mflowave2") by the monthly terminal facility throughput capacity ($30.5 \times L-D$ rate). The specific formulae used are shown in Table 2.2. It should be mentioned that seven days a week operations were assumed.

2.3.6 Iterations With Respect to Barge Size and Form

Next the program begins the iterative loops with respect to barge size and form. First it selects the next incremental values for the iterative variable "dwt" within the range "mindwt" to "maxdwt". This specifies the barge cargo deadweight to be used in the calculations to follow. Then the program selects the next incremental values for the iterative form variables "cb", "lb", and "bt" within the ranges "mincb" to "maxcb", "minlb" to "maxlb", and "minbt" to "maxbt",

respectively. This specifies the tug-barge C_b , L/B , and B/T to be used in the calculations to follow.

2.3.7 Calculation of Barge Hull Weight and Principal Dimensions

Given the values of the barge DWT and form variables specified in the iterative loops discussed above, the program calculates the barge length (1) and hull weight values via quadratic interpolation with respect to block coefficient from the formulae provided in Table 2.3. These formulae were derived from the output of the single-skin tank barge program discussed in the following chapter.

Next the program determines the barge breadth by dividing the tug-barge length ("litb") by the length-breadth ratio ("lb"). Similarly, the barge draft is determined by dividing the barge breadth by the breadth-draft ratio ("bt"). It should be noted that the tug-barge length is assumed to be equal to the barge length plus seven-tenths the tug length. This assumption closely approximates the lengths of the large articulated push-towed ocean going tug-barge systems now in operation.

Now the program begins a short iterative loop by using a formula found in Sharp (1975) to estimate the barge outfit

(1) Barge length used in these formulae refers to the distance at the waterline from two-thirds of the barge notch length forward of the stern to the barge stem. This is in accordance with ABS rules pertaining to articulated tug-barge systems presented in MARAD (1979).

weight as a function of barge length. Sample values from this formula are shown in Table 2.4. The value of outfit weight summed with the cargo deadweight and with the barge hull steel weight is used to estimate the barge displacement. The barge displacement is used, in turn, to obtain an improved estimate of the barge length. This procedure is then repeated iteratively until the barge length as well as barge displacement and outfit weight converge to unchanging values. Given the value of outfit weight, the program calculates an estimate of the barge cost by summing the product of outfit weight times a cost per ton outfit factor (1) with the product of hull steel weight times a cost per ton hull steel factor. (2)

If the barge length is found to exceed 750 feet, which is beyond the feasible range of the model, then the program skips to the next iterative value for "cb". Otherwise, the program calculates the barge freeboard using the rules specified in IMCO (1966). The specific formulae and table values used are shown in Table 2.5 and Figure 2.8. It should be noted that in the freeboard calculations it was assumed that the barge had no sheer and that the barge is unmanned so that a 25% reduction in freeboard is allowed. After the barge

(1) This factor is assumed to be \$12,820 per long ton outfit. This is based on the value found in (Sharp, 1975) inflated by 30% to bring it a January 1979 value.

(2) This factor is assumed to be \$1100 per long ton hull steel. This is based on 40 man-hours per ton at \$15 per man hour (including overhead) plus \$500 per ton material cost.

freeboard is calculated, the barge depth is determined. At this point the program checks to see if the tug-barge unit's dimensions exceed any of the length, breadth, or draft limitations ("maxl", "maxb", "maxt1", or "maxt2") that may have been specified in the semi-fixed parametric data for the port pair trades under consideration. If any of these limitations are exceeded, the program skips to the next "bt" iteration. Otherwise, it checks to see if the form parameters "cb" and "lb" are within the table interpolation ranges specified in Table 2.1. If they are not, the program skips to the next "lb" iteration; otherwise, it continues as described in the following section.

2.3.8 Iteration With Respect to Tug-Barge Speed and Calculation of Tug IHP and Cost

At this point the program begins the last of the iterative loops which is with respect to tug-barge speed. It selects the next incremental value for the iterative variable "speed" within the range "minspeed" to "maxspeed". Then the value of "speed" and the tug-barge principal dimensions ("lbarge", "bbarge", "tbarge", and "cb") are fed into the Subprogram "power". This program, described in detail in Chapter 4, returns the value of the horsepower required to be delivered by the propeller ("dhp") to propel the tug-barge system through the water at the specified speed. (1) From

(1) This program also calculates "ehp", the power required to propel the tug-barge system through the water, from which the

this value the shaft horsepower can be determined. It should be noted that a tug shaft efficiency of 98%, an appendage drag of 5%, and a linkage drag of 10% were assumed. (1)

Given the value of the service margin (2) for the tug, the horsepower of the engines required to be installed onboard the tug can be determined. From this the cost of the tug can be estimated using the formulae found in Sharp (1975), inflated 30% to bring them up to a January 1979 level. Sample values from these formulae are presented in Table 2.6.

2.3.9 Calculation of the Number of Tugs Required

Now the program begins the calculations to determine the number of tugs required to provide sufficient movement capacity for the required annual cargo flows. It does this for the drop-and-swap modes, balanced and unbalanced, (3) and then the

still water hull resistance can be determined. It also provides values for the self-propulsion factors ("wa", "th", and "hr") and open water propeller efficiency "propef".

(1) The assumptions for appendage and linkage drag are based on conversations with articulated tug-barge designers. They seem optimistic when compared to the results presented in Robinson (1966). However, this study used fairly crude prototype linkage forms and so probably overestimated the drag for the modern linkages which are well faired.

(2) In the model's base case the service margin, the additional fraction of "ehp" required to ensure that the service speed is achieved in most seas, is assumed to be 0.20.

(3) The unbalanced drop-and-swap mode ("dsopt" = 1) is the case where it is assumed that the tug will remain with the barge in the port with the shortest time spent for cargo operations. This would be appropriate for one-way trades with short loading and long discharging times. In this case the

integral mode.

To determine the number of tugs required in the drop-and-swap modes, the program first calculates the time required for cargo operations in both ports ("tport1" and "tport2"). For "tport1", this is the barge cargo deadweight divided by the terminal facility loading rate plus the cargo deadweight--weighted by a balance factor if cargo flows are not equal in both directions--divided by the terminal facility discharging rate. A similar formula pertains for "tport2". Then the program calculates the sea voyage time for a round trip. This is twice the distance divided by the system speed plus linking and unlinking times if appropriate plus any other expected port delays. (1) For the drop-and-swap modes, these values are fed into an iterative routine that calculates the total tug voyage time ("ttript") which includes seetime plus any time that tug is required to wait for completion of cargo operations ("twait1" for Port 1 and "twait2" for Port 2). The routine also calculates the minimum number of tugs ("mintug") required for the trades.

The logic of the iterative routine is rather simple.

waiting time for the port that the tug remains with the barge is equal to the cargo operations time, i.e., "twait1" = "tport1". The program selects the drop-and-swap mode that results in the lower rfr to be stored and printed.

(1) In the model's base case the port delay and tug-barge linking/unlinking times were estimated to be four hours. The port delay time takes in account the expected time for docking and undocking as well as time awaiting berth for the barge.

After the program assumes initial values for "mintug", "twait1" and "twait2"; it calculates new values for "twait1" and "twait2" based on the number of barges stationed at each port and the currently assumed value for "mintug". This is done by assuming the tugs are on equally spaced time schedules. Given the values of "twait1" and "twait2", the total voyage time "ttript" is determined. Given this value, the minimum number of tugs ("mintug") for the trade can be determined by comparing the required monthly cargo flows with the ton-mile capacity of each tug-barge unit. The program then will iterate and calculate new values for "twait1" and "twait2" until the total voyage time "ttript" converges on an unchanging value. If convergence does not occur, an error message is printed. If it is found that the total port waiting time exceeds the time that would be spent for cargo operations, then the program prints out a message stating that the drop-and-swap mode would not be appropriate. Otherwise, the program continues by calculating annual operating costs, as described in the next section.

To calculate the number of tugs required in the integral mode, a simpler approach is used than for the drop-and-swap mode. In this case, the total voyage time is equal to seatime and port time; and, the number of tugs required is simply the minimum that will provide sufficient flow capacity (number of tug-barges per month times barge cargo deadweight) to handle the monthly cargo flow requirements.

2.3.10 Calculation of Annual Operating Costs

Now that the program has determined (1) the duration of the tug seetime ("seatimet") which includes time for unlinking/linking and port delays, (2) the duration of in port time ("portimet") which includes the time that the tug must await cargo operation completion, (3) tug shaft horsepower ("shp") for achieving the specified speed, and (4) the tug engine size ("ihp"); it is now able to proceed to calculate the various components of the total annual operating cost per tug-barge unit. Discussion of the assumptions used in calculating each of the cost components is provided below.

Annual Diesel Fuel Cost. The annual diesel fuel cost is equal to the number of tug voyages per year ("nrtrips" = "tugopdays"/"ttript") times the amount of diesel fuel in long tons consumed per voyage ("fuelcons") times the cost per long ton diesel fuel. (1) The amount of fuel consumed per voyage ("fuelcons") is equal to the tug at sea time in hours ("seatimet") times the hourly at sea fuel consumption rate in tons per hour ("rseafuel") plus the tug in port time in hours ("portimet") times the hourly in port fuel consumption rate ("rportfuel" = 0.125 ton/hr). The at sea fuel consumption rate ("rseafuel") in long tons per hour, is in turn equal to the product of the diesel engine's specific fuel consumption

(1) In the model's base case, diesel fuel cost is assumed to be \$140 per long ton.

rate ("sfc") (1) in pounds per horsepower-hour and the tug's shaft horsepower ("shp"), all divided by 2240.

Annual Lube Oil Costs. The annual lube oil cost is equal to the number of tug voyages per year ("nrtrips") times the amount of lube oil in gallons consumed per voyage ("lubecons") times the cost per gallon for lube oil. (2) The amount of lube oil consumed per voyage ("lubecons") is assumed to be equal to the tug at sea time in hours ("seatimet") times the hourly at sea lube oil consumption rate in gallons per hour ("rlubeoil"). This hourly at sea fuel consumption rate has been assumed to be equal to the tug's shaft horsepower divided by 4000., in gallons per hour.

Annual Crew Costs. The annual crew costs are equal to the average annual crew member's wages and benefits ("cwages") plus subsistence expenses ("csubs"), all times the number of crew members onboard the tug ("nrcrew"). (3)

(1) In the model's base case, a sfc of 0.36 is assumed. This is a reasonable value for the medium speed diesels currently used in high powered tugs. In the future lower sfc's and fuel costs may be obtained with the use of low speed diesels burning heavy fuels.

(2) In the model's base case, lube oil is assumed to be \$1.75 per gallon.

(3) In the model's base case the average crew size has been assumed to be sixteen, which is very close to the minimum manning level of fourteen that the U. S. Coast Guard has previously allowed for "mechanically-linked" push-towed ocean going tug-barges. (The extra two men are used to fill cook/steward positions.) As for the average crew member's wages and subsistence, they were assumed to be \$65,000 and \$3,500 respectively. These values are in reasonable agreement with the Maritime Administration data shown in Table 2.8.

Annual Costs for Maintenance and Repairs, Insurance, and Stores, Supplies, and Equipment. The annual costs for maintenance and repairs ("amandr"), insurance ("ainsur"), and stores, supplies, and equipment ("asupplies") are determined from formulae found in Sharp (1975). These formulae are functions of the tug engine size and total deadweight of the tug-barge unit. Sample values from these formulae, which have been inflated to bring them up to January 1979 levels, are presented in Table 2.7.

Annual Port Charges. The annual port costs are equal to the number of voyages per year times the port charges per voyage. The voyage port charges consist of a fixed charge per port call ("cfixport1" and "cfixport2") plus a variable cost which is a function of the barge size ("cvarport1" x "dwt" and "cvarport2" x "dwt"). (1)

Annual Costs from the Time Value of Cargo. Since the cargo represents a significant capital investment for its owner, the cost of the capital that is tied up while the cargo is being transported should be considered in the total operating costs for the system. The annual cost for the time value of the cargo ("acargo") is equal to the product of the annual cargo flows ("aflowave1" + "aflowave2") times the sea time in years and times the discount rate for capital ("disrate").

(1) In the model's base case, all the port charge factors ("cfixport1", "cfixport2", "cvarport1", and "cvarport2") are assumed to be zero.

Annual Terminal and Storage Operating Costs. The annual terminal operating costs ("atermop") are simply the product of the annual cargo flows ("aflowave1" + "aflowave2") and the average cost per ton cargo for loading/discharging operations ("cvarterm"). Similarly, the annual storage costs ("astorop") are simply the product of the annual cargo flows and the average cost per ton cargo for in port storage. (1)

Calculation of the Total Annual Operations Costs. At this point the total operating costs per tug-barge unit ("aopcost") can be determined. It is simply the sum of fuel, lube oil, crewing and subsistence, maintenance and repair, stores, supplies, and equipment, insurance, cargo value, port charges, and other miscellaneous ("aother") (2) costs. The total operating costs ("totopcost") are then equal to the number of tug-barge units times the operating cost per tug plus the terminal operating costs plus fleet administrative costs ("admin"). (3) For the drop-and-swap modes 3% of the cost of the additional barges required to be stationed at the ports is added to take account for the maintenance and repair, stores, supplies and equipment, and insurance incurred by these addi-

(1) In the model's base case, all terminal and storage costs are assumed to be zero.

(2) In the model's base case it is assumed that the miscellaneous other costs amount to \$30,000 per year for each tug.

(3) In the model's base case it is assumed that the administrative costs per fleet amount to \$150,000 per year.

tional units. For the integral mode, the annual storage costs are added to take in account that shoreside storage facilities must be used for storage rather than the barges that remain in port for the drop-and-swap modes.

To show that the values resulting from the formulae and other assumptions used in calculating the operating costs are reasonable, a comparison of the model's output is made with some Maritime Administration tug-barge data in Table 2.8.

2.3.11 Calculation of System Capital Costs

After the total operating costs are calculated, the program then determines the total system capital cost ("totcapcost") for the drop-and-swap and integral modes of operation. This total capital cost consists of the barge price times the number of barges in the system plus terminal and storage facility capital costs. These costs are adjusted to take account of the increased productivity that result from multi-unit orders by use of the learning curve factor formula found in Sharp (1975). The barge and tug prices are equal to the tug and barge costs ("ctug" and "cbarge") calculated previously, but increased by 10% to take into account shipyard profit. The terminal facility cost is assumed to be some cost factor ("cfixterm") in dollars per ton-day times the sum of the terminal facility loading/discharging rates times the number of facilities per port. The storage facility cost, which pertains to the integral mode only, is assumed to be equal to

the product of some cost factor in dollars per cargo deadweight ton ("cfixstor") times the barge cargo deadweight times the number of terminal facilities ("minbarge1" + "minbarge2"). This formula assumes that a storage facility equal to the barge capacity must be built onshore for each terminal facility. This is only one of many possible ways of estimating shoreside storage requirements and was used just to give an indication of how storage costs might affect the tradeoffs in operating in the drop-and-swap versus integral mode. (1) In a real operating environment, the storage capacity will depend primarily on the ability of the tug-barge systems remaining on rigid schedules.

2.3.12 Calculation of Required Freight Rates

At this point the program has completed all the calculations required to determine the required freight rate for recovering all operating and capital costs for the system. The required freight rate is simply the total system capital cost ("totcapcost") divided by the present value factor ("pvf") plus the total system operating costs ("totopcost"), all divided by the total annual cargo flows. Now all calculations have been completed within the iterative loops. All that remains to be done is the storage within the computer program of the system variable and cost values for the

(1) In the model's base case terminal and storage facility capital costs were assumed to be zero.

iterations resulting in the lowest required freight rate for the port pair under consideration.

2.3.13 Storage of System Parameters for Minimum RFR Iterations

For each iteration of barge size, form, and speed, the program determines the associated required freight rate for the drop-and-swap, unbalanced drop-and-swap, and integral modes of operation. If for a particular iteration the rfr for either regular or unbalanced drop-and-swap mode is found to be lower than that found for previous iterations of the drop-and-swap modes, then the system values for that iteration replace those previously stored in the array "best1". The same pertains to the integral mode which has its optimum system values stored in the array "best2". So, at the end of all the iterations with respect to tug-barge DWT, Cb, L/B, B/T, and speed; the arrays "best1" and "best2" contain the system values for the drop-and-swap mode (the better of the balanced and unbalanced modes) and integral mode, respectively, that result in the minimum rfr. At this point the program will print out (if printed output was requested) the optimum system parameters for both drop-and-swap and integral modes for each trade specified by its port separation distance and loading/discharging rate. It also will store certain of the system parameters such as minimum rfr, optimum barge size and speed, etc., in storage arrays that will be used in the graphical output routines. Following this, the program will

return to the beginning, requesting new inputs.

This concludes the detailed description of the drop-and-swap program. Discussion of the tug-barge powering and barge hull weight subprograms follows in the next two chapters.

Objective Function:

Minimize (Required Freight Rate)

For a port pair trade defined by:

- (1) Port separation distance
- (2) Terminal facilities loading/discharging rates
- (3) Required annual cargo flows

Capacity Constraints:

$$\left\{ \begin{array}{l} \text{Ton-mile:} \\ (\text{No. Tugs}) \times (\text{Barge DWT}) \times \left(\frac{\text{No. Voyages}}{\text{Tug-Year}} \right) \geq (\text{Required Annual Cargo Flows}) \\ \\ \text{Terminal Facility:} \\ (\text{No. Facilities per Port}) \times \left(\frac{\text{Annual Thruput}}{\text{per Facility}} \right) \geq \left(\begin{array}{l} \text{Required Annual} \\ \text{Cargo flows} \\ \text{Through Port} \end{array} \right) \end{array} \right\}$$

Continuity Constraint:

$$(\text{No. Barges}) = (\text{No. Tugs}) + (\text{No. Barges-Port 1}) + (\text{No. Barges-Port 2})$$

Parameter Boundary Constraints:

$$\begin{array}{llll} \text{mindwt} & \leq & \text{Barge DWT} & \leq & \text{maxdwt} \\ \text{minspeed} & \leq & \text{Tug-Barge Speed} & \leq & \text{maxspeed} \\ \text{mincb} & \leq & \text{Barge } C_B & \leq & \text{maxcb} \\ \text{minlb} & \leq & \text{Tug-Barge L/B} & \leq & \text{maxlb} \\ \text{minbt} & \leq & \text{Barge B/T} & \leq & \text{maxbt} \end{array}$$

Integrality Conditions

$$\left\{ \begin{array}{l} \text{No. Tugs} \\ \text{No. Barges - Port 1} \\ \text{No. Barges - Port 2} \end{array} \right\} \text{ integer}$$

FIGURE 2.1
MODEL FORMULATION

$$\text{Required Freight Rate} = \left\{ \frac{\text{Capital Costs}}{\text{PVF}} + \frac{\text{Annual Operating Costs}}{\text{Annual Cargo Flows}} \right\}$$

Capital Costs = (No. Barges) x (Barge Cost) + (No. Tugs) x (Tug Cost)
 + (Terminal Costs) + (Storage Costs)

$$\left\{ \begin{array}{l} \text{Barge Cost} = f\{\text{Barge DWT, L/B, B/T, } C_B\} \\ \text{Tug Cost} = f\{\text{Installed Horse power (IP)}\} \\ \text{IHP} = f\{\text{Barge DWT, L/B, B/T, } C_B, \text{ Tug-Barge Speed}\} \\ \text{Terminal Cost} = f\{\text{Terminal Loading/Discharging Rates}\} \\ \text{Storage Cost} = f\{\text{Barge DWT}\} \end{array} \right.$$

Annual Operating Costs = Fuel Costs + Terminal Costs
 + Storage Costs + M&R Costs
 + Crewing and Subsistence Costs
 + Insurance Costs + Administrative Costs

Fuel Costs = Seetime x (Sea Fuel Consumption Rates)
 + Port time x (Port Fuel Consumption Rate)

$$\left\{ \begin{array}{l} \text{Seetime} = f\left(\frac{\text{Port Separation Distance}}{\text{Tug-Barge Speed}}\right) \\ \text{Port Time} = f\left(\frac{\text{Barge DWT}}{\text{Terminal Facility L/D Rate}}\right) \\ \text{Sea Fuel Consumption Rate} = f\left\{ \begin{array}{l} \text{Barge DWT, L/B, B/T} \\ \text{Block Coefficient;} \\ \text{Tug-Barge Speed} \end{array} \right\} \\ \text{Port Fuel Consumption Rate} = f(\text{Hotel Load}) = \text{constant} \end{array} \right.$$

Terminal Costs = f(Annual Cargo Flows)
 Storage Costs = f(Annual Cargo Flows)

$$\left. \begin{array}{l} \text{Maintenance and Repair Costs} \\ \text{Supplies and Equipment Costs} \\ \text{Insurance Costs} \end{array} \right\} = f(\text{Barge DWT, Tug IHP})$$

$$\left. \begin{array}{l} \text{Crew Wages and Benefits} \\ \text{Crew Subsistence} \end{array} \right\} = f(\text{crew size}) = \text{constant}$$

Administrative Costs = constant

FIGURE 2.2

DEFINITION OF REQUIRED FREIGHT RATE

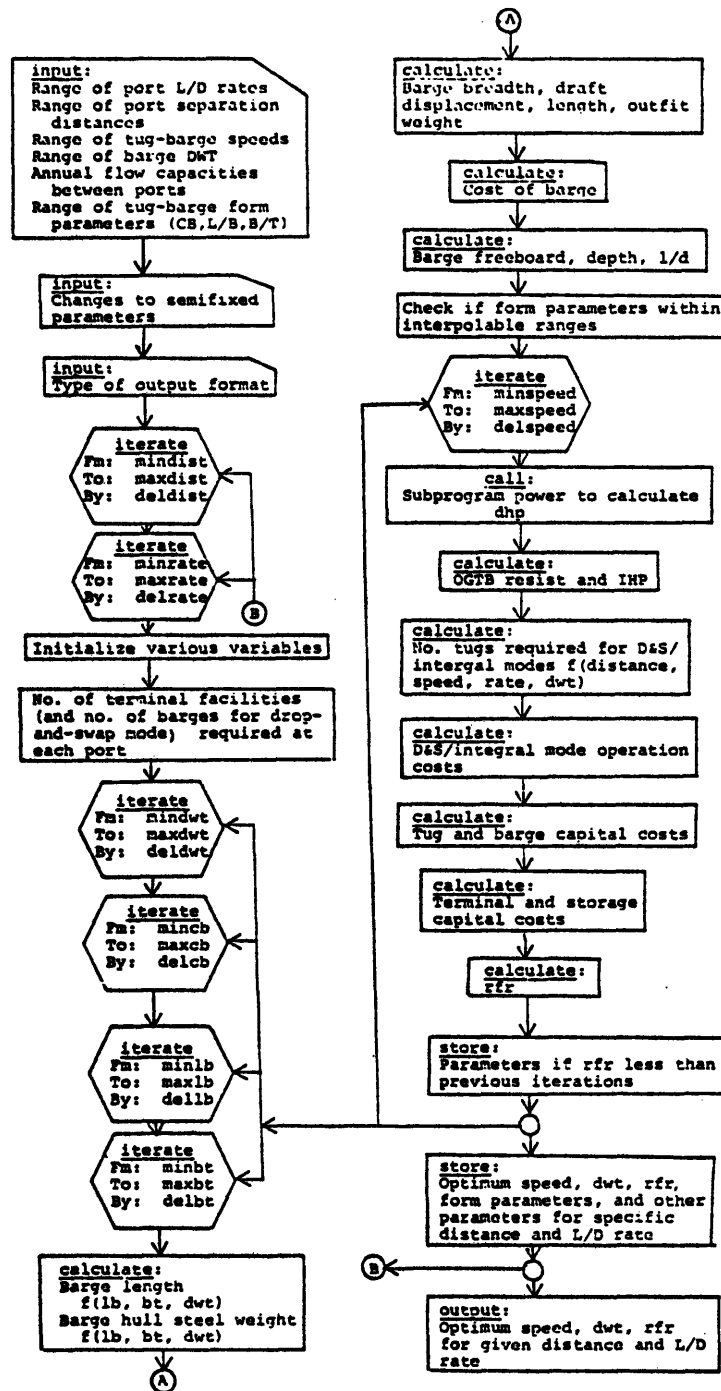


FIGURE 2.3

SUMMARY FLOWCHART FOR DROP-AND-SWAP
COMPUTER MODEL

drop and swap

Input via list format the following parameters:

Do you wish to specify individual port L-D rates?: no

minrate, maxrate, delrate: 2000.,14000.,2000.,

mindist, maxdist, deldist: 500.,5000.,1500.,

minspeed, maxspeed, delspeed: 6.,13.,1.,

mindwt, maxdwt, deldwt: 5000.,100000.,5000.,

aflowave1, aflowave2: 1000000.,0.,

mincb, maxcb, delcb: .75,.83,.01,

minlb, maxlb, dellb: 6.,6.4,.2,

minbt, maxbt, delbt: 2.,3.,.5,

input changes to semifixed data via get data format maxt1=38., maxt2=38.;

Do you want printed output?: y

Do you want detailed output?: y

FIGURE 2.4

SAMPLE INPUT FOR THE DROP-AND-SWAP MODEL

This is the semi-fixed parametric data file.

tugopdays	350.00
bargeopdays	350.00
tlink (hrs)	4.00
tunlink (hrs)	4.00
servmargin	0.20
fuelmargin	0.20
sfc (lb/HP-hr)	0.36
cfuel (\$/LT)	140.00
clube (\$/gal)	1.75
nrcrew	16.00
cwages (\$/yr)	65000.00
csubs (\$/yr)	3500.00
csteelt (\$1000/LT)	2.94
coutfitt (\$1000/LT)	15.08
csteelb (\$1000/LT)	1.10
coutfitb (\$1000/LT)	12.82
ltug (ft)	140.00
wmisc (LT)	460.00
aother (\$/yr)	30000.00
admin (\$/yr)	150000.00
cfixport1 (\$/call)	0.00
cfixport2 (\$/call)	0.00
cvarport1 (\$/DWT)	0.00
cvarport2 (\$/DWT)	0.00
cfixterm (\$/DWT-day)	0.00
cvarterm (\$/DWT)	0.00
cfixstor (\$/DWT)	0.00
cvarstor (\$/DWT)	0.00
delay1 (hrs)	4.00
delay2 (hrs)	4.00
maxl (ft)	999.00
maxb (ft)	999.00
maxt1 (ft)	99.00
maxt2 (ft)	99.00
disrate	0.10
econlife (yrs)	20.00
inflafctr	0.00
vcargo (\$/DWT)	200.00

BASE CASE VALUES FOR
SEMI-FIXED PARAMETRIC DATA

FIGURE 2.5

DATE (1000)	SPD (KTS)	RFR (\$)	CB LENGTH	L/B	B/T	L/D	ACOST (\$M)	CTUG (\$M)	CHARGE (\$M)	IHP NO_TUG/ITB	MTRIPS	SEATIME (DAYS)	PORTIME (DAYS)	AFLOWCAP T/AIT/1/1 (DAYS)
5	6.0	14.18	0.78 200.9	6.00	2.00	7.0	14.180	5.941	1.194	516	44.06	7.94	0.00	220280
5	6.0	16.53	0.78 200.9	6.00	2.00	7.0	16.530	5.941	1.194	516	33.51	7.94	2.50	167553
5	6.0	20.83	0.78 200.9	6.00	2.00	7.0	20.830	5.941	1.194	516	28.51	7.94	2.50	142534
5	7.0	11.84	0.78 200.9	6.00	2.00	7.0	11.816	6.085	1.194	800	50.34	6.95	0.00	251712
5	7.0	16.84	0.78 200.9	6.00	2.00	7.0	16.816	6.085	1.194	800	37.03	6.95	2.50	185132
5	7.0	21.18	0.78 200.9	6.00	2.00	7.0	21.118	6.085	1.194	800	31.01	6.29	5.00	155033
5	8.0	12.18	0.78 200.9	6.00	2.00	7.0	12.117	6.281	1.194	1186	56.38	6.21	2.50	281573
5	8.0	14.52	0.78 200.9	6.00	2.00	7.0	14.520	6.281	1.194	1186	40.19	6.21	2.50	200457
5	8.0	21.64	0.78 200.9	6.00	2.00	7.0	21.618	6.281	1.194	1186	33.20	5.54	5.00	166008
5	9.0	12.56	0.78 200.9	6.00	2.00	7.0	12.558	6.408	1.194	1610	62.17	5.63	0.00	310555
5	9.0	14.89	0.78 200.9	6.00	2.00	7.0	14.890	6.408	1.194	1610	43.05	5.63	2.50	215262
5	9.0	18.36	0.78 200.9	6.00	2.00	7.0	18.360	6.408	1.194	1610	35.13	4.96	5.00	175651
5	10.0	10.47	0.78 200.9	6.00	2.00	7.0	10.471	7.056	1.194	2685	67.74	5.17	0.00	338710
5	10.0	15.84	0.78 200.9	6.00	2.00	7.0	15.845	7.056	1.194	2685	45.65	5.17	2.50	228261
5	10.0	19.38	0.78 200.9	6.00	2.00	7.0	19.317	7.056	1.194	2685	36.84	4.50	5.00	184211
5	11.0	11.03	0.78 200.9	6.00	2.00	7.0	11.003	7.518	1.194	3560	73.10	4.79	0.00	365506
5	11.0	16.60	0.78 200.9	6.00	2.00	7.0	16.598	7.518	1.194	3560	48.02	4.79	2.50	240126
5	11.0	20.17	0.78 200.9	6.00	2.00	7.0	20.169	7.518	1.194	3560	38.37	4.12	5.00	191860
5	12.0	14.12	0.78 200.9	6.00	2.00	7.0	14.122	9.858	1.194	7799	78.26	4.47	2.50	391304
5	12.0	16.48	0.78 200.9	6.00	2.00	7.0	16.485	9.858	1.194	7799	50.20	4.47	2.50	250936
5	12.0	24.17	0.78 200.9	6.00	2.00	7.0	24.172	9.858	1.194	7799	39.75	3.81	5.00	198718
5	13.0	15.68	0.78 200.9	6.00	2.00	7.0	15.676	11.106	1.194	9942	83.23	4.21	0.00	416159
5	13.0	17.97	0.78 200.9	6.00	2.00	7.0	17.971	11.106	1.194	9942	52.20	4.21	2.50	260994
5	13.0	21.21	0.78 200.9	6.00	2.00	7.0	21.209	11.106	1.194	9942	40.99	3.54	5.00	204955

DATE (1000)	SPD (KTS)	RFR (\$)	CB LENGTH	L/B	B/T	L/D	ACOST (\$M)	CTUG (\$M)	CHARGE (\$M)	IHP NO_TUG/ITB	MTRIPS	SEATIME (DAYS)	PORTIME (DAYS)	AFLOWCAP T/AIT/1/1 (DAYS)
5	6.0	14.30	0.78 222.6	6.00	2.50	8.9	14.300	5.934	1.315	502	44.06	7.94	0.00	220280
5	6.0	16.63	0.78 222.6	6.00	2.50	8.9	16.630	5.934	1.315	502	33.51	7.94	2.50	167553
5	6.0	20.98	0.78 222.6	6.00	2.50	8.9	20.980	5.934	1.315	502	28.51	7.94	2.50	142534
5	7.0	11.98	0.78 222.6	6.00	2.50	8.9	11.980	6.095	1.315	821	50.34	6.95	0.00	251712
5	7.0	16.98	0.78 222.6	6.00	2.50	8.9	16.976	6.095	1.315	821	37.03	6.95	2.50	185139
5	7.0	21.33	0.78 222.6	6.00	2.50	8.9	21.331	6.095	1.315	821	31.01	6.29	5.00	155053
5	8.0	12.37	0.78 222.6	6.00	2.50	8.9	12.373	6.320	1.315	1262	56.38	6.21	0.00	281979
5	8.0	14.70	0.78 222.6	6.00	2.50	8.9	14.695	6.320	1.315	1262	40.19	6.21	2.50	200357
5	8.0	21.86	0.78 222.6	6.00	2.50	8.9	21.861	6.320	1.315	1262	33.20	5.54	5.00	166008
5	9.0	12.98	0.78 222.6	6.00	2.50	8.9	12.976	6.659	1.315	1923	62.17	5.63	0.00	310355
5	9.0	15.28	0.78 222.6	6.00	2.50	8.9	15.282	6.659	1.315	1923	43.05	5.63	2.50	215262
5	9.0	18.77	0.78 222.6	6.00	2.50	8.9	18.771	6.659	1.315	1923	35.13	4.96	5.00	175651
5	10.0	10.54	0.78 222.6	6.00	2.50	8.9	10.541	7.025	1.315	2626	67.74	5.17	0.00	338710
5	10.0	15.69	0.78 222.6	6.00	2.50	8.9	15.695	7.025	1.315	2626	45.65	5.17	2.50	228261
5	10.0	19.42	0.78 222.6	6.00	2.50	8.9	19.420	7.025	1.315	2626	36.84	4.50	5.00	184211
5	11.0	11.53	0.78 222.6	6.00	2.50	8.9	11.526	7.766	1.315	4025	73.10	4.79	0.00	365506
5	11.0	17.12	0.78 222.6	6.00	2.50	8.9	17.122	7.766	1.315	4025	48.02	4.79	2.50	240126
5	12.0	12.37	0.78 222.6	6.00	2.50	8.9	12.375	7.766	1.315	4025	38.37	4.12	5.00	191860
5	12.0	14.70	0.78 222.6	6.00	2.50	8.9	14.701	8.416	1.315	5222	78.26	4.47	2.50	391304
5	12.0	21.79	0.78 222.6	6.00	2.50	8.9	21.788	8.416	1.315	5222	50.20	4.47	2.50	250936
5	13.0	17.00	0.78 222.6	6.00	2.50	8.9	17.004	12.100	1.315	11599	83.23	4.21	0.00	198718
5	13.0	19.28	0.78 222.6	6.00	2.50	8.9	19.278	12.100	1.315	11599	52.20	4.21	2.50	260994
5	13.0	22.59	0.78 222.6	6.00	2.50	8.9	22.588	12.100	1.315	11599	40.99	3.54	5.00	204955

DATE (1000)	SPD (KTS)	RFR (\$)	CB LENGTH	L/B	B/T	L/D	ACOST (\$M)	CTUG (\$M)	CHARGE (\$M)	IHP NO_TUG/ITB	MTRIPS	SEATIME (DAYS)	PORTIME (DAYS)	AFLOWCAP T/AIT/1/1 (DAYS)
5	6.0	14.92	0.78 242.4	6.00	3.00	10.8	14.924	5.967	1.708	568	44.06	7.94	0.00	220280
5	6.0	17.17	0.78 242.4	6.00	3.00	10.8	17.172	5.967	1.708	568	33.51	7.94	2.50	167553
5	6.0	21.54	0.78 242.4	6.00	3.00	10.8	21.540	5.967	1.708	568	28.51	7.94	2.50	142534
5	7.0	12.54	0.78 242.4	6.00	3.00	10.8	12.579	6.150	1.708	929	50.34	6.95	0.00	251712

FIGURE 2.6

EXAMPLE OF DETAILED OUTPUT FROM THE DROP-AND-SWAP COMPUTER MODEL

$$\text{Barge Freeboard (inches)} = \text{"fbd"} = 0.75 \times \underbrace{\left(\frac{\text{Minimum Freeboard Value}}{\text{Found in Table 2.5}} \times \left(\frac{\text{"cb"} + 0.68}{1.38} \right) + \text{"shrfact"} + \text{"ldfact"} \right)}_{\text{"minfbd"}}$$

Where: 0.75 accounts for the 25% freeboard reduction pertaining to unmanned barges

"shrfact" = sheer factor
correction

$$= 0.0375 \times \text{"lbarge"} + 3.75$$

"ldfact" = length-depth ratio
correction

$$= \begin{cases} = 0 & \text{if } \frac{\text{"lbarge"}}{\text{depth}} > 15 \\ \\ = \left(\text{depth} - \frac{\text{"lbarge"}}{15} \right) \times \begin{cases} \frac{\text{"lbarge"}}{132.2} \\ 1 - \frac{\text{"lbarge"}}{12 \times 131.2} \end{cases} & \text{if } \begin{cases} \frac{\text{"lbarge"}}{\text{depth}} < 15 \\ \text{"lbarge"} < 393.6' \end{cases} \\ \\ = 4 \times \left(\text{depth} - \frac{\text{"lbarge"}}{15} \right) & \text{if } \begin{cases} \frac{\text{"lbarge"}}{\text{depth}} < 15 \\ \text{"lbarge"} > 393.6' \end{cases} \end{cases}$$

$$\text{where depth} = \left(\frac{\text{"minfbd"}}{12} + \text{"tbarge"} \right)$$

SOURCE: Chapter III of the IMCO International Conference on Load Lines, 1966

FIGURE 2.8

EQUATIONS USED IN THE CALCULATION OF BARGE FREEBOARD

TABLE 2.1
RANGES FOR SYSTEM PARAMETERS

Parameter	Lower Bound	Upper Bound	Restrictions/ Comments	Reason for Bound
Barge DWT	9,000 6,000	80,000 100,000	Reduced confidence range	Sharp (1975) Valid ranges for operating cost formulae
Barge Length	100'	750'		ABS (1973) Valid ranges for barge hull weight formula
Froude Number	0	0.22	$F_n = \frac{0.298 \text{ Tug Speed}}{\sqrt{\text{Two-Barge Length}}}$	Tsuchida (1969) Valid range for residual resistance coefficients
Barge C_B^*	0.775 0.75	0.835 0.85	Reduced confidence range	" " "
Tug-Barge L/B*	6.2 6.0	7.6 8.0	Reduced confidence range	" " "
Barge B/T	2.46 2.00	2.76 3.25	Reduced confidence range	" " "
Barge L/D	0	16.0		ABS (1973) valid range for barge hull weight formula
Tug IHP	5000 3000	35,000 35,000	Reduced confidence range	Sharp (1975) Valid range for operating cost formulae and tug capital cost formula

*Either C_B or L/B can be in the reduced confidence range, but not both.

TABLE 2.2
FORMULAE FOR DETERMINING NUMBER OF TERMINAL
FACILITIES/BARGES AT EACH PORT

Port	Special Conditions	Formulae
1	none	"minbargel"= $\text{ceil}(\frac{\text{"mflowave1"}}{30.5\text{"rload1"}} + \frac{\text{"mflowave2"}}{30.5\text{"runload1"}})$
	"rload1"=0	"minbargel"= $\text{ceil}(\frac{\text{"mflowave2"}}{30.5\text{"runload1"}})$
	"runload1"=0	"minbargel"= $\text{ceil}(\frac{\text{"mflowave1"}}{30.5\text{"rload1"}})$
2	none	"minbarge2"= $\text{ceil}(\frac{\text{"mflowave1"}}{30.5\text{"runload2"}} + \frac{\text{"mflowave2"}}{30.5\text{"rload2"}})$
	"rload2"=0	"minbarge2"= $\text{ceil}(\frac{\text{"mflowave1"}}{30.5\text{"runload2"}})$
	"runload2"=0	"minbarge2"= $\text{ceil}(\frac{\text{"mflowave2"}}{30.5\text{"rload2"}})$

Note: The function ceil in the above formulae is used to round to the next largest integer the expression in parenthesis.

TABLE 2.3

BARGE HULL WEIGHT AND LENGTH FORMULAE

Parameter	Block Co-efficient	Formula
Barge Length "lbarge"	0.75	"lbarge75"= $e^{1.128 lb^{0.742} bt^{0.382} dwt^{0.336}}$
	0.80	"lbarge80"= $e^{1.108 lb^{0.739} bt^{0.381} dwt^{0.336}}$
	0.85	"lbarge85"= $e^{1.088 lb^{0.737} bt^{0.379} dwt^{0.336}}$
Barge Hull Weight	0.75	"hullwt75"= $e^{-6.206 lb^{1.348} bt^{0.884} dwt^{1.104}}$
	0.80	"hullwt80"= $e^{-6.396 lb^{1.368} bt^{0.909} dwt^{1.111}}$
"wsteelb"	0.85	"hullwt85"= $e^{-6.569 lb^{1.391} bt^{0.930} dwt^{1.117}}$

TABLE 2.4

BARGE OUTFIT WEIGHT

Formula: "woutfitb" = $\max(50, 1.496 \times "lbarge" - 284.24)$

Barge Length	Outfit Weight
200	50.00
250	89.76
300	164.56
350	239.36
400	314.16
450	388.96
500	463.76
550	538.56
600	613.36
650	688.16
700	762.96
750	837.76

TABLE 2.5
MINIMUM FREEBOARD VALUES

Barge Length	Minimum Freeboard	Barge Length	Minimum Freeboard	Barge Length	Minimum Freeboard
100'	9.8"	320'	43.2"	540'	86.3"
110	10.8	330	45.0	550	88.0
120	11.9	340	46.9	560	89.6
130	13.0	350	48.8	570	91.1
140	14.2	360	50.7	580	92.6
150	15.5	370	52.7	590	94.1
160	16.9	380	54.7	600	95.5
170	18.3	390	56.8	610	96.9
180	19.8	400	58.8	620	98.3
190	21.3	410	60.9	630	99.6
200	22.9	420	62.9	640	100.9
210	24.5	430	65.0	650	102.1
220	26.2	440	67.0	660	103.3
230	27.8	450	69.1	670	104.4
240	29.5	460	71.1	680	105.5
250	31.1	470	73.1	690	106.6
260	32.8	480	75.1	700	107.7
270	34.6	490	77.1	710	108.7
280	36.3	500	79.0	720	109.7
290	38.0	510	80.9	730	110.7
300	39.7	520	82.7	740	111.7
310	41.4	530	84.5	750	112.6

SOURCE: Table A, Chapter III, IMCO International Convention on Load Lines, 1966.

TABLE 2.6

TUG CAPITAL COSTS

$$\text{Tug Hull Weight} = \text{"wsteelt"} = 0.64 \left(\frac{\text{IHP}}{1000} \right)^2 + 16.79 \left(\frac{\text{IHP}}{1000} \right) + 378$$

$$\text{Tug outfit Weight} = \text{"woutfitt"} = 0.1866 \left(\frac{\text{IHP}}{1000} \right)^2 + 2.733 \left(\frac{\text{IHP}}{1000} \right) + 154$$

$$\text{Tug Machinery Weight} = \text{"wmacht"} = -0.08889 \left(\frac{\text{IHP}}{1000} \right)^2 + 32.88 \left(\frac{\text{IHP}}{1000} \right) + 4.999$$

$$\text{Tug Machinery Cost} = 1,300 * (0.314 * \text{IHP} + 1730)$$

$$\text{Tug Cost} = 1,000 [2.94 \left(\frac{\text{Tug Hull Weight}}{\text{Weight}} \right) + 15.08 \left(\frac{\text{Tug Outfit Weight}}{\text{Weight}} \right) + \left(\frac{\text{Tug Machinery Cost}}{\text{Cost}} \right)]$$

IHP	Tug Hull Weight (LT)	Tug Outfit Weight (LT)	Tug Machinery Weight (LT)	Tug Machinery Cost (\$1000)	Tug Cost (\$1000)
2,000	414.14	160.21	70.40	3065.4	6,698.1
2,500	423.98	162.00	86.64	3269.5	6,958.1
3,000	434.13	163.88	102.84	3473.6	7,220.4
4,000	455.40	167.92	135.10	3881.8	7,752.0
5,000	477.95	172.33	167.18	4290.0	8,293.0
6,000	501.78	177.12	199.08	4698.2	8,843.3
7,000	526.89	182.27	230.80	5106.4	9,403.1
8,000	553.28	187.81	262.35	5515.6	9,972.3
9,000	580.95	193.71	293.72	5922.8	10,550.8
10,000	609.90	199.99	324.91	6331.0	11,138.7
12,000	671.64	213.67	386.76	7147.4	12,342.8
14,000	738.50	228.84	447.90	7963.8	13,584.4
16,000	810.48	245.50	508.32	8780.2	14,863.5
18,000	887.58	263.65	568.04	9596.6	16,180.2
20,000	969.80	283.30	627.04	10413.0	17,534.4
22,500	1079.78	309.96	699.80	11433.5	19,280.1
25,000	1197.75	338.95	771.44	12454.0	21,084.4
27,500	1323.73	370.27	841.98	13474.5	22,947.3
30,000	1457.70	403.93	911.40	14495.0	24,869.0

SOURCE: Sharp (1975)

*Cost factors inflated 30% from values given in source.

TABLE 2.7

ANNUAL OPERATING COSTS FOR SUPPLIES AND EQUIPMENT, MAINTENANCE AND
REPAIR, AND INSURANCE

Supplies and Equipment = "asupplies" = $1.3 \cdot (40 + 0.0018 \cdot \text{IHP} + 0.00026 \cdot \text{TDWT})$

Maintenance and Repair = "amandr" = $1.5 \cdot (128.8 + 4.539 \cdot (\frac{\text{IHP}}{1000}) - 0.04 \cdot (\frac{\text{IHP}}{1000})^2 + 2.477 \cdot (\frac{\text{TDWT}}{1000}) - 0.009107 \cdot (\frac{\text{TDWT}}{1000})^2)$

Insurance = "ainsur" = $1.5 \cdot (210 + 0.0036 \cdot \text{IHP} + 0.0018 \cdot \text{TDWT})$

TDWT	IHP	Supplies & Equipment (\$1000)	Maintenance & Repair (\$1000)	Insurance (\$1000)
5,000	2,000	58.4	224.8	339.3
"	6,000	67.7	250.1	360.9
"	10,000	77.1	273.5	382.5
"	14,000	86.5	295.0	404.1
20,000	2,000	63.4	275.4	379.8
"	6,000	72.8	300.7	401.4
"	10,000	82.2	324.1	423.0
"	14,000	91.5	345.6	444.6
35,000	2,000	68.5	319.9	420.3
"	6,000	77.9	345.2	441.9
"	10,000	87.2	368.6	463.5
"	14,000	96.6	390.1	485.1
50,000	2,000	73.6	358.2	460.8
"	6,000	82.9	383.5	482.4
"	10,000	92.3	406.9	504.0
"	14,000	101.7	428.4	525.6
"	18,000	111.0	447.9	547.2
65,000	2,000	78.7	390.4	501.3
"	6,000	88.0	415.7	522.9
"	10,000	97.4	439.1	544.5
"	14,000	106.7	460.6	566.1
"	18,000	116.1	480.1	587.7

SOURCE: Sharp (1975)

NOTE: TDWT=Cargo Deadweight + Fuel Weight +Miscellaneous Weight

*Inflation Factor to bring cost to 1 Jan 79 level

TABLE 2.8

COMPARISON OF MODEL AND MARAD DATA
FOR ANNUAL OPERATING COSTS

Vessel Type	Annual Crew Wages	Annual Subsistence Costs	Annual M&R Costs	Annual Supplies & Equipment	Annual Insurance	Annual M&R Supplies & Equipment	Other Annual Costs	Total Operating Costs
5,000 DWT 3,000 IHP cy 1978	658,152	23,725	200,000	63,270	496,000	759,270	36,000	1,477,147
Model 1 Jan 1979	845,000	45,500	231,321	60,710	344,700	636,731	30,000	1,557,231
33,500 DWT 15,000 IHP 1 Jan 77	821,250	47,815	405,150	127,385	542,755	1,075,290	29,200	1,973,555
Model 1 Jan 1979	1,040,000	56,000	390,966	98,423	486,450	975,839	30,000	2,101,839
40,000DWT 18,000 IHP cy 1980	1,060,000	50,000	460,683	71,317	352,250	884,250	189,600	2,183,850
Model 1 Jan 1979	1,040,000	56,000	442,247	107,640	520,200	1,070,387	180,000	2,346,387
40,000 DWT 18,000 IHP cy 1980	1,100,000	60,000	705,000	220,000	561,000	1,486,000	20,000	2,666,000
Model 1 Jan 1979	1,040,000	56,000	442,747	107,640	520,200	1,070,387	30,000	2,196,387

SOURCE: Data Received from Maritime Administration Office of Ship Construction,
December 1978.

CHAPTER 3

BARGE DESIGN PROGRAM TO ESTIMATE HULL STEEL WEIGHT

3.1 Introduction

The barge design program has been developed to provide an estimate of the hull weight of tank barges as a function of their size and form. This weight estimate is used by the drop-and-swap program to calculate the barge capital construction costs. This is done by multiplying the hull steel weight estimate by a dollar per ton factor to yield the hull cost which is added to the outfit cost to yield the barge cost estimate. This approach was taken rather than a simple regression equation through historical costs because too few large tank barges have been built to yield a reliable prediction equation, and those that have been built were of various designs and arrangements so that their costs were really not comparable. Also, since barge capital costs make up a significant part of the total cost of a tug-barge project, it was deemed necessary to develop a non-trivial model for obtaining reasonably accurate and consistent estimates of the hull steel weight. An additional benefit derived from the barge design model is that it provides hull weights for various barge forms

(length-breadth ratio, breadth-draft ratio, and block coefficient) for a given deadweight. This information is used by the drop-and-swap program to determine the least costly form with respect to capital and propulsive costs.

The barge design model's approach for estimating the weight of a barge of given size, form, and longitudinal frame and transverse girder spacing is basically the same as used by a naval architect in his preliminary design work. That is, the program determines the minimum weight mid-ship scantlings that will satisfy the requirements promulgated in the American Bureau of Shipping's offshore barge (ABS (1973)) and steel vessel (ABS (1978)) rules and that will conform with usual shipyard practice. From the midship scantlings a weight per longitudinal foot can be determined from which the hull weight can be estimated. The program does optimize in the sense that it selects the longitudinal frame and transverse girder spacings within user specified ranges and increments that yield the minimum weight scantlings. It was felt that this parametric optimization procedure which determines a local minimum within the specified parametric ranges was the simplest though not most elegant approach to take. A more theoretical optimization procedure using a stress and bending moment analysis could probably be used to develop a more rational barge design model, but it most likely could not take account of the complexities involved in classification society rules that are used in barges constructed today.

It should be mentioned that it has been assumed that the minimum weight barge would be the optimum. In reality, the minimum weight design may not be the least costly. For example, a design with wider longitudinal spacing may be less expensive due to reduced welding and cutting costs than the less heavy design of narrower spacing. However, it was found from the model's outputs that the calculated hull weight was not very sensitive to longitudinal spacing; (1) so that as long as a reasonable minimum longitudinal spacing is specified (eliminating the possibility of unreasonably small and numerous longitudinal frames), the minimum hull weight design should not be very far from the optimal, minimum cost design.

The design model was originally developed for the single-skin, center longitudinal tank barge design shown in Figure 3.1 for the following two reasons: (1) these barges have the simplest scantling form and thus were the easiest to model, and (2) these types of barges have been the most prevalent of the large ocean going barges so that actual barge hull weight data could be obtained to calibrate the model. However, if proposed U. S. Coast Guard anti-pollution regulations become effective, no single-skin tank barges will be constructed in the future. Tank barges will be required either to have double hulls or protective segregated ballast

(1) As an example, the minimum hull steel weights calculated for the six longitudinal spacings (1.50'-2.75') shown in Figure 3.9 differ less than 3%.

tanks located either in double sides, bottoms, or hulls. If this change occurs, the single-skin design model will have to be modified to account for the additional internal structure. In order to see how just one of these types of modifications will affect the hull weight of a barge of specified deadweight, the single-skin model was written with the option for calculating the hull weights of barges with double-bottom scantlings as shown in Figure 3.2. The comparison of single-skin and double-bottom tank barges presented in Section 3.6 indicates the approximate additional weight (and cost) such internal structure will cause.

In the text that follows, the logical structure of the single-skin tank barge program and the modifications required to allow for double-bottoms are presented. This is followed by a presentation and discussion of the results of the program.

3.2 Logical Structure

The logical structure of the program is shown in flow chart form in Figure 3.3. In the text following, each block of the flow chart will be expanded so that the equations (with associated references) and assumptions used will be enumerated. It should be mentioned that it has been assumed for simplicity that deck, bottom, side, and bulkhead longitudinal frame spacings will be the same. The same holds for transverse girder spacings.

3.2.1 Input of Parameter Values

The program requires as input the barge principal dimensions, block coefficient, and scantling arrangement option (single-skin or double-bottom), as well as the parametric range and increment for the spacing of longitudinal frames and transverse girders. The ranges and increments chosen should conform with current shipbuilding practice. Specifically, the following values for the barge design must be provided:

"l":	scantling length at the waterline
"b":	molded breadth
"d":	molded depth
"t":	summer waterline draft
"cb":	block coefficient
"minlonspac":	minimum longitudinal frame spacing to be investigated
"maxlonspac":	maximum longitudinal frame spacing to be investigated
"minxvrspac":	minimum transverse girder spacing to be investigated
"maxxvrspac":	maximum transverse girder spacing to be investigated
"incrlnspac":	the increment to be used in the parametric variation of longitudinal frame spacing from its minimum to its maximum allowed value
"incrxxvrspac":	the increment to be used in the parametric variation of transverse web spacing from its minimum value to its maximum allowed value

Limitations on the values of these parameters as given within the specified paragraphs of ABS (1973) are as follows:

<u>Limitation</u>	<u>Paragraph</u>
$100 \text{ ft} \leq "l" \leq 750 \text{ ft}$	(Table 3.2)
$"l"/"d" < 15$	(3.5)
$"\text{maxlonspac}" < 3 \frac{1}{3} \text{ ft}$	(5.1.1)

3.2.2 Calculation of Required Section Modulus

In accordance with ABS (1973, par. 3.3.1), the program determines the required hull-girder section modulus at amidships ("smreq") as follows:

$$"smreq" = "fact" \times "b" \times ("cb" + 1/2)$$

where "fact" has its value determined by its length from ABS (1973, Table 3.2) which is replicated in Table 3.1. This section modulus is the best approximation that can be made for design purposes without knowing the actual bending moments that the barge would experience in still water.

3.2.3 Calculation of the Unsupported Spans of Transverse Girders and Required Number of Stanchions

The unsupported span of deck and bottom transverse girders is shown as "l2" in Figures 3.1 and 3.2. "l2" has

been limited to less than 15 feet in accordance with ABS (1978, para. 11.5) by inserting stanchions equally spaced between the bulkhead and side girders. To obtain the required span and the number of stanchions ("stans"), the program assumes certain arbitrary values for the depth of longitudinal bulkhead and side girders as shown in Table 3.2. These assumptions, although arbitrary, have been shown reasonable when compared with actual barge scantlings.

The unsupported span of longitudinal bulkhead and side girders is shown as "l4" in Figure 3.1. It is essentially the length between the top of the bottom girder (or double bottom plate) to the bottom of the deck girder.

3.2.4 Beginning of Program Iterations with respect to Transverse Girder and Longitudinal Frame Spacing

At this point the program has finished its initialization and begins to calculate values with respect to a specific longitudinal frame and transverse girder spacing. The first thing that is calculated is the spacing of transverse bulkheads for tanks. In this calculation, it has been assumed that tank length will be limited to less than 15% of the barge length in accordance with MARPOL '73, that tanks will be of equal length, and that the minimum integral number of transverse girders are to be included in each tank.

Following this calculation, the program begins the main task of calculating the scantlings required to fulfill the minimum requirements specified in the ABS (1973) and (1978).

Scantlings are calculated for bottom members, followed by side members, deck members, and finally the longitudinal bulkhead members. The actual scantlings are calculated by the use of two subroutines--"smact1" for longitudinal frames and "smact2" for transverse girders.

The subroutine "smact1" requires the following inputs:

1. The thickness of the shell plating
2. The minimum required section modulus for the member
3. The effective width of shell plating to be used in the section modulus calculations within the subroutine.

The subroutine "smact2" requires similar inputs, except for the addition of the depth of the associated longitudinal frame which is obtained as output from subroutine "smact1".

3.2.5 Derivation of the Inputs for Subroutines "smact1" and "smact2"

3.2.5.1 Plate thickness. The thicknesses of plating to be used as inputs into subroutines "smact1" and "smact2" are the minimums allowed by ABS (1973). The appropriate rules are summarized in Table 3.3. These thicknesses are rounded up to the nearest 1/16 inch in order to conform with standard plate stock. For the same reasoning, plate stock is limited to 1.4375 inches.

3.2.5.2 Section modulus. The section moduli to be used as inputs into subroutines "smact1" and "smact2" are the minimums required by ABS (1973). The appropriate rules, including the definition of pressure heads, are summarized in

Table 3.4 for longitudinal frames and in Table 3.5 for transverse girders.

3.2.5.3 Effective plating. The width of effective plating to be used in subroutines "smact1" and "smact2" is the minimum of the following three values based on the referenced rules:

1. Longitudinal frame spacing ("lonspac") for subroutine "smact1"; transverse girder spacing ("xvrspac") for subroutine "smact2", per ABS (1973, par. 1.25.1)
2. One third the unsupported span of the member (l in Tables 3.4 and 3.5), per ABS (1973, par. 1.25.1)
3. Sixty times the plate thickness as per Evans (1975)

3.2.6 Calculation of Hull Girder Section Modulus

At this point, the minimum scantlings required to satisfy the individual frames and girders have been determined by repetitive calling of the subroutines "smact1" and "smact2". The next step is to determine whether the individual longitudinal members (plates and longitudinal frames) provide sufficient material to satisfy the minimum hull-girder section modulus requirement calculated in Section 3.2.2. This is done by using the method shown in Comstock (1967, pp.183-184) which uses all longitudinal material in the calculation. In our case, this includes bottom, side, deck and longitudinal bulkhead longitudinal frames and plates.

3.2.7 Strategy to Satisfy Minimum Hull Girder Section Modulus Requirement

If the calculation of the bottom and deck hull-girder section modulus results in values greater than the minimum required, then the hull weight is calculated, as explained in the Section 3.2.8. Otherwise, it is necessary to increase the dimensions of certain of the longitudinal members in such a way that should provide the maximum increase in the calculated deck and/or bottom section moduli with the smallest amount of additional material. Since the barge hull is essentially a stiffened box girder, it is apparent that material should be added to the extreme flanges--the deck and bottom plates and longitudinals--to get the maximum increase in section modulus for the given material added. This is exactly what the program does. It begins with the deck longitudinal members since the deck section modulus will be initially smaller than the bottom section modulus due to smaller hydrostatic pressure experienced by the deck members.

Specifically, the program increases the dimensions of deck scantlings in the order: (1) deckplate (by 1/16"s), (2) deck longitudinal flange width (by 1/2"s), and (3) deck longitudinal flange thickness (by 1/16"s). The program increases the first dimension on the list that is not constrained by usual yard practice or structural considerations. This procedure continues until the deck modulus is at least as great as the bottom modulus. Then, if

the deck and bottom section moduli are still smaller than the required section modulus, the program will increase the dimensions of the bottom scantlings in the order: (1) bottom plate (by 1/16"s), (2) bottom longitudinal flange width (by 1/2"s), and (3) bottom longitudinal flange thickness (by 1/16"s). The program increases the first dimension on the list that is not constrained by yard practice or structural considerations. This procedure continues until the bottom section modulus exceeds the deck section modulus; then the program goes back to increase the deck scantlings. This iterative procedure keeps the deck and bottom section moduli balanced until both exceed the required hull-girder section modulus, or until all constraints become binding. In this case, the program prints an error message that states the scantlings cannot be increased sufficiently to provide the required section modulus. The program then continues by incrementing the spacing of either the longitudinals or transverses and proceeding with the individual section moduli calculations.

The constraints used in this iterative procedure are as follows:

1. Web depth $> 2 \times$ flange width
2. Plate thickness $< 1 \frac{7}{16}$ "
3. Flange thickness $< (\text{web thickness} + \frac{1}{8})$ "
4. Flange width $< 32 \times$ web thickness

The first three constraints are based on reasonable yard

practices, and the last is used to conform with Jenkins (1977) to ensure flange material is effectively utilized.

The hull-girder section modulus routine has two special features that help ensure longitudinal scantlings are of reasonable dimensions and minimum weight. The first is that when either the deck or bottom plate is increased in thickness, the program calls the subroutine "smact1" to ensure the associated longitudinal stiffener has scantlings that reasonably match the plate to which it is attached.

The second feature is one that limits the deck plate thickness to " n "/16 inches greater than the bottom plate before any increases are made to the other two deck scantlings. The value of " n " is initially set to zero and is then increased incrementally. For each value of " n ", longitudinal member scantlings are calculated to satisfy the hull section modulus requirement. The resultant hull weight is then determined (as per Section 3.2.8), and " n " is allowed to increase until hull weight no longer decreases. This feature was heuristically determined to provide the minimum hull weight satisfying the required hull-girder section modulus for a specified longitudinal frame and transverse girder spacing.

3.2.8 Calculation of Hull Weight

Given the longitudinal member scantlings obtained from the previous hull-girder section modulus routine, the program calculates the resulting longitudinal steel cross-sectional

area. From this, the longitudinal weight per foot is determined simply by multiplying the cross-sectional area in square inches by the steel weight factor: 3.4 pounds per square inches-foot. To this must be added the longitudinal weight per foot contributed by bottom, side, deck, and longitudinal bulkhead transverse girders, as well as by the stanchions and transverse bulkhead plates with associated stiffeners. The details of how the weights per foot for these transverse members are determined are shown in Table 3.6. The hull weight is then simply found by multiplying the weight in pounds per longitudinal foot by the factor: $1/2240$, which converts it into weight in long tons. This weight could then be multiplied by a correlation factor which would account for the different scantlings used in the bow and stern areas as well as weld weights. However, at this point, it appears that the hull weight calculated is in reasonable agreement with the hull weights of single-skin tank barges currently in operation.

The program finishes the routine for a specific value of longitudinal and transverse girder spacing and "n" by storing these values and those of the hull weight and associated deck and bottom hull girder section moduli if the calculated hull weight is less than that found during previous iterations. Thus, at the end of all iterations with respect to longitudinal and transverse spacing, the minimum hull weight barge particulars will be available for outputting. It is al-

so at this point of the program that it is determined whether "n" should be incremented. This is done by examining whether the last increase in "n" had changed the calculated values of the deck and bottom hull-girder section moduli. If no change is observed, then iterations with respect to "n" are terminated and either the longitudinal or transverse spacing is incremented.

Finally, at the end of all iterations, the program outputs the values of hull weight, longitudinal and transverse girder spacing, deck and bottom section moduli, and "n" for the barge of the least hull weight found within the parametric ranges and increments of longitudinal and transverse girder spacing.

3.3 Calculation of Longitudinal Frame Scantlings (The Logic of "smact1")

As explained previously "smact1" calculates the scantlings of a longitudinal member given the input:

1. Plate thickness ("t")
2. Plate effective width as a flange ("s")
3. Required section modulus for the member ("smr")

The logical flow of how these scantlings are calculated is shown in Figure 3.4 and will be explained in detail in the following text.

First the subroutine assumes certain initial values for the dimensions of the scantlings. These are:

<u>Dimension</u>	<u>Initial Value</u>
Plate thickness	as specified in inputted value "t"
Plate depth	as specified in inputted value "s"
Web depth	3.5"
Web thickness	plate thickness, for plate thinner than 3/8"
	plate thickness less 1/8", for plate thicker than 3/8"
Flange width	the greater of:
	(1) web thickness
	(2) 3/4"
Flange thickness	plate thickness

These values were chosen as to be reasonable for the smallest allowed scantlings, based on available rolled sections.

The strategy of the subroutine is then to calculate the section modulus of the longitudinal with the above minimum scantlings. If the calculated section modulus is less than that required by the inputted value "smr", the program increments the scantling dimensions in a manner that will attempt to add the least amount of steel area while achieving the required section modulus. It was found from theoretical analysis and empirical observation that the best approach was to increase the dimensions of the scantlings in the order: (1) web depth (by 1/2"s), (2) flange width (by 1/2"s), (3) flange thickness (by 1/16"s), and (4) web thickness (by 1/16"s). The subroutine will increase the first dimension in

the above list that is not constrained by usual yard practice or structural considerations. The constraints based on usual shipyard practice were as follows:

1. $5 \times \text{flange width} > \text{web depth} > 2 \times \text{flange width}$
2. Flange thickness $<$ plate thickness
3. Web thickness $<$ plate thickness
4. Web depth $<$ 18"

Certainly, none of these constraints have to be rigidly maintained, but they do represent reasonable values. The constraints based on structural considerations were as follows:

1. Web depth less than 75 times web thickness to conform with Horne (1958) to ensure stability against web buckling
2. Flange width less than 32 times web thickness to conform with Jenkins (1977) to ensure flange width material is really effective in the section modulus.

After any scantling is increased, the subroutine recalculates the section modulus. If it exceeds the required section modulus, the subroutine returns the scantling dimensions to the main program. Otherwise, it repeats the above procedure until no dimension can be increased without violating a constraint. At that point an error message prints out that scantling limits have been exceeded in "smact1" and sends an indicator to the main program that this condition exists so appropriate action can be taken.

3.4 Calculation of Transverse Girder Scantlings (The Logic of "smact2")

As explained previously, subroutine "smact2" calculates the scantlings of a transverse girder given the input:

1. Plate thickness ("tp")
2. Plate effective width as a flange ("s")
3. Required section modulus for the member ("smr")
4. Depth of the associated longitudinal frame ("dep"),
i.e., depth of the deck longitudinal for a deck girder

The logical flow of how these scantlings are calculated is shown in Figure 3.5 and will be explained in detail in the following text.

First the subroutine assumes some initial values for the dimensions of the scantlings. Many of these initial values are the minimums allowed by ABS (1973) and (1978). The values and the associated references are presented in Table 3.7. The strategy of the program is similar to that described for "smact1". First the subroutine calculates the section modulus of the girder with the above minimum scantlings. If the calculated section modulus is less than that required by the inputted value "smr", then the program increments the scantling dimensions in a manner that will attempt to add the least amount of steel area while achieving the required section modulus. It was found from theoretical analysis and empirical observation of the program that the best approach was to in-

crease the dimensions of the scantlings in the order: (1) web depth (by 1"s), (2) flange width (by 1"s), (3) flange thickness (by 1/16"s), (4) and web thickness (by 1/16"s). The subroutine will increase the first dimension in the list that is not constrained by usual yard practice or structural considerations. The constraints based on usual shipyard practice were:

1. $5 \times \text{flange width} > \text{web depth} > 2 \times \text{flange width}$
2. Web depth less than 48" for bottom and deck girders
3. Web depth less than 80" for side and bulkhead girders
4. $\text{Web thickness} < (\text{plate thickness} + 1/2")$
5. $\text{Flange thickness} < (\text{web thickness} + 1/2")$
6. $\text{Flange thickness} < 1 \frac{7}{16}"$
7. $\text{Web thickness} < 1 \frac{7}{16}"$

Certainly, none of these constraints have to be rigidly maintained, but they do represent reasonable values as verified by conversations with shipyard personnel. The constraints based on structural considerations were:

1. $\text{Web depth} < 75 \times \text{web thickness}$, to conform with Horne (1958) to ensure stability against web buckling
2. $20 \times \text{web thickness} < \text{flange width} < 32 \times \text{web thickness}$, to conform with Jenkins (1977) to ensure flange width material is really effective in the section modulus.

After any scantling is increased, the subroutine recalculates the section modulus. If it exceeds the required

section modulus, the subroutine returns the scantling dimensions to the main program. Otherwise, it repeats the above procedure until no dimension can be increased without violating a constraint. At that point an error message is printed out that states scantling limits have been exceeded in subroutine "smact2" and the program sends an indicator to the main program that this condition exists so appropriate action can be taken.

3.5 Modification to Allow Hull Weight Calculation of Double-Bottom Tank Barges

The barge design program requires just a few simple modifications to allow for the calculation of the hull weight of double bottom barges. The mid-section plan of the double-bottom barge including the delineation of the dimensions is shown in Figure 3.2. It should be noted that "l4", h for stanchions, and "h1" differ from Figure 3.1 and that the depth of the double bottom ("dbdep") and the head to the inner bottom longitudinals ("h5") are added.

The program logic for double-bottom barges is identical to that for single-skin barges except for the following items:

1. Instead of calculating the scantlings of the bottom transverse girder which is not part of a double-bottom barge, the program calculates the depth of the double bottom in accordance with the rule presented in Table 3.8.

2. The values of "l4", "h2", and h for stanchions are based on the geometry of Figure 3.2 which differs from the

single-skin case of Figure 3.1.

3. After calculating the minimum scantlings required to satisfy the required section modulus for longitudinal bulkhead transverses, the program calculates values for the double-bottom plate thickness ("dbplt"), double-bottom side and center girder thicknesses ("siddbgirdth" and "cendbgirdth"), and head for double-bottom longitudinals ("h5") in accordance with the rules of ABS (1973) and (1978), presented in summary form in Table 3.8.

4. Following 3. above, the program determines the required section modulus for double-bottom longitudinal stiffeners in accordance with rules presented in Table 3.4. Following this, subroutine "smact1" is called using this required section modulus, "dbplt", and the effective plate width (calculated in a manner similar to the other scantlings) as input parameters. After this, the program proceeds to the routine that calculates hull-girder section moduli.

5. In the hull-girder section moduli calculation, proper account is taken of the longitudinal members of the double bottom, i.e., double bottom plate and longitudinals as well as center and side girders.

6. In the hull weight calculation, proper account is taken for the longitudinal members of the double bottom as well as the double-bottom floors which replace the bottom transverse girders of single-skin barges.

3.6 Barge Design Model Results

3.6.1 Input and Output Formats

To run the barge design model the simple input/output program "bargtest1", shown in Figure 3.6, was written. This program allows the user to specify values of the following input variables: (1)

1. Barge Principal Dimensions: length--"l", breadth--"b", depth--"d", draft--"t", and block coefficient--"cb" (2)

2. Range and Increment for Longitudinal Frame Spacings: "minlonspac", "maxlonspac", and "incrlonspac"

3. Range and Increment for Transverse Web Spacings: "minxvrspac", "maxxvrspac", and "incrxxvrspac"

4. Barge Internal Structure: "opt". For single-skin tank barges with a center longitudinal bulkhead, set "opt" to zero. For single-skin barges with two equally spaced longitudinal bulkheads and no stanchions, set "opt" to two. Or, for double-bottom barges with center longitudinal bulkheads, set "opt" to three.

5. Maximum "n" Value: "maxn". See section 3.2.7 for an explanation of the use of the variable "n".

(1) For more discussion concerning these variables see Section 3.2.1.

(2) If a depth of zero is inputted, the program "bargtest1" will calculate the minimum freeboard allowed by the rules specified in IMCO (1966) and add this amount to the draft to obtain the depth. See Figure 2.8 for the rules used.

6. Output Format: "cycle". For a detailed summary of the hull scantlings for the optimum barge, set "cycle" to zero. For an overall summary of each case of longitudinal and transverse spacing and "n" value considered, set "cycle" to one. Or, for an overall summary of only the optimum system, set "cycle" to two.

Examples of the output format types are shown in Figures 3.7-3.9 for a single-skin, center longitudinal bulkhead tank barge of dimensions 550'x85.94'x35.81' (LxBxT) (1) and block coefficient of 0.8. In these examples, longitudinal spacings of 1.00-3.25 feet in 3 inch increments and transverse spacings of 6-10 feet in one foot increments are considered. Figure 3.7 shows the input and the summary hullweight/section modulus output for the optimum barge ("cycle"=2). Figure 3.8 shows the detailed hull scantlings output for the same barge ("cycle"=0). And, figure 3.9 shows the complete summary hull weight/section modulus output for all the different transverse web spacings, longitudinal frame spacings, and "n" values considered ("cycle"=1).

3.6.2 Barge Design Model Output: Tabular and Graphical

The barge design model was run for barges of the follow-

(1) A barge depth of 44.98' was calculated by the program "bargtest1" to satisfy the minimum freeboard requirements specified in IMCO (1966).

ing size and form: (1)

Length:	250'-700' by 75' increments
Length-Breadth Ratio:	5.6-8.0 by 0.2 increments
Breadth-Draft Ratio:	2.2-3.0 by 0.2 increments
Block Coefficient	
Single-Skin:	0.75, 0.80, 0.85
Double-Bottom:	0.80

Tabular and graphical output of the hull steel weight for single-skin, center longitudinal bulkhead tank barges of the above size and form are presented in Tables 3.9-3.11 and Figures 3.10-3.12. Similarly, tabular output of the hull steel weight for double-bottom, center longitudinal bulkhead tank barges is presented in Table 3.12. The hull steel weight found was the minimum for barges with longitudinal spacings ranging from 1.00-3.25 feet in 3 inch increments and with transverse web spacings ranging from 5-16 feet in one foot increments.

3.6.3 Regressions Through the Barge Design Model Outputs

In order to reduce the cost of running the drop-and-swap model, regression equations rather than calls to the subprogram "bargdes" are used to estimate barge hull steel weight and length. These regressions, based on the single-skin and double-bottom tank barge runs described in the

(1) Barge depth was determined using the loadline rules specified in Figure 2.8.

previous section, are presented in Table 3.13. They estimate barge hull steel weight and deadweight as a function of barge length, length-breadth ratio (L/B), and breadth-draft ratio (B/T). They also estimate the barge hull steel weight and length as a function of barge deadweight, L/B , and B/T . These regressions are rather good, with "R-Squared" values of 0.999 or better. A comparison of the hullweight regression (function of length, L/B , and B/T) with the barge design model output data is shown in Figures 3.10-3.12. On the graphs, the regression estimates are shown by asterisks while the actual model output data is shown by the curves.

3.6.4 Some Comments on Barge Design Model Output

After examining the barge design model's output for the runs described in Section 3.6.2, the following general observations can be made:

1. Single-skin barges of less than 350' and all double-bottom barges have bottom hull section moduli significantly greater than the minimum value specified by the ABS rules. This is because the ABS rules pertaining to the section moduli of the individual bottom members result in very heavy bottom scantlings.

2. Generally, the optimum transverse and longitudinal spacings of longer barges are wider than for shorter barges.

3. Barges of greater than 650', especially with large depths, often result in member scantlings exceeding the

constraints specified in the model. Barges of these forms will probably need to have a tank-ship internal structure. That is, they will probably require two longitudinal bulkheads to be used to create side tanks with very deep, built-up webs.

4. The addition of a double-bottom increases the hull steel weight of single-skin tank barges of lengths 750', 625', 550', 475', 400', 325', and 250' approximately 18%, 14%, 8%, 6%, 4%, 1%, and 0%, respectively.

3.6.5 Model Validation

It was originally planned to validate the barge design model by comparing its estimates with the actual hull steel weight values of tank barges currently in operation. However, the detailed design data necessary to accomplish this validation could not be obtained. Although length, breadth, draft, deadweight, and hull steel weight values for several single-skin tank barges are available, the scantling length and block coefficient values are not. (1) Unfortunately, these two data elements are essential for validating and calibrating the model. Nevertheless, a partial validation can be made by selecting values for these variables that yield a cargo deadweight capacity equal to that of the barge. This

(1) The scantling length is the length used in classification society rules for the calculation of hull girder and member section moduli. This length depends on the type of linkage and depth of notch. See MARAD (1979) for a detailed explanation.

has been done for four tank barges ranging from 4,000 to 40,000 DWT as shown in Table 3.14. There it is seen that when the length between perpendiculars is used for the scantling length and the block coefficient is adjusted to yield the barge deadweight, then the model under/overestimates hull steel weight by 11/22%. And when a block coefficient of 0.90 is used and the hull scantling length is adjusted to yield the barge deadweight, then the model under/overestimates by 21/6%.

These results indicate that the barge design model will produce results of sufficient accuracy for the purpose of cost estimation in the drop-and-swap model. However, before its results can be trusted for preliminary design purposes, it should be calibrated and adjusted by comparing its hull steel weight estimates with the actual hull steel weights of barges for which all particulars are known including notch size, scantling length, block coefficient, and percentage lightweight, high strength steel used in its construction. (1)

(1) The barge design model assumes that the barge is totally fabricated out of mild steel. Therefore, if high strength, light weight steel is used, then the model's hull steel weight estimate must be reduced in proportion to the amount of mild steel replaced by the light weight steel.

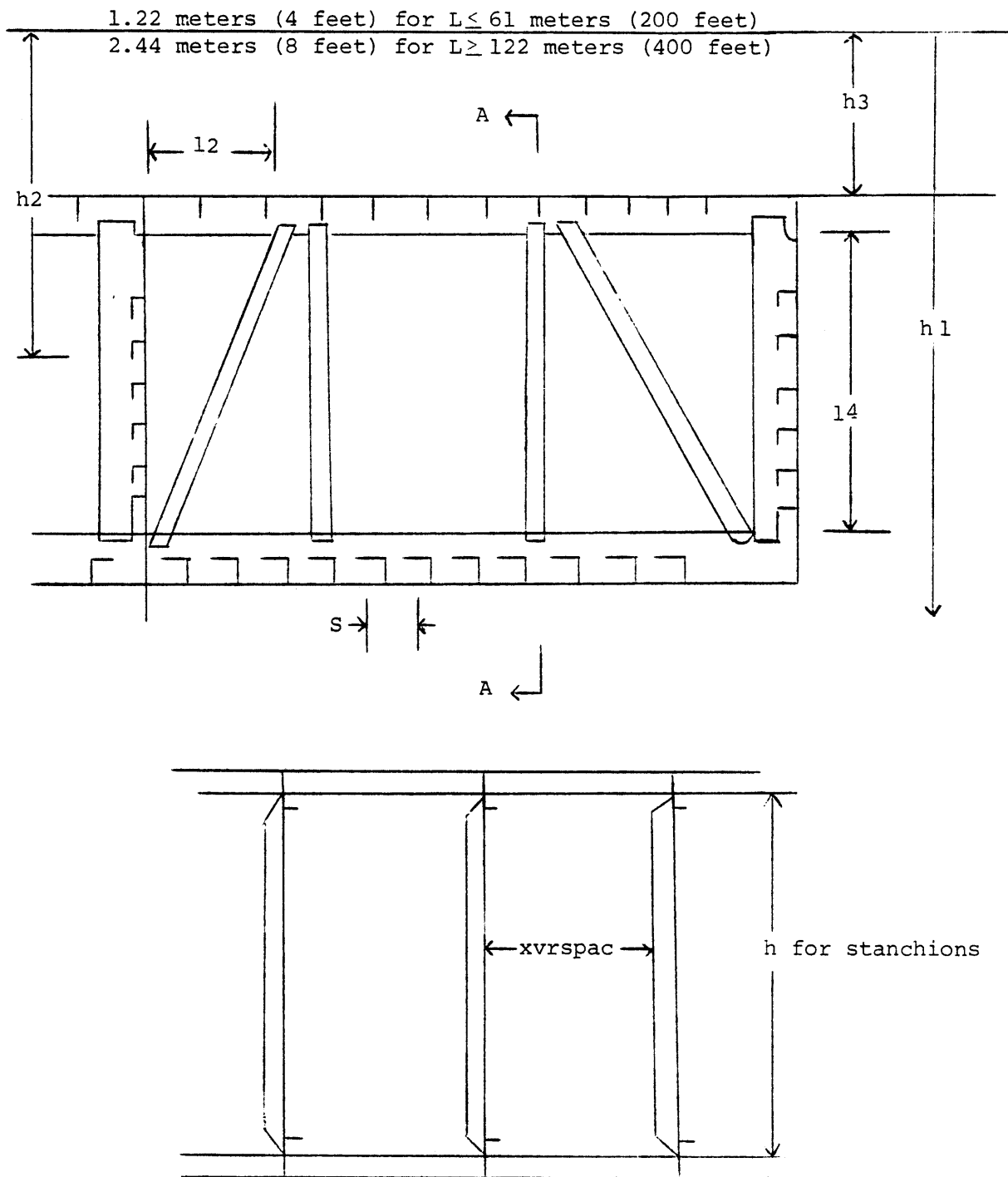


FIGURE 3.1

SCANTLING ARRANGEMENT FOR SINGLE-SKIN TANK BARGE

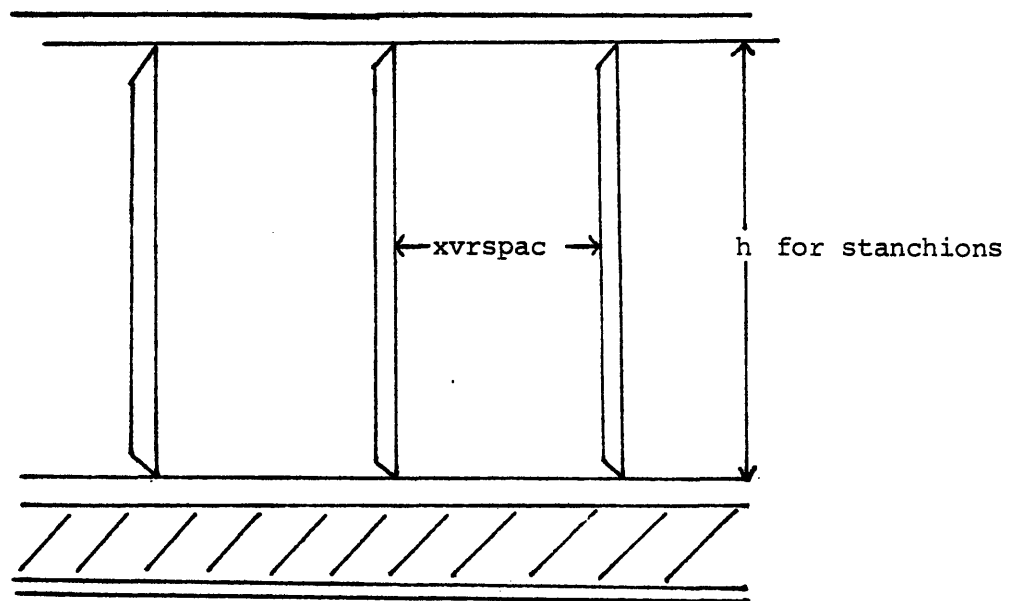
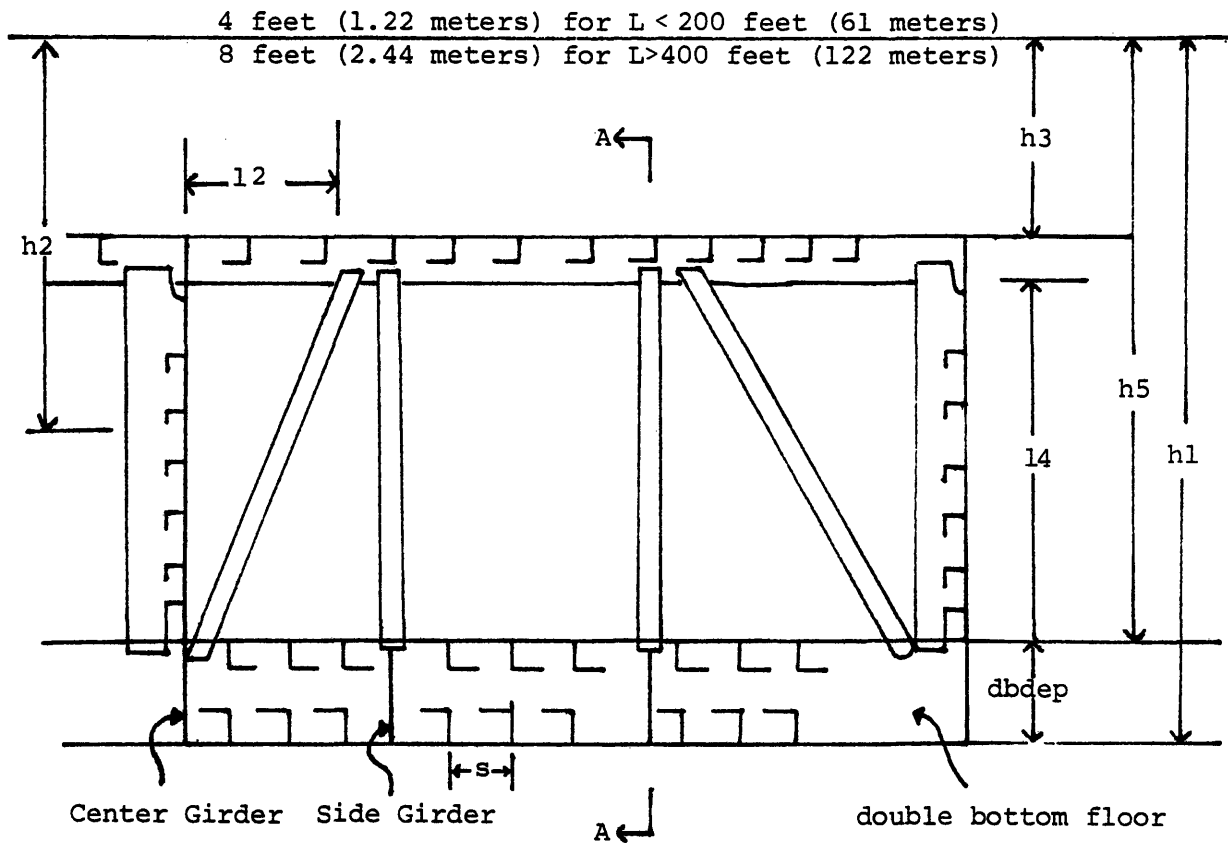


FIGURE 3.2

SCANTLING ARRANGEMENT FOR DOUBLE-BOTTOM TANK BARGE

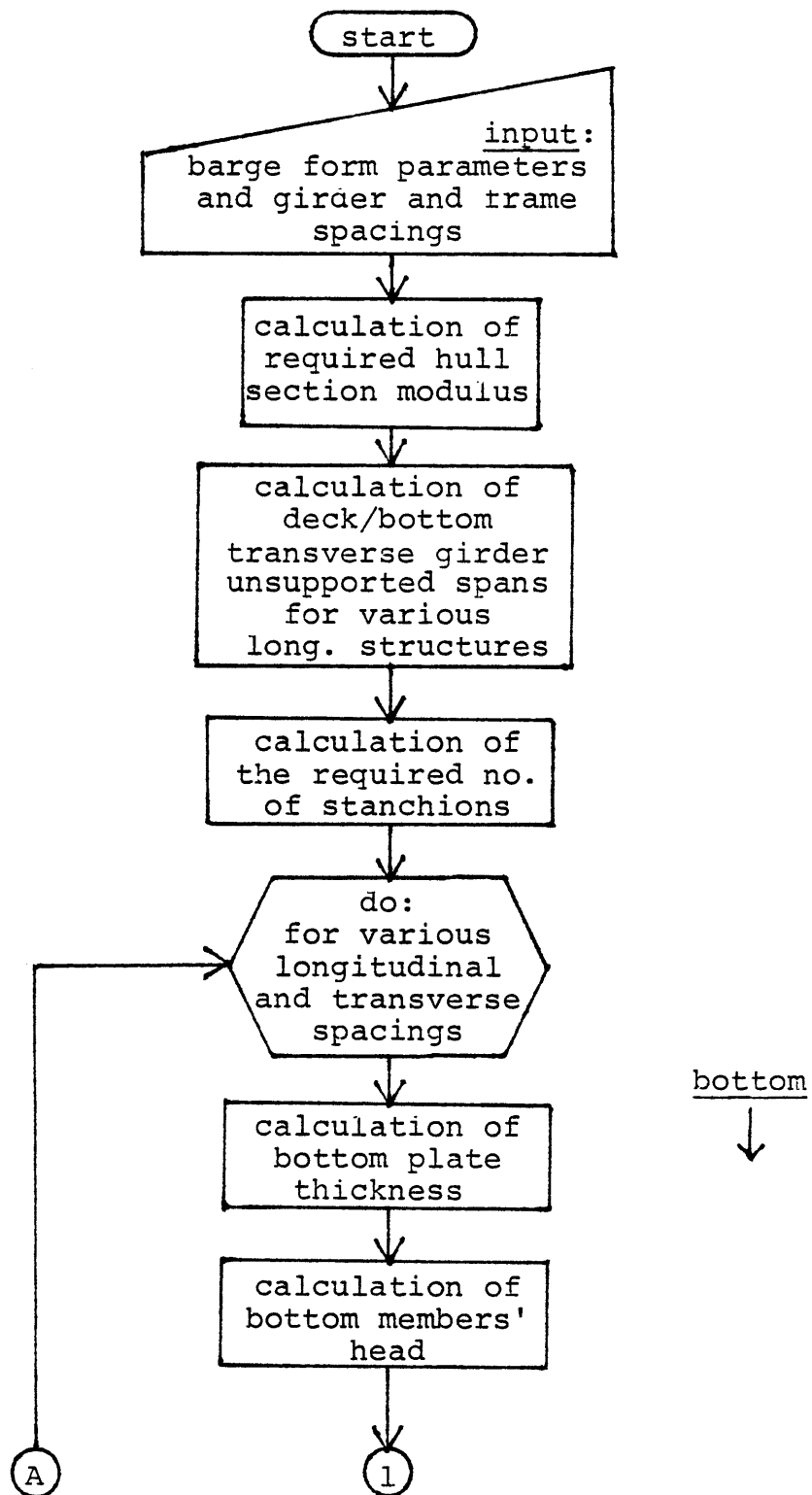


FIGURE 3.3
SUMMARY FLOWCHART OF BARGE DESIGN
SUBPROGRAM

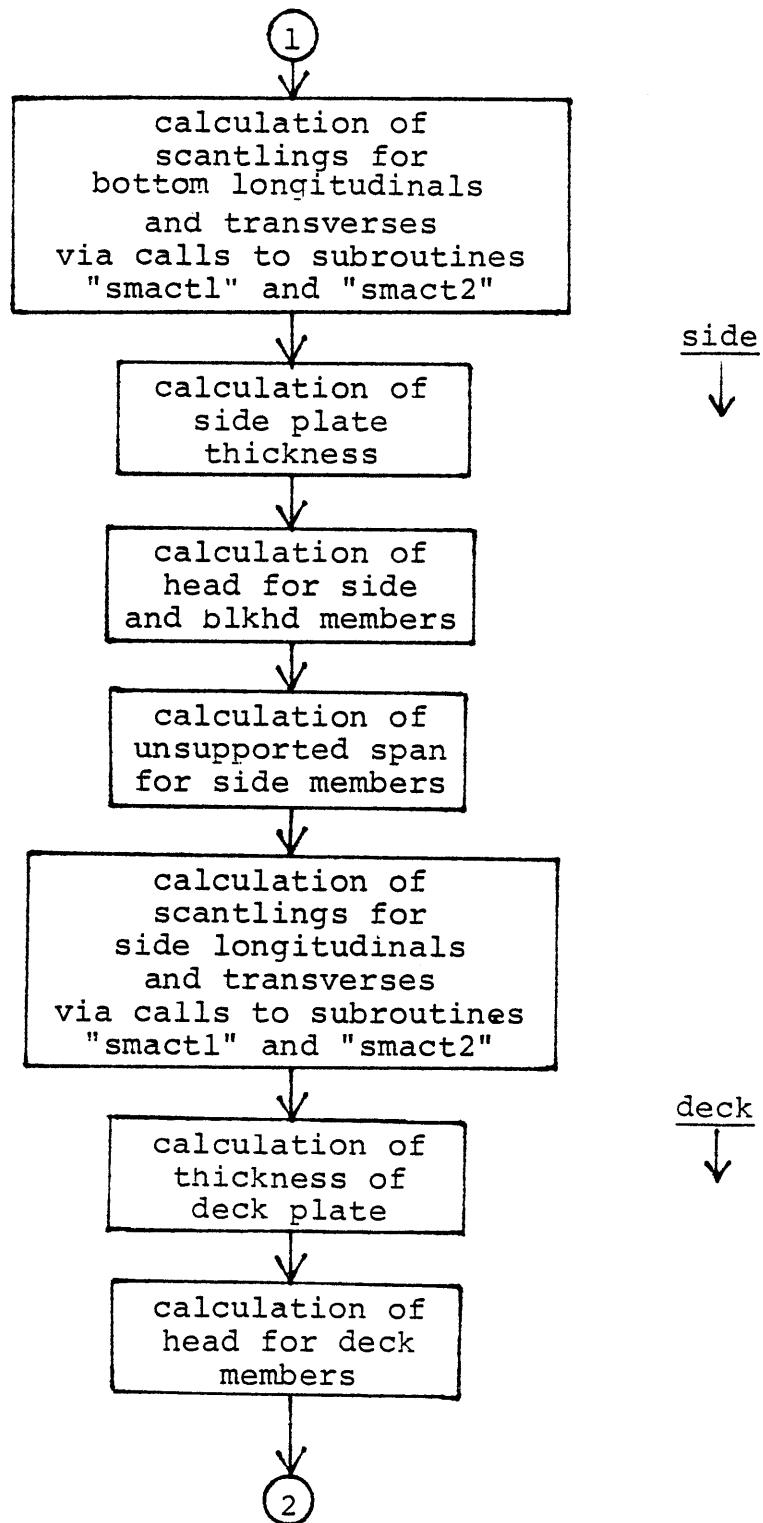


FIGURE 3.3... (Continued)

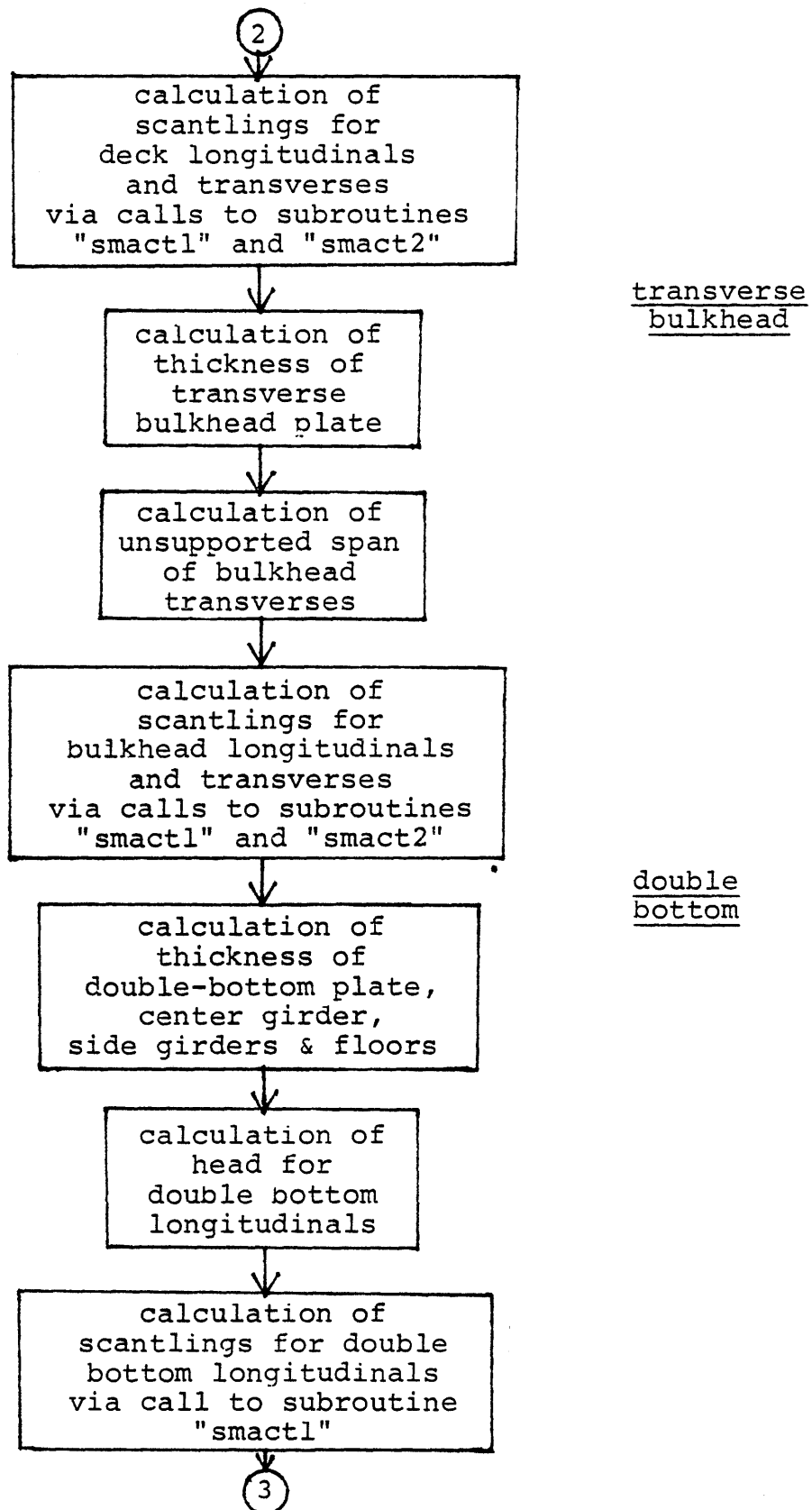


FIGURE 3.3... (Continued)

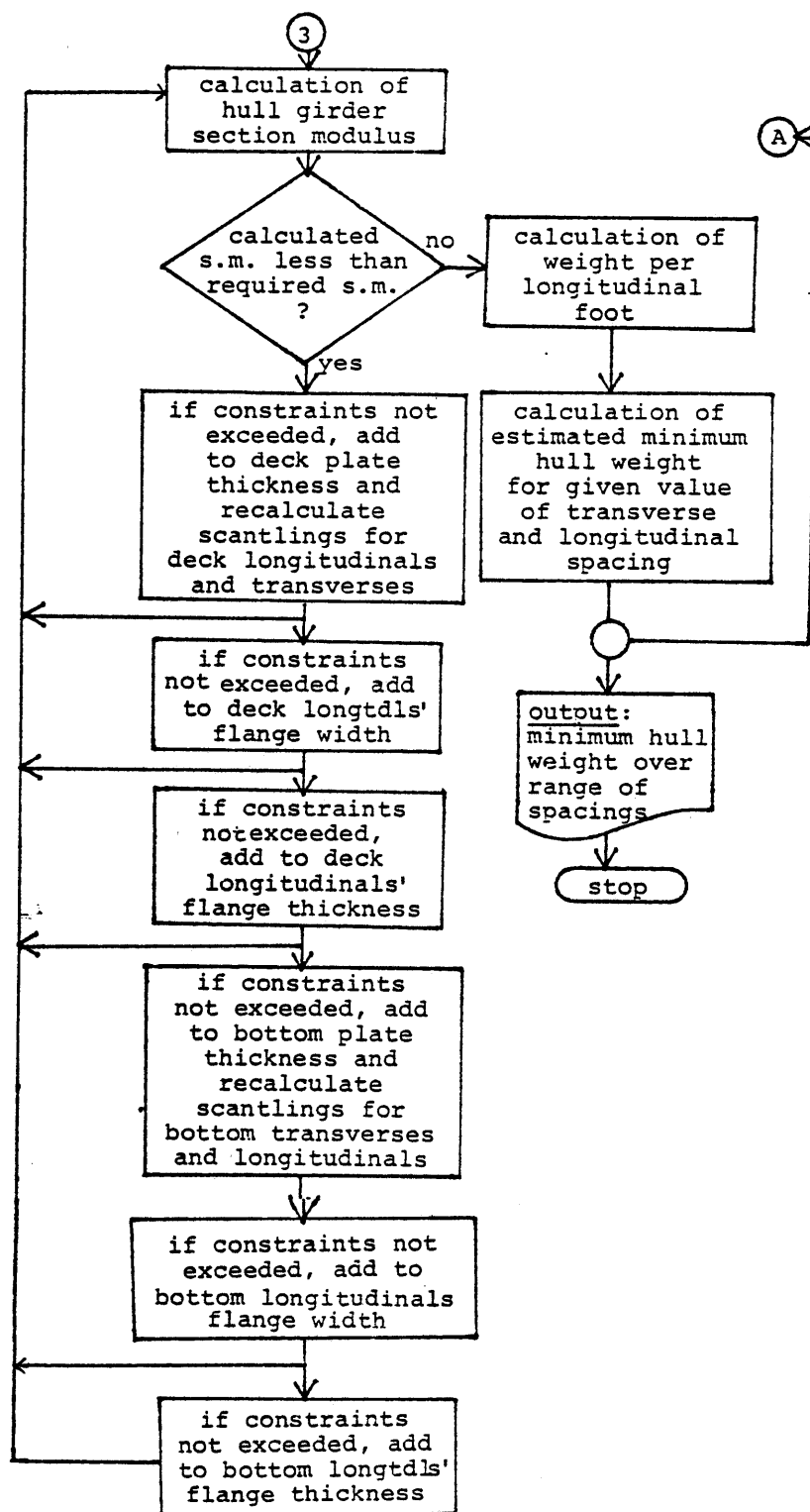


FIGURE 3.3... (Continued)

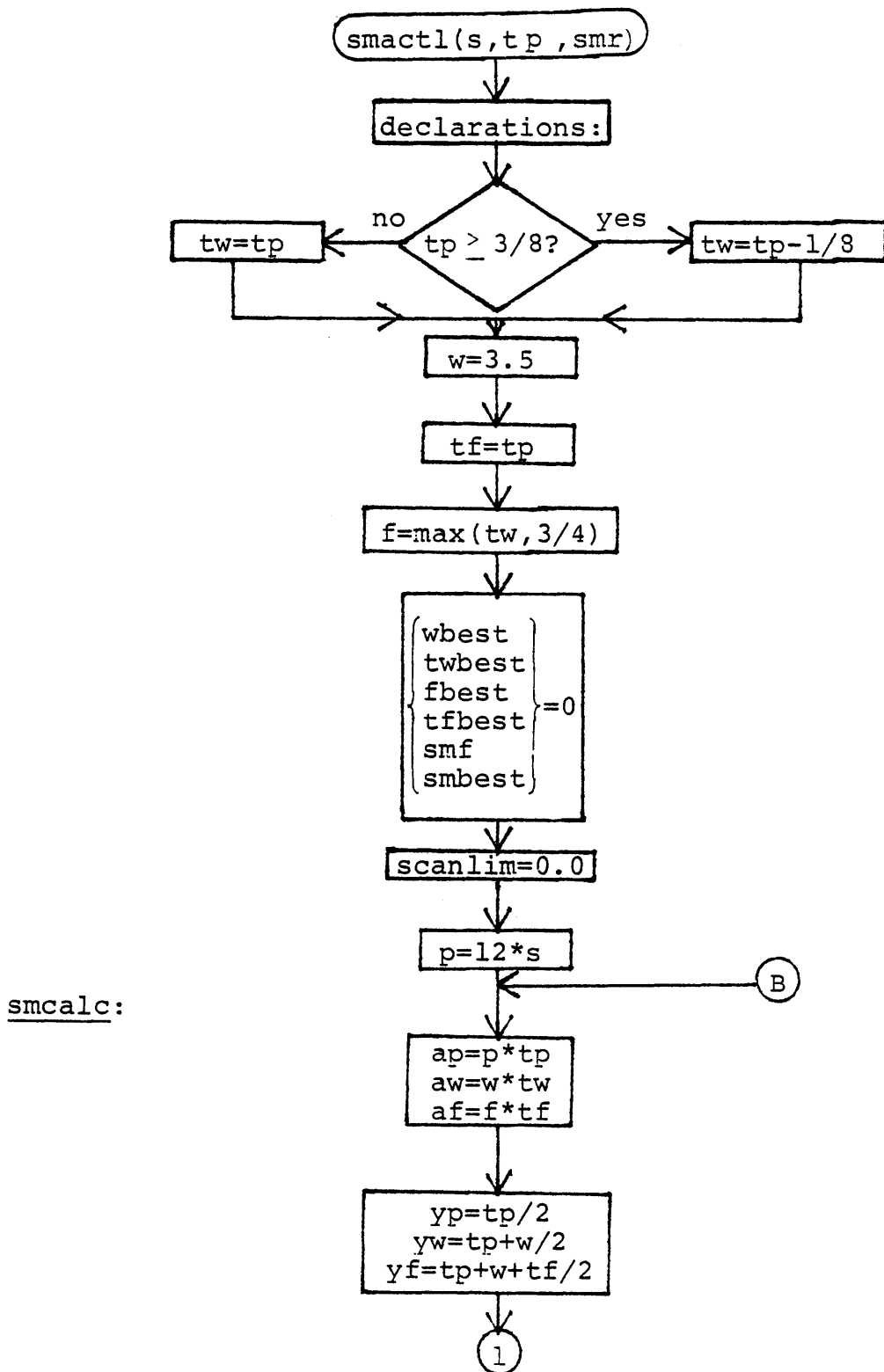


FIGURE 3.4

FLOWCHART OF SUBROUTINE "smact1"

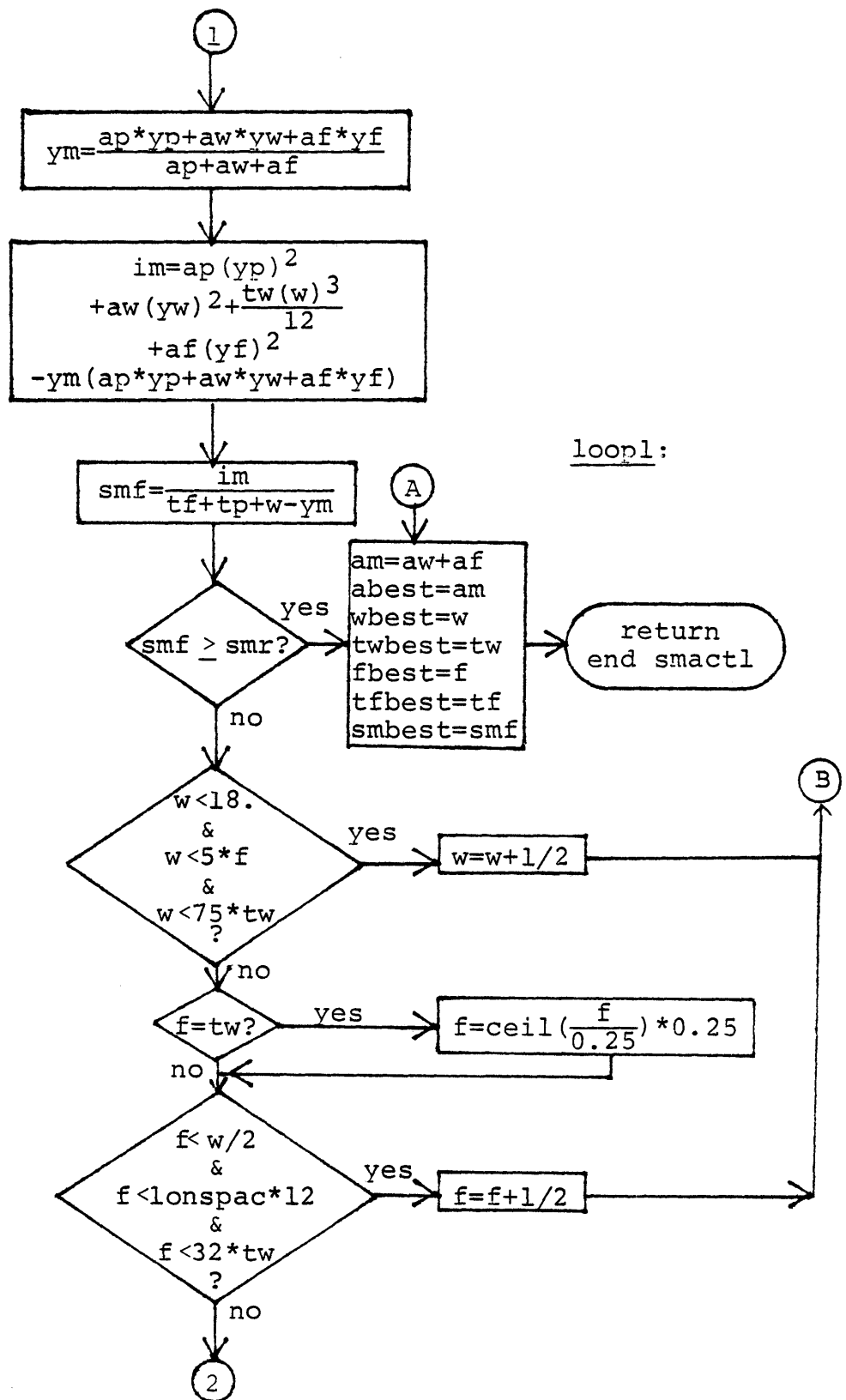


FIGURE 3.4... (Continued)

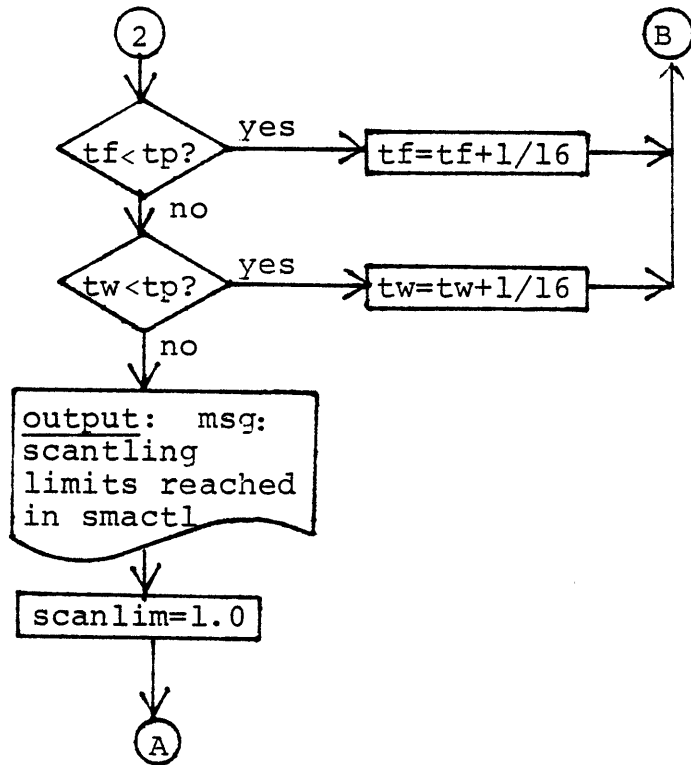


FIGURE 3.4...(Continued)

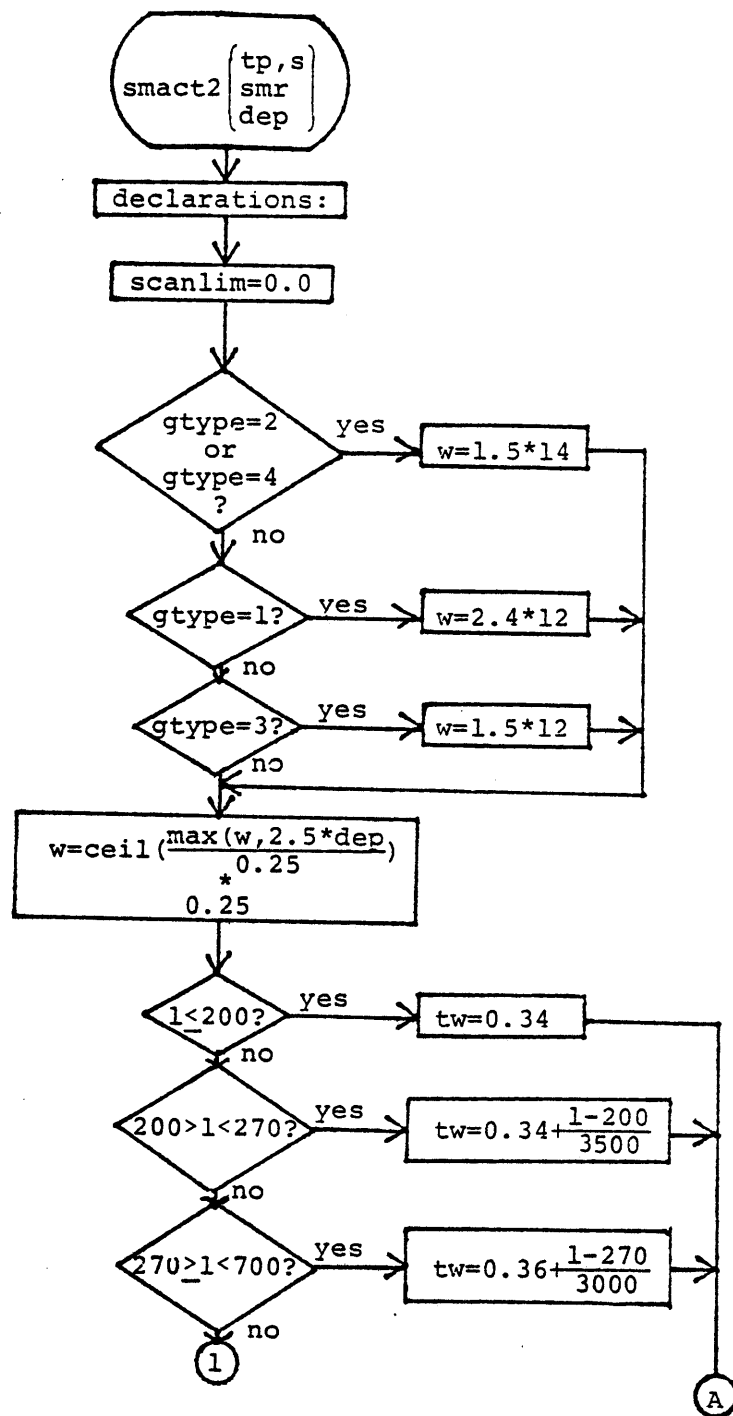


FIGURE 3.5
FLOWCHART OF SUBROUTINE "smact2"

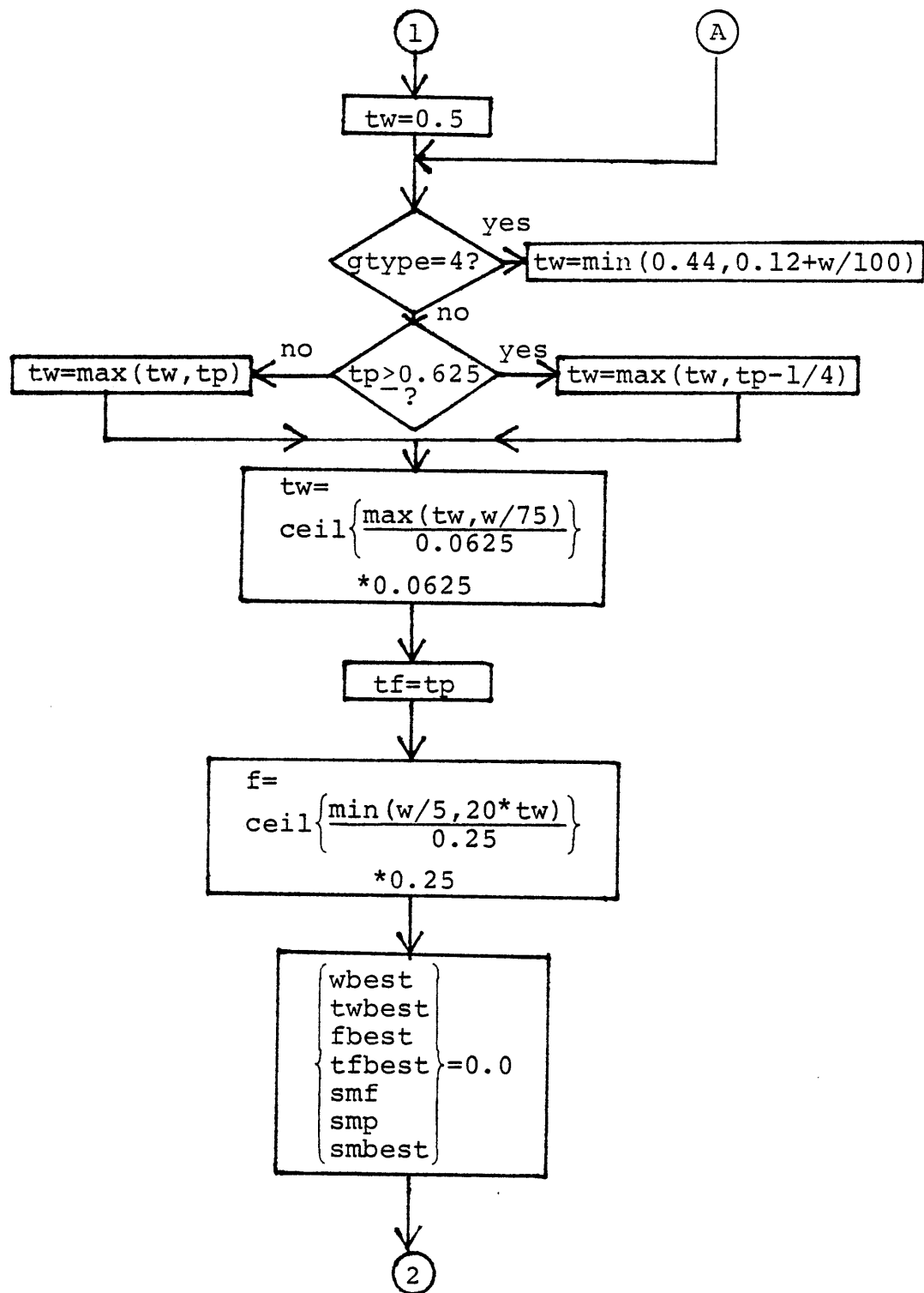
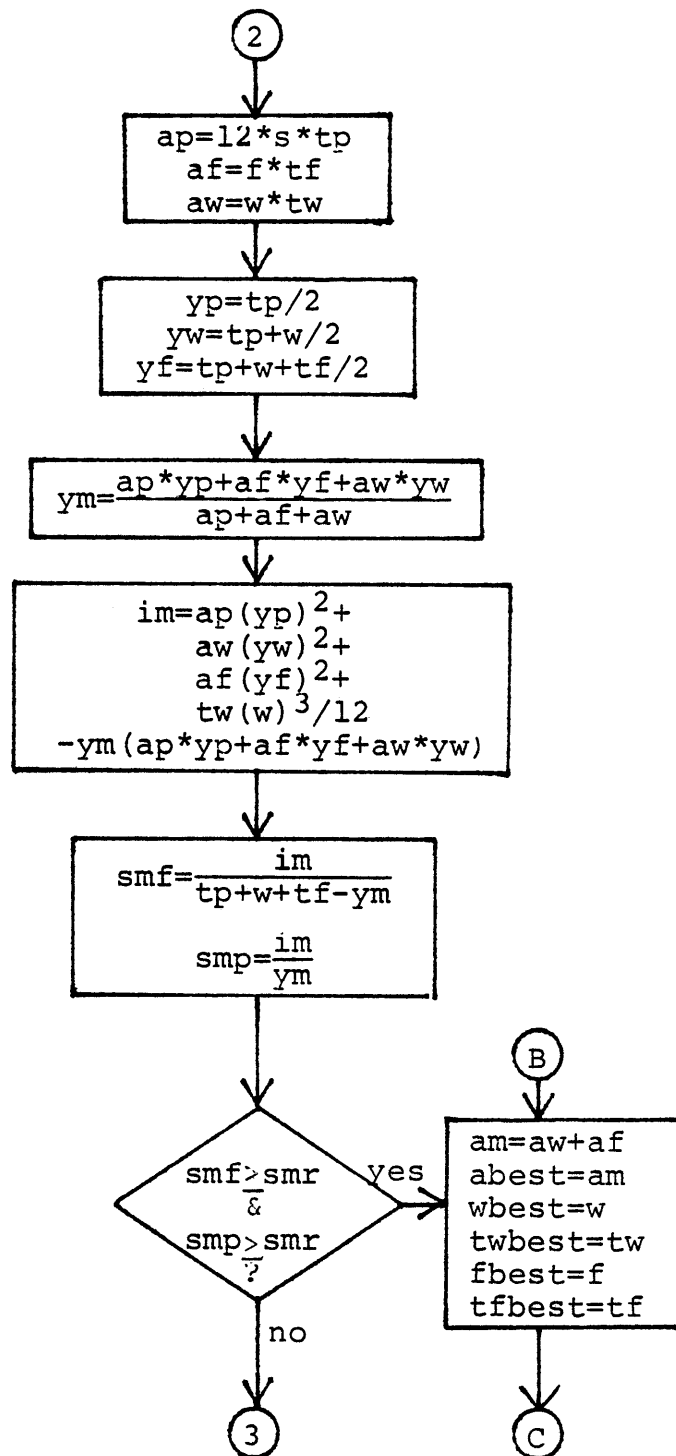


FIGURE 3.5... (Continued)

smcalc:



loop1:

FIGURE 3.5... (Continued)

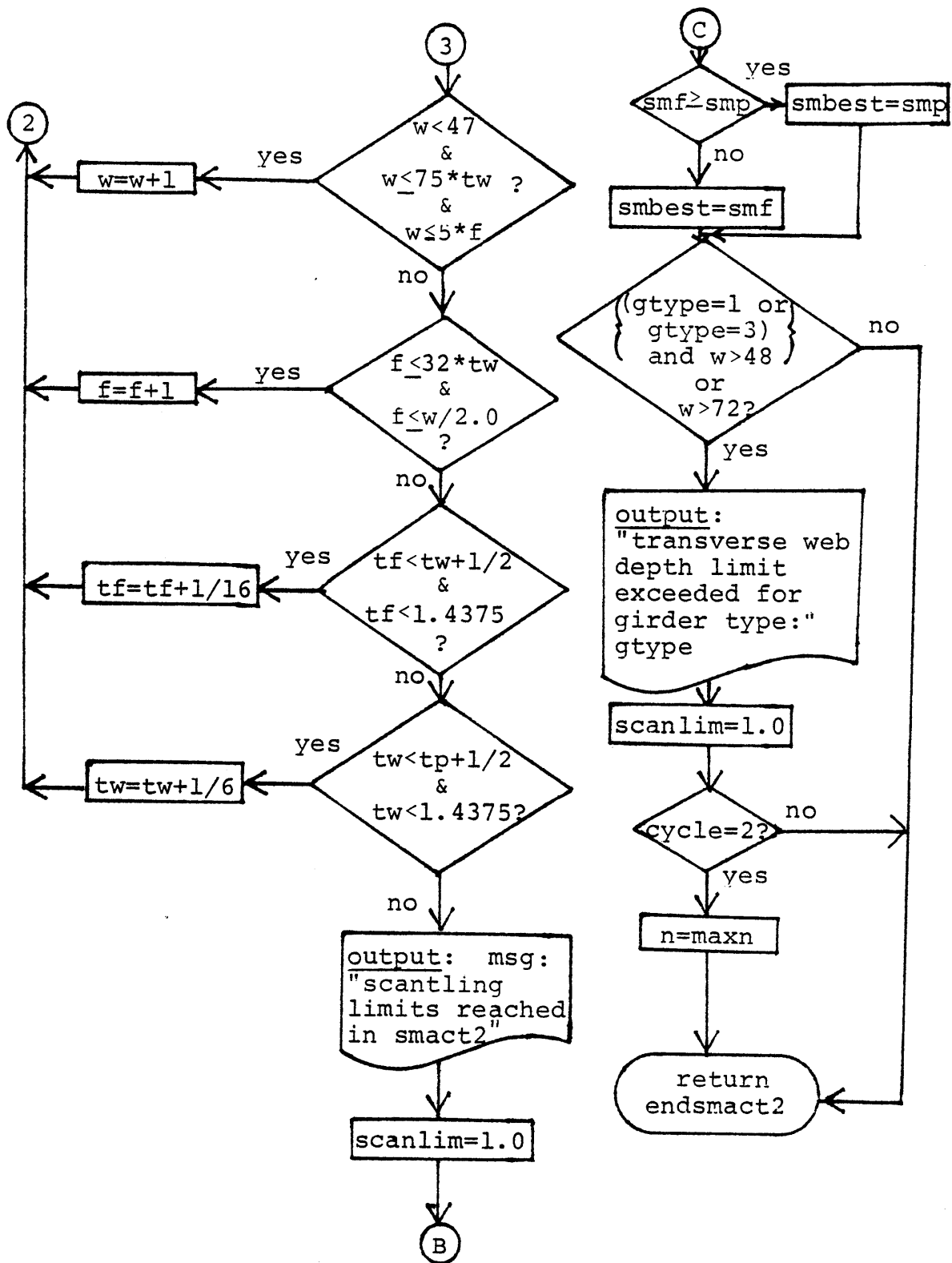


FIGURE 3.5... (Continued)

FIGURE 3.6

```

1 bargtest1: proc;
2   del(1,b,d,t,cb,minlonspac,maxlonspac,inclonspac,minxvrspac,maxxvrspac,incrxvrspac,coeff(66),llcoef(66)) dec float;
3   del (minfbd,fbd,ldfact,shrfact,minw,displ,vol,volwt,outfitwt,dwt,minwt,bestlonspac,bestxvrspac) dec float;
4   del (cycle,countr,opt,minn,maxn,bestn) bin fixed;
5   del (sysin,absfact) file stream input;
6   del sysprint file stream output;
7   del bargdes6 entry(dec float,dec float,dec float,dec float,dec float,dec float,dec float,dec float,dec float,dec float,dec float,dec float,bin fixed,bin fixed,dim(66) dec float,bin fixed,bin fixed,dec float,dec float,dec float,bin fixed) external;
8   open file(absfact) title("vfile_ abscoeff.data");
9   get file(absfact) list(coeff);
10  get file(absfact) list(llcoef);
11  close file(absfact);
12
13  return:
14    put skip(3) list("Please specify values for the following variables:");
15    put skip list("minlonspac, maxlonspac, inclonspac: ");
16    put skip;
17    get list(minlonspac,maxlonspac,inclonspac);
18    put skip list("minxvrspac, maxxvrspac, incrxvrspac: ");
19    put skip;
20    get list(minxvrspac,maxxvrspac,incrxvrspac);
21    put skip list("1,b,d,t,cb: ");
22    put skip;
23    get list(1,b,d,t,cb);
24    put skip list("maxn, cycle, opt:");
25    put skip;
26    get list(maxn,cycle,opt);
27    minn = 0;
28    put skip(5) list("LENGTH BFAM DEPTH DRAFT L/B B/T L/D T/D HULL-WT OUTFIT-WT LT-WT DISPL DWT FRBD VOLWT");
29    if l=750. then minfbd=llcoef(66);
30    else do;
31      countr=ceil((1-99.99999)/10.);
32      minfbd=llcoef(countr) + (llcoef(countr+1)-llcoef(countr))*(1-90.-10.*countr)/10.;
33    end;
34    minfbd = minfbd*(cb + 0.68)/1.38;
35    if (1/(minfbd/12. + t)) < 15. then do;
36      if l<393.6 then ldfact = (t + minfbd/12. - 1/15.)*(1/131.2)/(1-1/(12.*131.2));
37      else ldfact = (t + minfbd/12. - 1/15.)*4.;
38    end;
39    else ldfact = 0.;
40    shrfact = 0.03/45*1 + 3.75;
41    fbd = 0.75*(minfbd + ldfact + shrfact);
42    mind=(t + fbd/12.);
43    if d=0. then d=mind;
44    displ = 1*b*t*cb/35.;
45    call bargdes6(1,b,d,t,cb,minlonspac,maxlonspac,inclonspac,minxvrspac,maxxvrspac,incrxvrspac,minn,maxn,coeff,cycle,opt,minwt,bestlonspac,bestxvrspac,bestn);
46    vol = 1*b*t*cb + 1*b*(d-t) - (minwt/490.);
47    volwt = vol/42.;
48    outfitwt = ((400. + (1-290.)*1.7) -230.)*0.88;
49    dwt = displ -minwt - outfitwt;
50    put skip edit(1,b,d,t,l/b,b/t,l/d,t/d,minwt,outfitwt,minwt+outfitwt,displ,dwt,fbd,volwt)
51    (col(1),f(3),f(8,2),2 f(5,1),f(4,1),f(5,2),f(5,1),f(5,2),f(6),x(5),f(4),x(1),f(7),2 f(6),f(6,2),f(6));
52  if cycle = 1 ; cycle=2 then do;
53    put page;
54    go to return;
55  end;
56  end bargtest1;

```

bargtest1

Please specify values for the following variables:

minlonspac, maxlonspac, incrlonspac:

1.00,3.25,0.25,

minxvrspac, maxxvrspac, incrxvrspac:

5.,12.,1.,

l,b,d,t,cb:

550.0,85.9375,0.,35.8073,0.80,

maxn, cycle, opt:

7,2,0,

LENGTH BEAM DEPTH DRAFT L/B B/T L/D T/D HULL-WT OUTFIT-WT LT-WT DISPL DWT FRBD VOLWT
550 85.94 45.0 35.8 6.4 2.40 12.2 0.80 5016 539 5555 38685 33130110.08 42560
\\014

FIGURE 3.7

SAMPLE INPUT/OUTPUT OF THE BARGE DESIGN MODEL: SUMMARY OUTPUT FOR THE OPTIMUM BARGE

FIGURE 3.8

bargtest1

Please specify values for the following variables:

minlonspac, maxlonspac, incrlonspac:
2.25,2.25,1.,

minxvrspac, maxxvrspac, incrxvrspac:
9.,9.,1.,

l,b,d,t,cb:
550.0,85.9375,0.,35.8073,0.80,

maxn, cycle, opt:
2,0,0,

LENGTH BEAM DEPTH DRAFT L/B B/T L/D T/D HULL-WT OUTFIT-WT LT-WT DISPL DWT FRBD VOLWT
length=550 beam= 85.94 depth=44.98 draft=35.81 required modulus= 51167

longitudinal frame spacing = 2.25 transverse web spacing = 9.00
bottom modulus= 54031 deck modulus= 51549 estimated bare hull weight= 5016

summary of principal scantlings

bottom shell - 0.6875 in. side shell - 0.6250 in. bulkhead plating - 0.5625 in. deck plate - 0.8125
bottom longitudinals: flange- 2.75 x 0.6875; web- 13.50 x 0.5625
bottom transverses: flange- 7.25 x 0.6875; web- 35.50 x 0.5000
side longitudinals: flange- 2.25 x 0.6250; web- 10.00 x 0.5000
side transverses: flange- 26.75 x 0.8750; web- 58.50 x 0.8125
bulkhead longitudinals: flange- 2.25 x 0.5625; web- 9.50 x 0.4375
bulkhead transverses: flange- 26.25 x 0.8750; web- 60.50 x 0.8125
deck longitudinals: flange- 1.25 x 0.8125; web- 4.00 x 0.6875
deck transverses: flange- 4.00 x 0.8125; web- 19.25 x 0.5625
bestn= 2 smreq= 5.116718750e+004 bestlonspac= 2.250000000e+000 bestxvrspac= 9.000000000e+000
550 85.94 45.0 35.8 6.4 2.40 12.2 0.80 5016 bestmodbot= 5.403096746e+004 bestmoddk= 5.154864626e+004;
539 5555 38685 33130110.08 42560

bargtest1

Please specify values for the following variables:

minlonspac, maxlonspac, incllonspac:
1.50,2.75,0.25,

minxvspac, maxxvspac, inclxvspac:
7.,10.,1.,

l,b,d,t,cb:
550.0,85.9375,0.,35.8073,0.80,

maxn, cycle, opt:
5,1,0,

LENGTH BEAM DEPTH DRAFT L/B B/T L/D T/D HULL-WT OUTFIT-WT LT-WT DISPL DWT FRBD VOLWT
length=550 beam= 85.94 depth=44.98 draft=35.81 required modulus= 51167

longitudinal frame spacing = 1.50 transverse web spacing = 7.00
bottom modulus= 58896 deck modulus= 52220 estimated bare hull weight= 5324

longitudinal frame spacing = 1.50 transverse web spacing = 7.00
bottom modulus= 55734 deck modulus= 51815 estimated bare hull weight= 5233

longitudinal frame spacing = 1.50 transverse web spacing = 7.00
bottom modulus= 52528 deck modulus= 51352 estimated bare hull weight= 5163

longitudinal frame spacing = 1.50 transverse web spacing = 7.00
bottom modulus= 52528 deck modulus= 51352 estimated bare hull weight= 5163

longitudinal frame spacing = 1.50 transverse web spacing = 8.00
bottom modulus= 60871 deck modulus= 52506 estimated bare hull weight= 5298

longitudinal frame spacing = 1.50 transverse web spacing = 8.00
bottom modulus= 57763 deck modulus= 52100 estimated bare hull weight= 5220

longitudinal frame spacing = 1.50 transverse web spacing = 8.00
bottom modulus= 54181 deck modulus= 51617 estimated bare hull weight= 5145

longitudinal frame spacing = 1.50 transverse web spacing = 8.00
bottom modulus= 54181 deck modulus= 51617 estimated bare hull weight= 5145

longitudinal frame spacing = 1.50 transverse web spacing = 9.00
bottom modulus= 59456 deck modulus= 51568 estimated bare hull weight= 5301

longitudinal frame spacing = 1.50 transverse web spacing = 9.00
bottom modulus= 56953 deck modulus= 51885 estimated bare hull weight= 5245

longitudinal frame spacing = 1.50 transverse web spacing = 9.00
bottom modulus= 53991 deck modulus= 51402 estimated bare hull weight= 5177

longitudinal frame spacing = 1.50 transverse web spacing = 9.00
bottom modulus= 53917 deck modulus= 51170 estimated bare hull weight= 5187

longitudinal frame spacing = 1.50 transverse web spacing = 9.00
bottom modulus= 53917 deck modulus= 51170 estimated bare hull weight= 5187

longitudinal frame spacing = 1.50 transverse web spacing = 10.00
bottom modulus= 61198 deck modulus= 51819 estimated bare hull weight= 5343

EXAMPLE OF COMPLETE SUMMARY OUTPUT FROM
BARGE DESIGN SUBPROGRAM

FIGURE 3.9

FIGURE 3.9... (Continued)

longitudinal frame spacing = 2.25 transverse web spacing = 8.00
 bottom modulus= 52976 deck modulus= 51630 estimated bare hull weight= 5114
 longitudinal frame spacing = 2.25 transverse web spacing = 9.00
 bottom modulus= 60320 deck modulus= 52519 estimated bare hull weight= 5169
 longitudinal frame spacing = 2.25 transverse web spacing = 9.00
 bottom modulus= 57936 deck modulus= 52039 estimated bare hull weight= 5086
 longitudinal frame spacing = 2.25 transverse web spacing = 9.00
 bottom modulus= 54031 deck modulus= 51549 estimated bare hull weight= 5016
 longitudinal frame spacing = 2.25 transverse web spacing = 9.00
 bottom modulus= 54031 deck modulus= 51549 estimated bare hull weight= 5016
 longitudinal frame spacing = 2.25 transverse web spacing = 10.00
 bottom modulus= 58471 deck modulus= 51298 estimated bare hull weight= 5250
 longitudinal frame spacing = 2.25 transverse web spacing = 10.00
 bottom modulus= 55071 deck modulus= 51276 estimated bare hull weight= 5200
 longitudinal frame spacing = 2.25 transverse web spacing = 10.00
 bottom modulus= 55071 deck modulus= 51276 estimated bare hull weight= 5200
 longitudinal frame spacing = 2.50 transverse web spacing = 7.00
 bottom modulus= 56693 deck modulus= 51494 estimated bare hull weight= 5299
 longitudinal frame spacing = 2.50 transverse web spacing = 7.00
 bottom modulus= 53806 deck modulus= 51551 estimated bare hull weight= 5220
 longitudinal frame spacing = 2.50 transverse web spacing = 7.00
 bottom modulus= 53806 deck modulus= 51551 estimated bare hull weight= 5220
 longitudinal frame spacing = 2.50 transverse web spacing = 8.00
 bottom modulus= 58193 deck modulus= 52170 estimated bare hull weight= 5219
 longitudinal frame spacing = 2.50 transverse web spacing = 8.00
 bottom modulus= 55350 deck modulus= 51702 estimated bare hull weight= 5140
 longitudinal frame spacing = 2.50 transverse web spacing = 8.00
 bottom modulus= 52240 deck modulus= 51678 estimated bare hull weight= 5080
 longitudinal frame spacing = 2.50 transverse web spacing = 8.00
 bottom modulus= 52568 deck modulus= 53466 estimated bare hull weight= 5132
 longitudinal frame spacing = 2.50 transverse web spacing = 8.00
 bottom modulus= 52568 deck modulus= 53466 estimated bare hull weight= 5132
 longitudinal frame spacing = 2.50 transverse web spacing = 9.00
 bottom modulus= 59735 deck modulus= 52807 estimated bare hull weight= 5191
 longitudinal frame spacing = 2.50 transverse web spacing = 9.00
 bottom modulus= 56874 deck modulus= 52358 estimated bare hull weight= 5116
 longitudinal frame spacing = 2.50 transverse web spacing = 9.00
 bottom modulus= 53535 deck modulus= 51799 estimated bare hull weight= 5042
 longitudinal frame spacing = 2.50 transverse web spacing = 9.00
 bottom modulus= 53535 deck modulus= 51799 estimated bare hull weight= 5042
 longitudinal frame spacing = 2.50 transverse web spacing = 10.00
 bottom modulus= 57730 deck modulus= 51366 estimated bare hull weight= 5130
 longitudinal frame spacing = 2.50 transverse web spacing = 10.00
 bottom modulus= 54725 deck modulus= 51417 estimated bare hull weight= 5076

longitudinal frame spacing = 2.50 transverse web spacing = 10.00
 bottom modulus= 54976 deck modulus= 52444 estimated bare hull weight= 5111
 longitudinal frame spacing = 2.50 transverse web spacing = 10.00
 bottom modulus= 54976 deck modulus= 52444 estimated bare hull weight= 5111
 longitudinal frame spacing = 2.75 transverse web spacing = 7.00
 bottom modulus= 56646 deck modulus= 51234 estimated bare hull weight= 5325
 longitudinal frame spacing = 2.75 transverse web spacing = 7.00
 bottom modulus= 53737 deck modulus= 51237 estimated bare hull weight= 5245
 longitudinal frame spacing = 2.75 transverse web spacing = 7.00
 bottom modulus= 53737 deck modulus= 51237 estimated bare hull weight= 5245
 longitudinal frame spacing = 2.75 transverse web spacing = 8.00
 bottom modulus= 57897 deck modulus= 51889 estimated bare hull weight= 5218
 longitudinal frame spacing = 2.75 transverse web spacing = 8.00
 bottom modulus= 54807 deck modulus= 51405 estimated bare hull weight= 5136
 longitudinal frame spacing = 2.75 transverse web spacing = 8.00
 bottom modulus= 52031 deck modulus= 51375 estimated bare hull weight= 5086
 longitudinal frame spacing = 2.75 transverse web spacing = 8.00
 bottom modulus= 52458 deck modulus= 53721 estimated bare hull weight= 5151
 longitudinal frame spacing = 2.75 transverse web spacing = 8.00
 bottom modulus= 52458 deck modulus= 53721 estimated bare hull weight= 5151
 longitudinal frame spacing = 2.75 transverse web spacing = 9.00
 bottom modulus= 59115 deck modulus= 51281 estimated bare hull weight= 5178
 longitudinal frame spacing = 2.75 transverse web spacing = 9.00
 bottom modulus= 56185 deck modulus= 52003 estimated bare hull weight= 5127
 longitudinal frame spacing = 2.75 transverse web spacing = 9.00
 bottom modulus= 53270 deck modulus= 51498 estimated bare hull weight= 5081
 longitudinal frame spacing = 2.75 transverse web spacing = 9.00
 bottom modulus= 53270 deck modulus= 51498 estimated bare hull weight= 5081
 longitudinal frame spacing = 2.75 transverse web spacing = 10.00
 bottom modulus= 57320 deck modulus= 51356 estimated bare hull weight= 5110
 longitudinal frame spacing = 2.75 transverse web spacing = 10.00
 bottom modulus= 57320 deck modulus= 51356 estimated bare hull weight= 5110 bestlonspac= 2.250000000e+000 bestxvrspac= 9.000000000e+000 bestn= 2 smreq= 5.116718750e+004 bestmodbot= 5.403096746e+004 bestmoddk= 5.154864626e+004;
 550 85.94 45.0 35.8 6.4 2.40 12.2 0.80 5016 539 5555 38685 33130110.08 42560
 1014

FIGURE 3.9... (Continued)

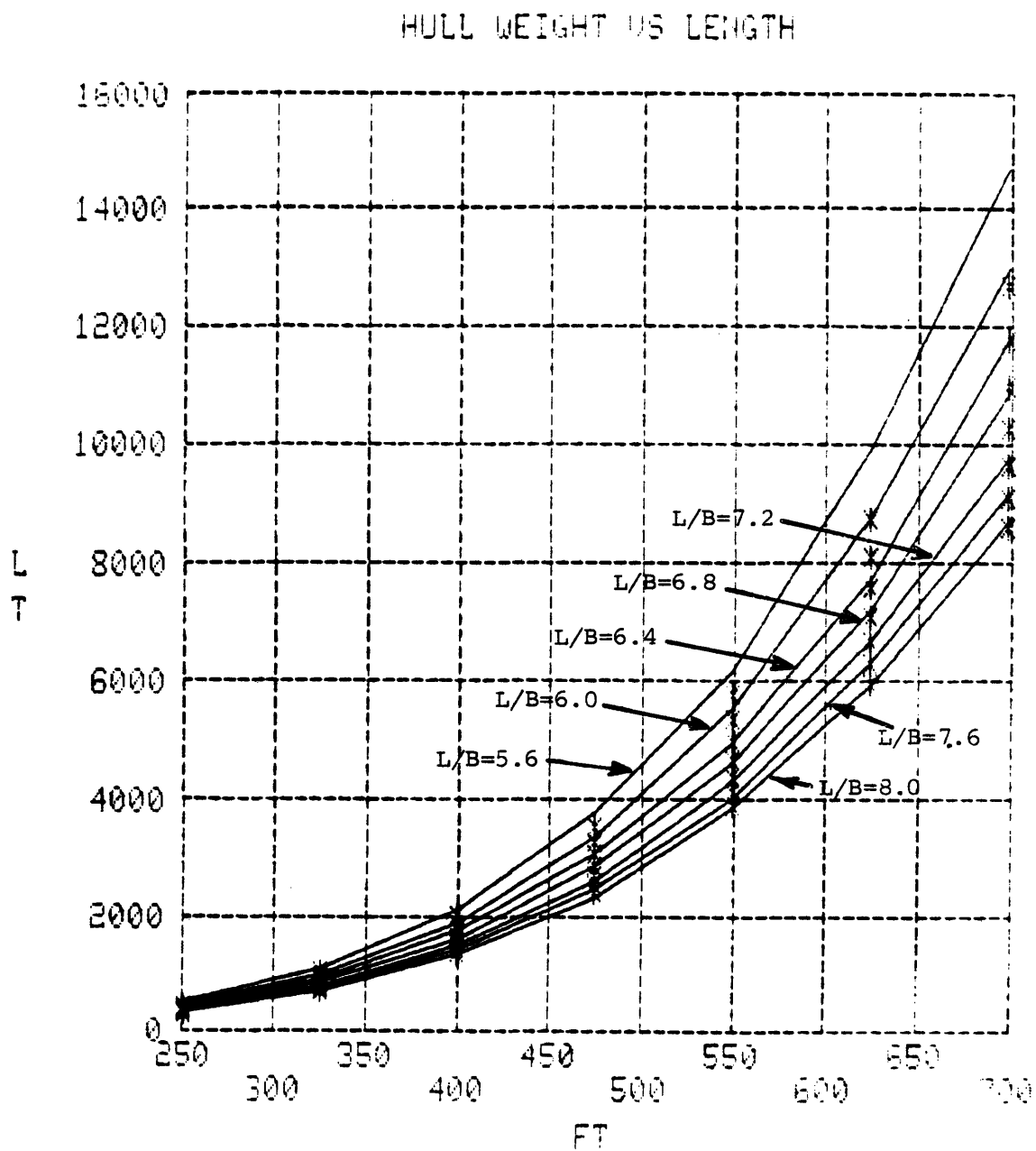


FIGURE 3.10
 PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH
 FOR $C_B=0.75$ AND $B/T=2.2$

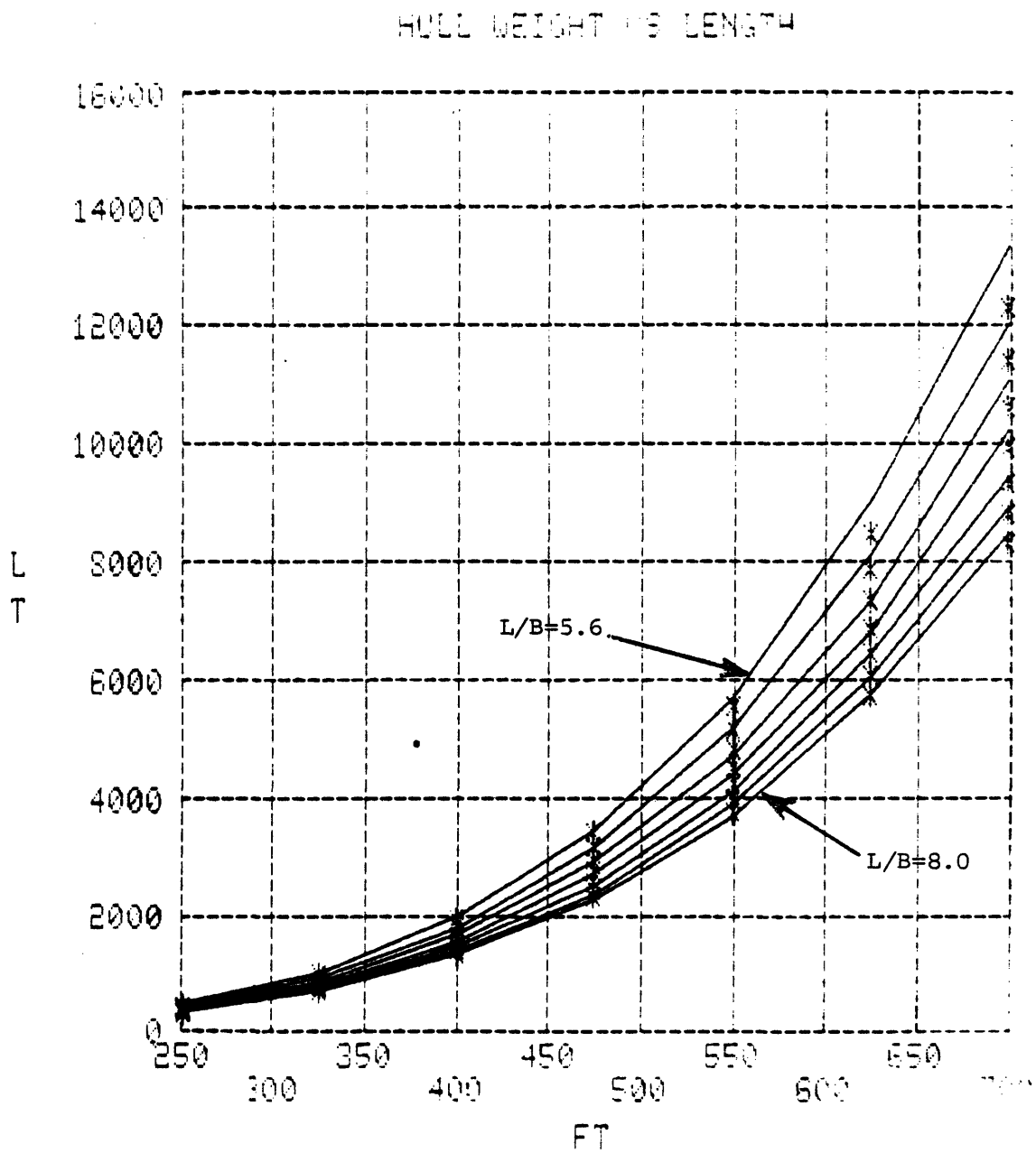


FIGURE 3.10

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH
FOR $C_B=0.75$ AND $B/T=2.4$

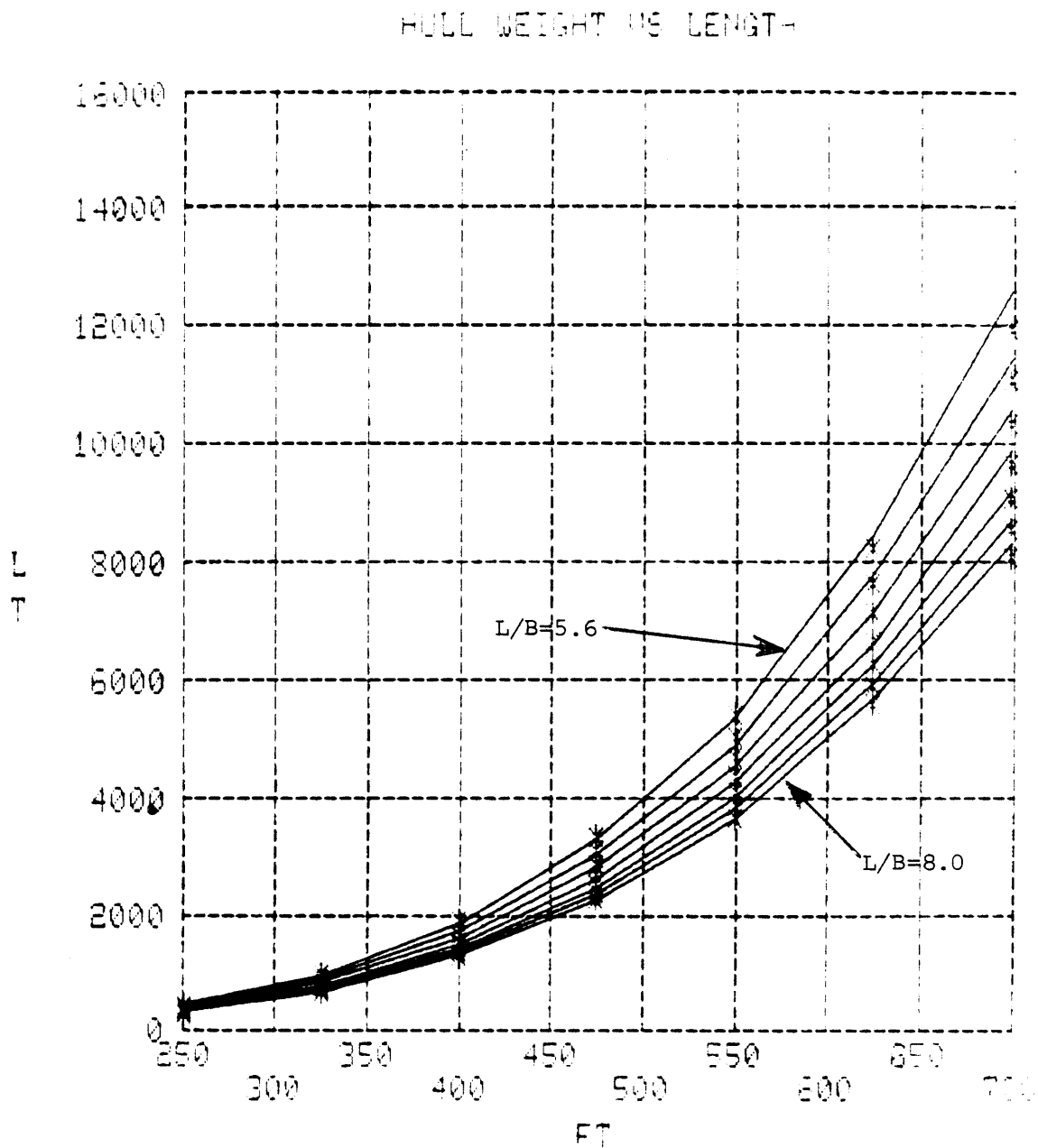


FIGURE 3.10
 PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH
 FOR $C_B=0.75$ AND $B/T=2.6$

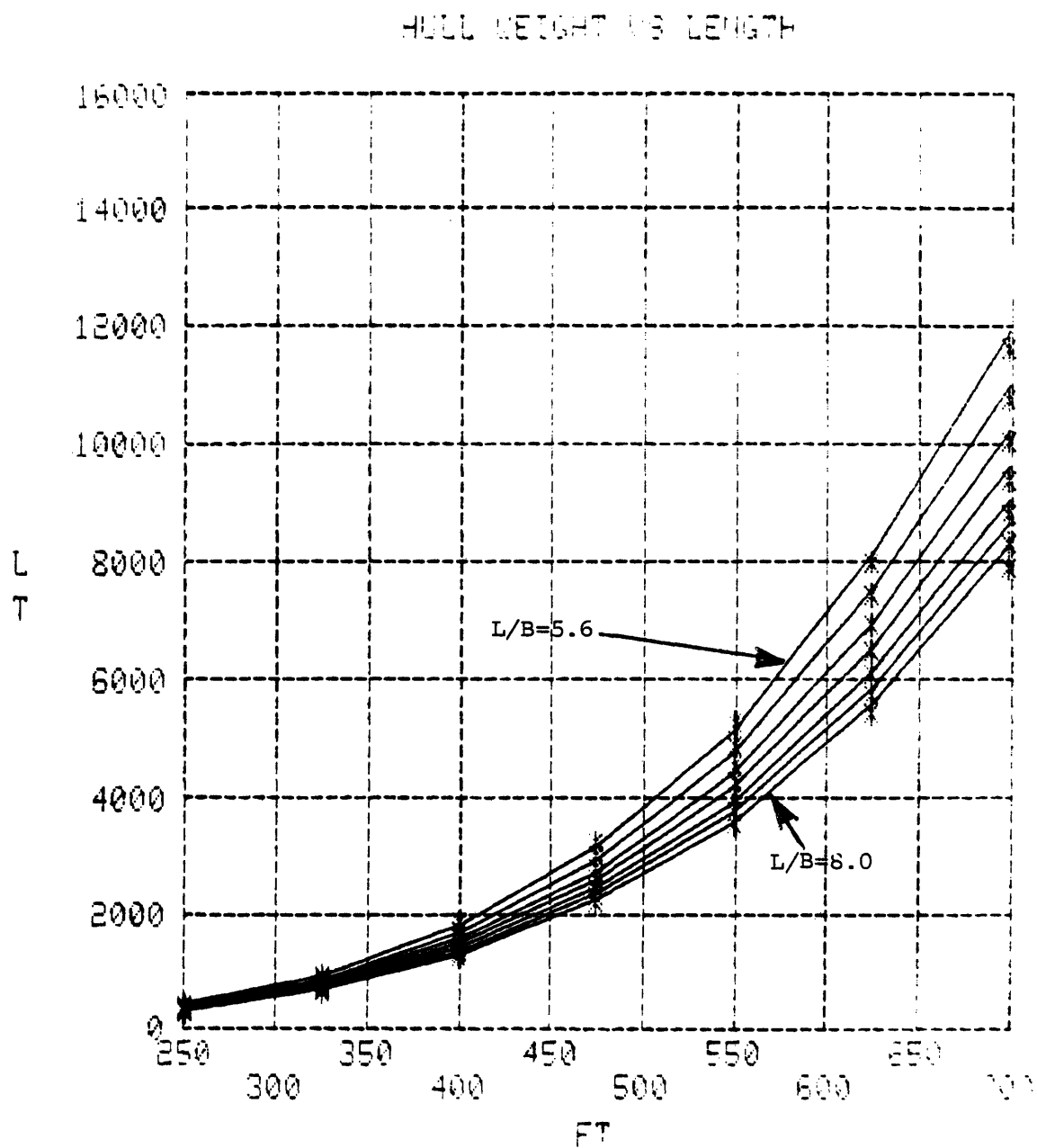


FIGURE 3.10
 PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH
 FOR $C_B=0.75$ AND $B/T=2.8$

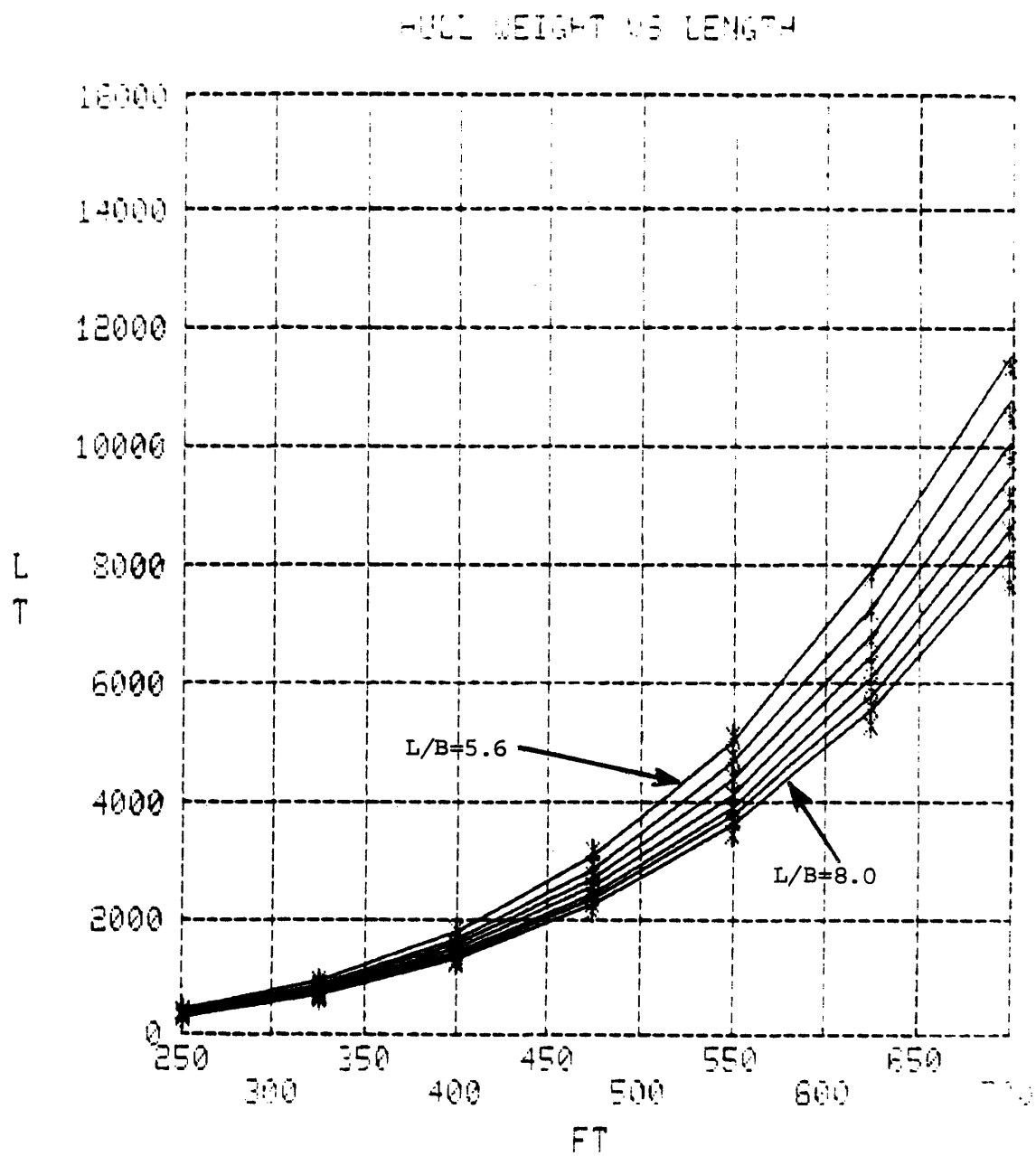


FIGURE 3.10
 PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH
 FOR $C_B=0.75$ AND $B/T=3.0$

HULL WEIGHT VS. LENGTH

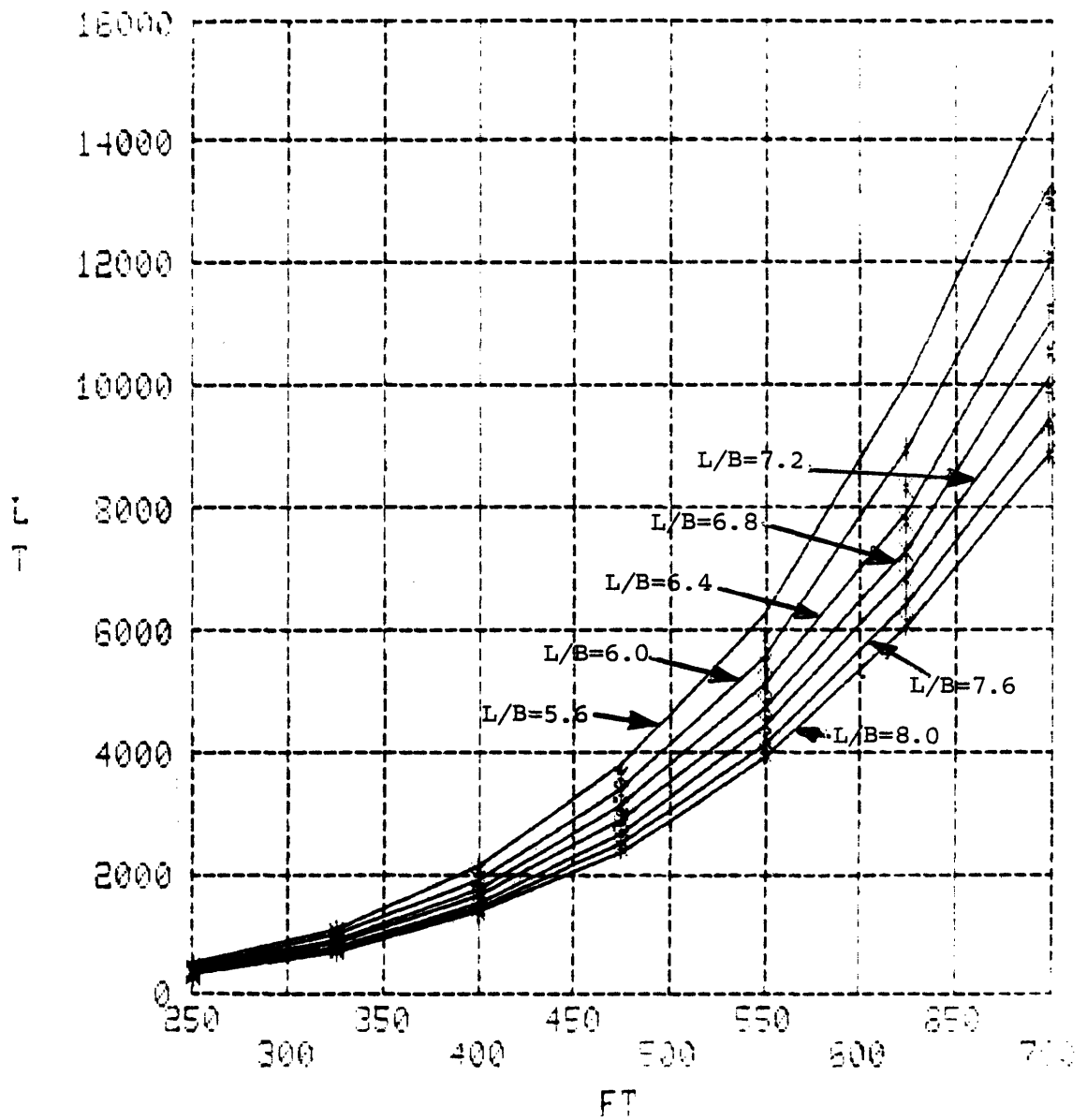


FIGURE 3.11

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH
FOR $C_B=0.80$ AND $B/T=2.2$

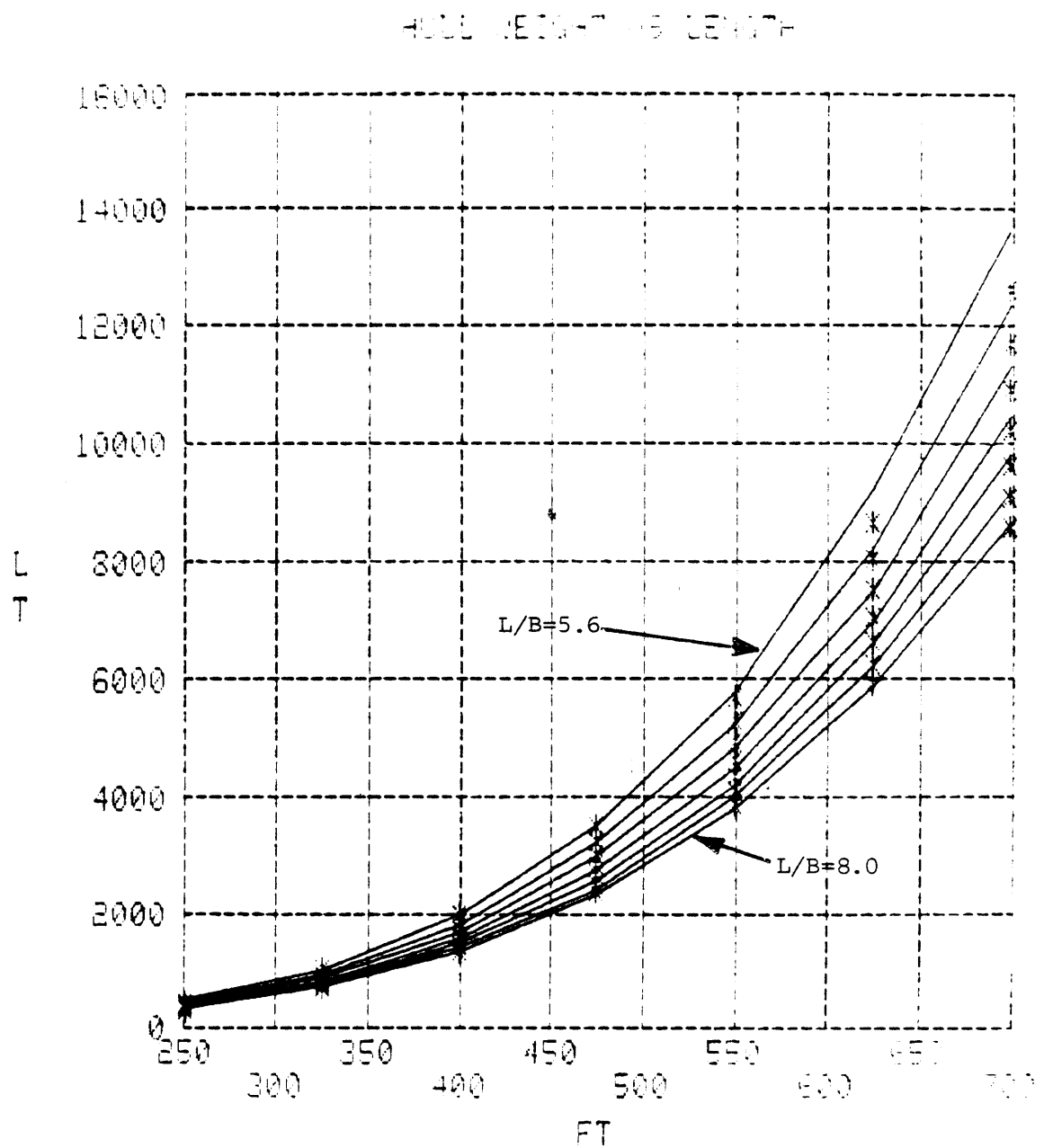


FIGURE 3.11
 PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH
 FOR $C_B=0.80$ AND $B/T=2.4$

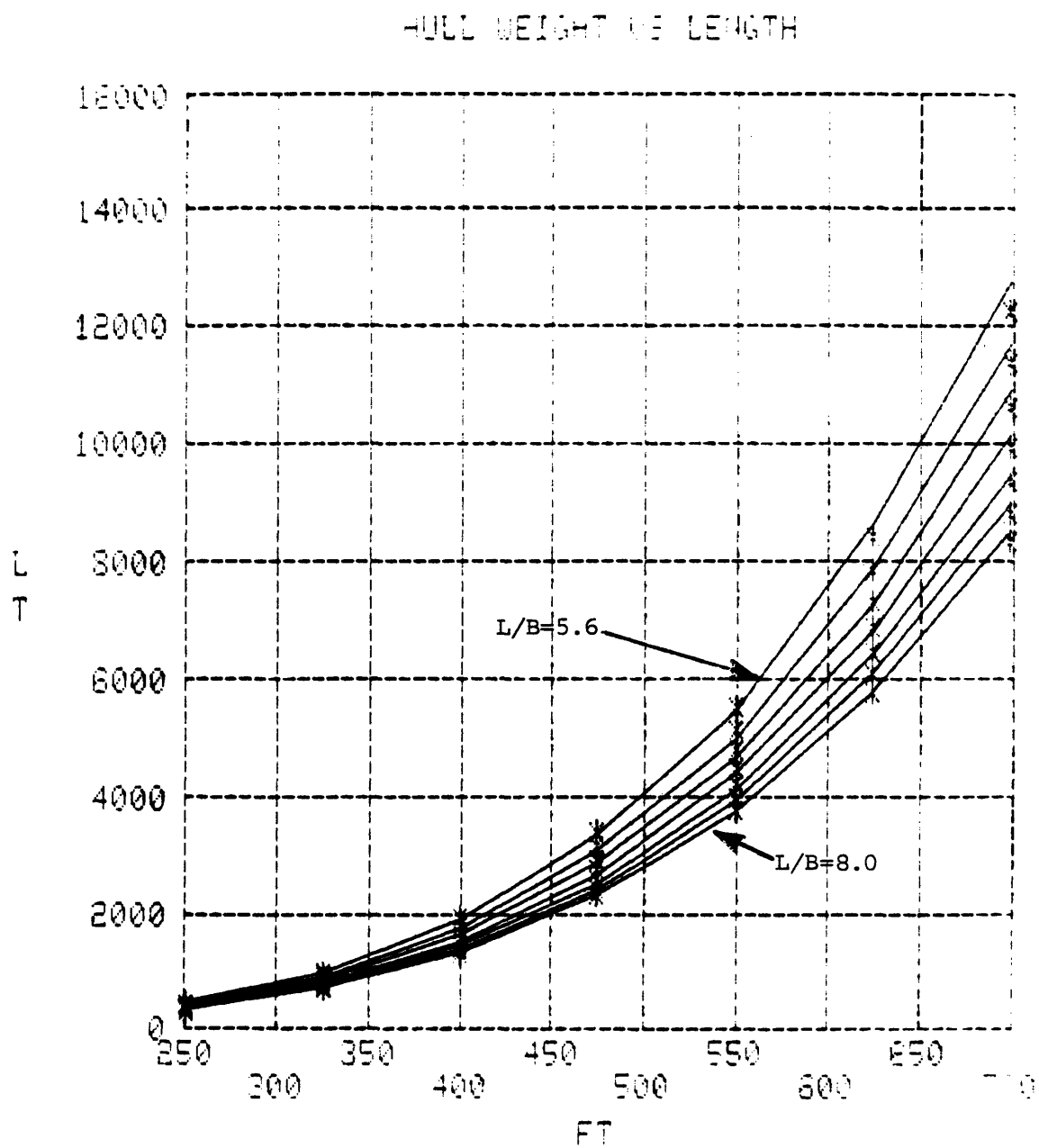


FIGURE 3.11
PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH
FOR $C_B=0.80$ AND $B/T=2.6$

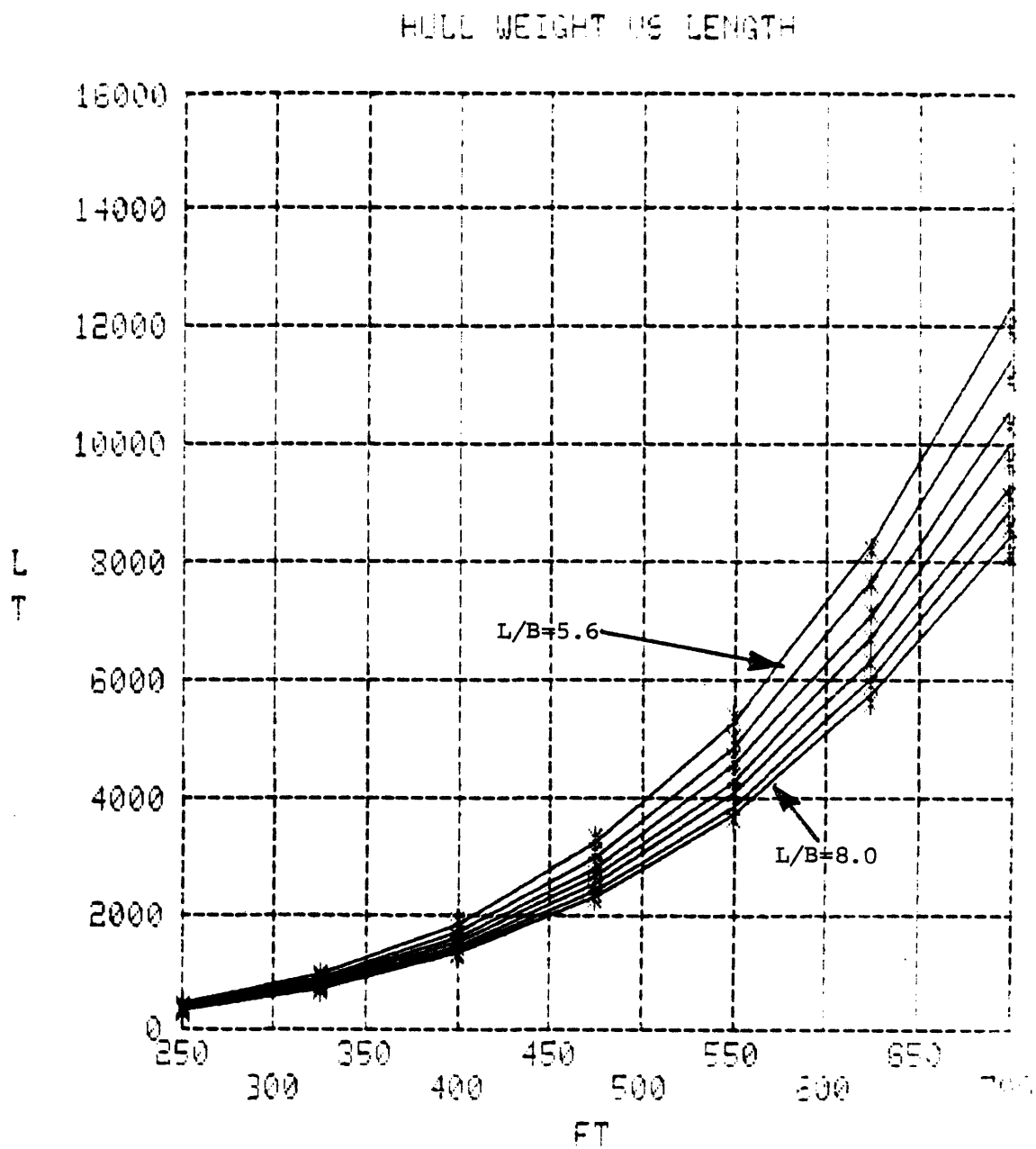


FIGURE 3.11
 PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH
 FOR $C_B=0.80$ AND $B/T=2.8$

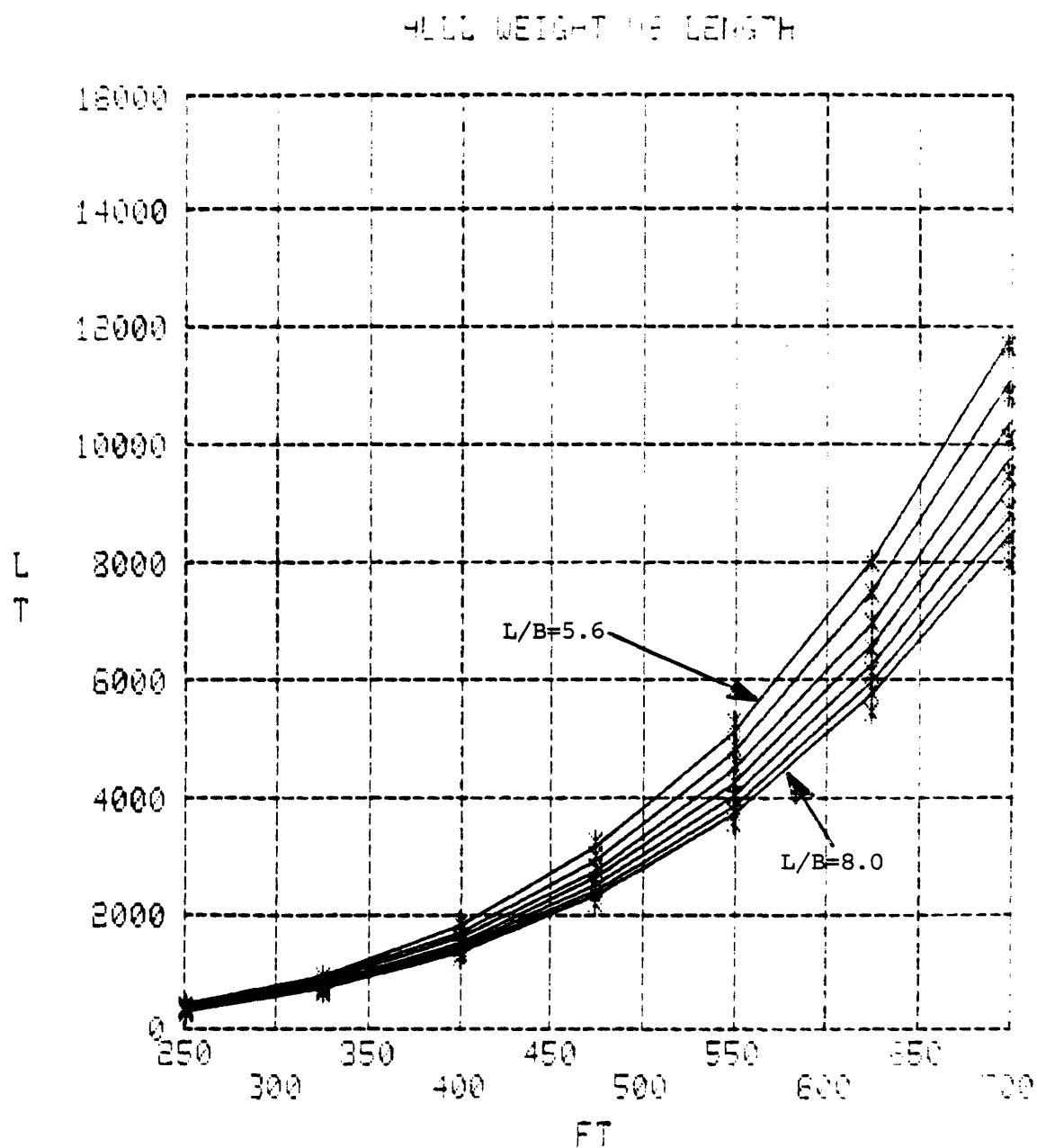


FIGURE 3.11
 PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH
 FOR $C_B=0.80$ AND $B/T=3.0$

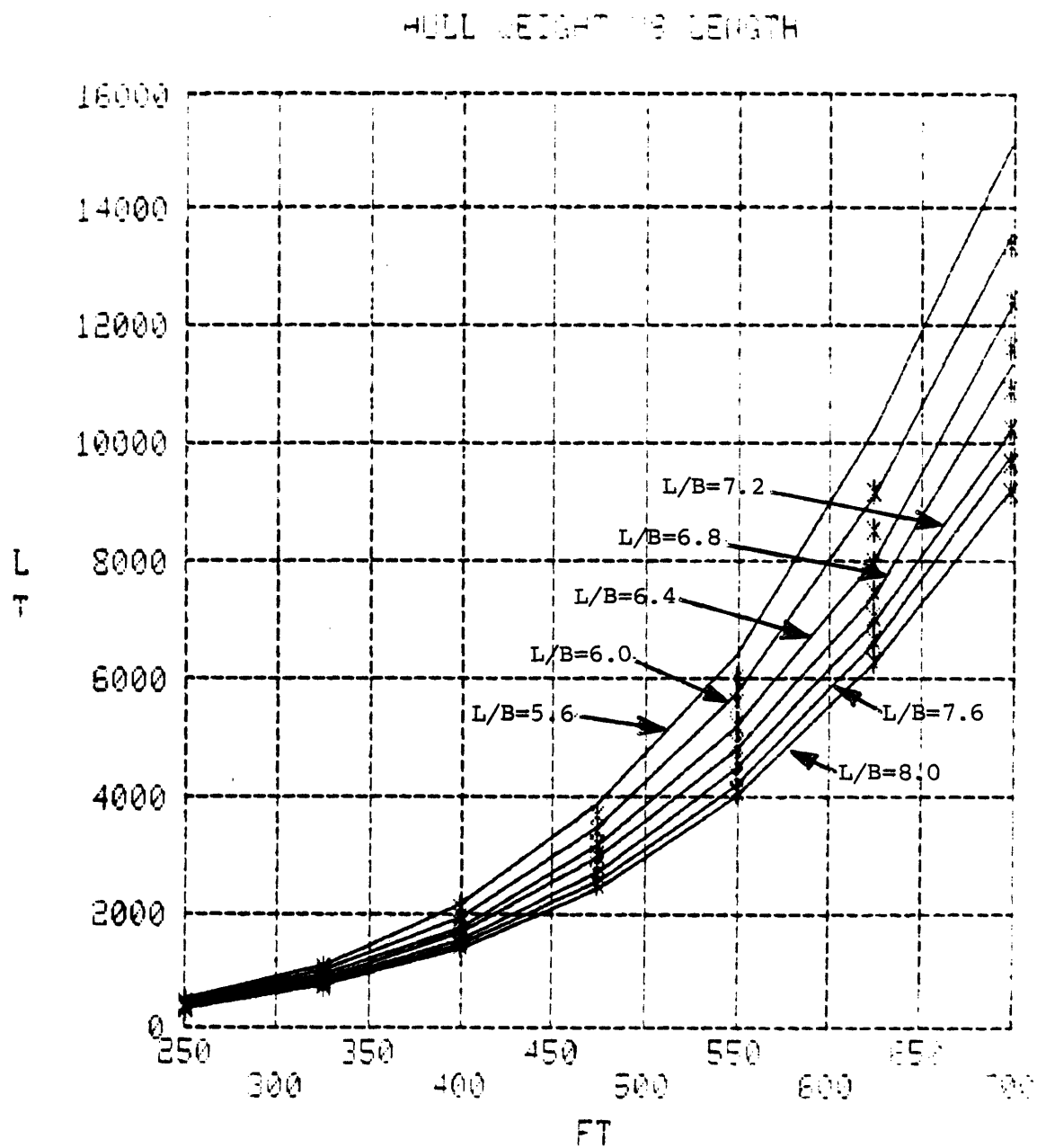


FIGURE 3.12

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH
FOR $C_B=0.85$ AND $B/T=2.2$

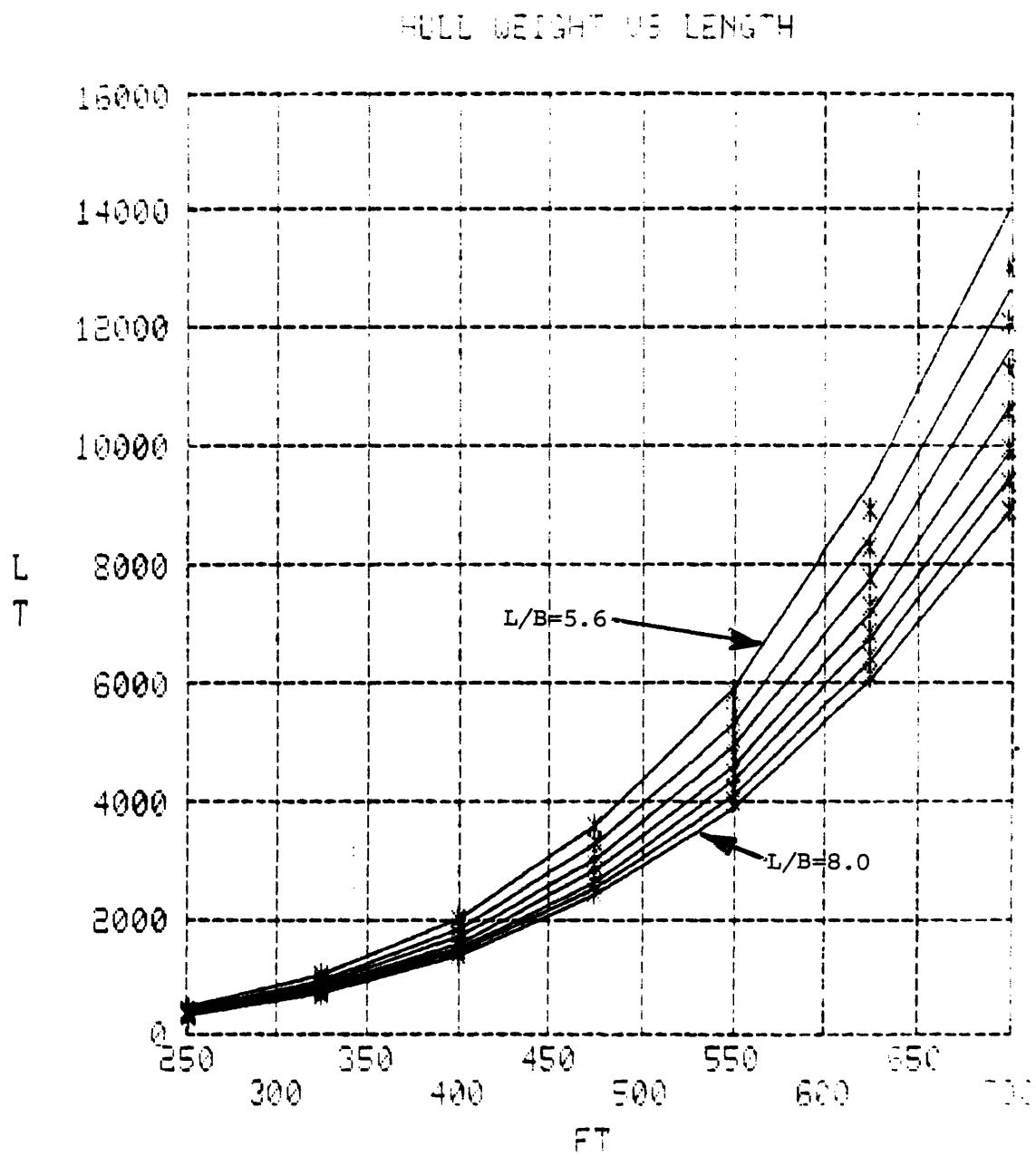


FIGURE 3.12
 PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH
 FOR $C_B=0.85$ AND $B/T=2.4$

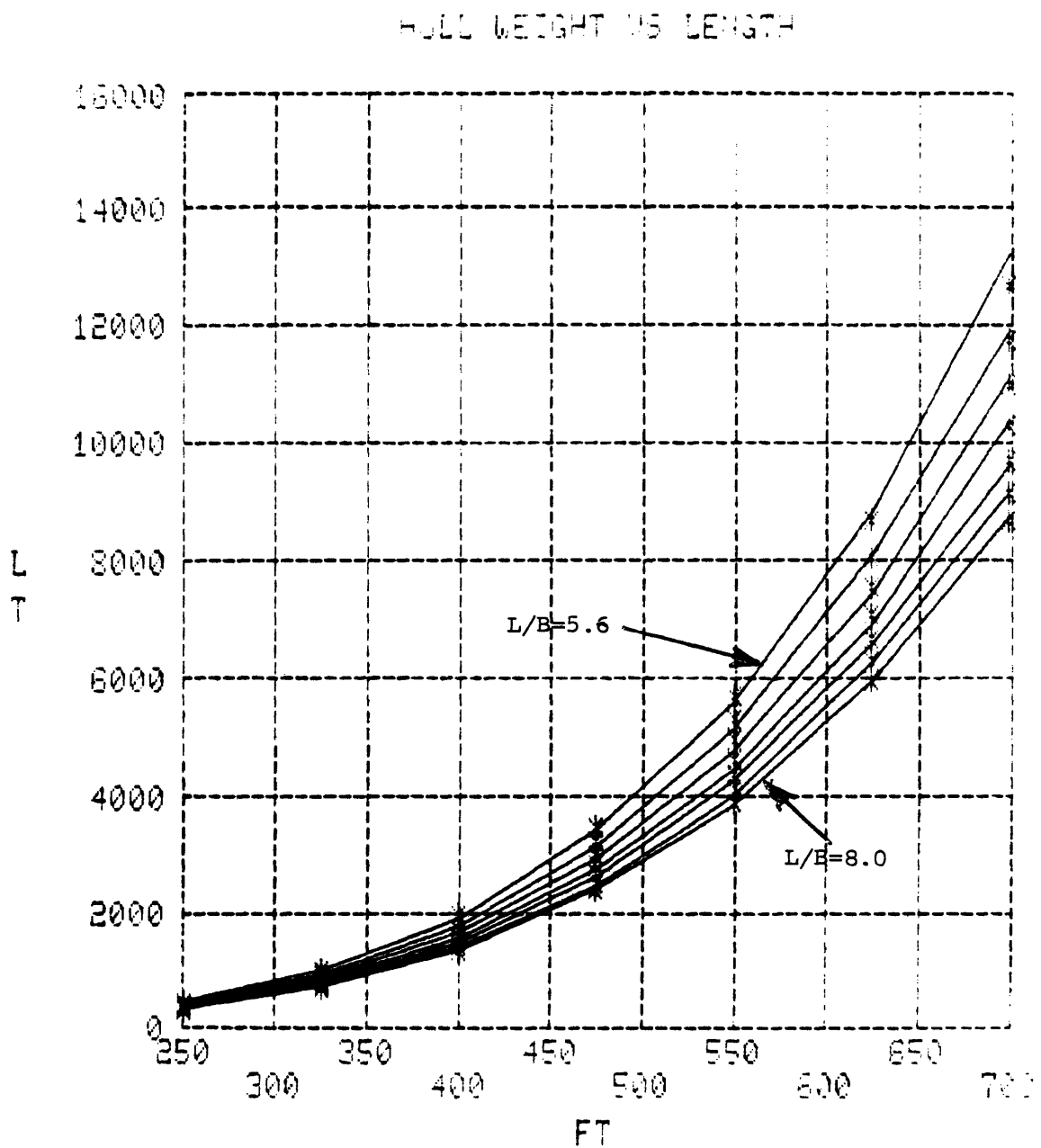


FIGURE 3.12
 PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH
 FOR $C_B=0.85$ AND $B/T=2.6$

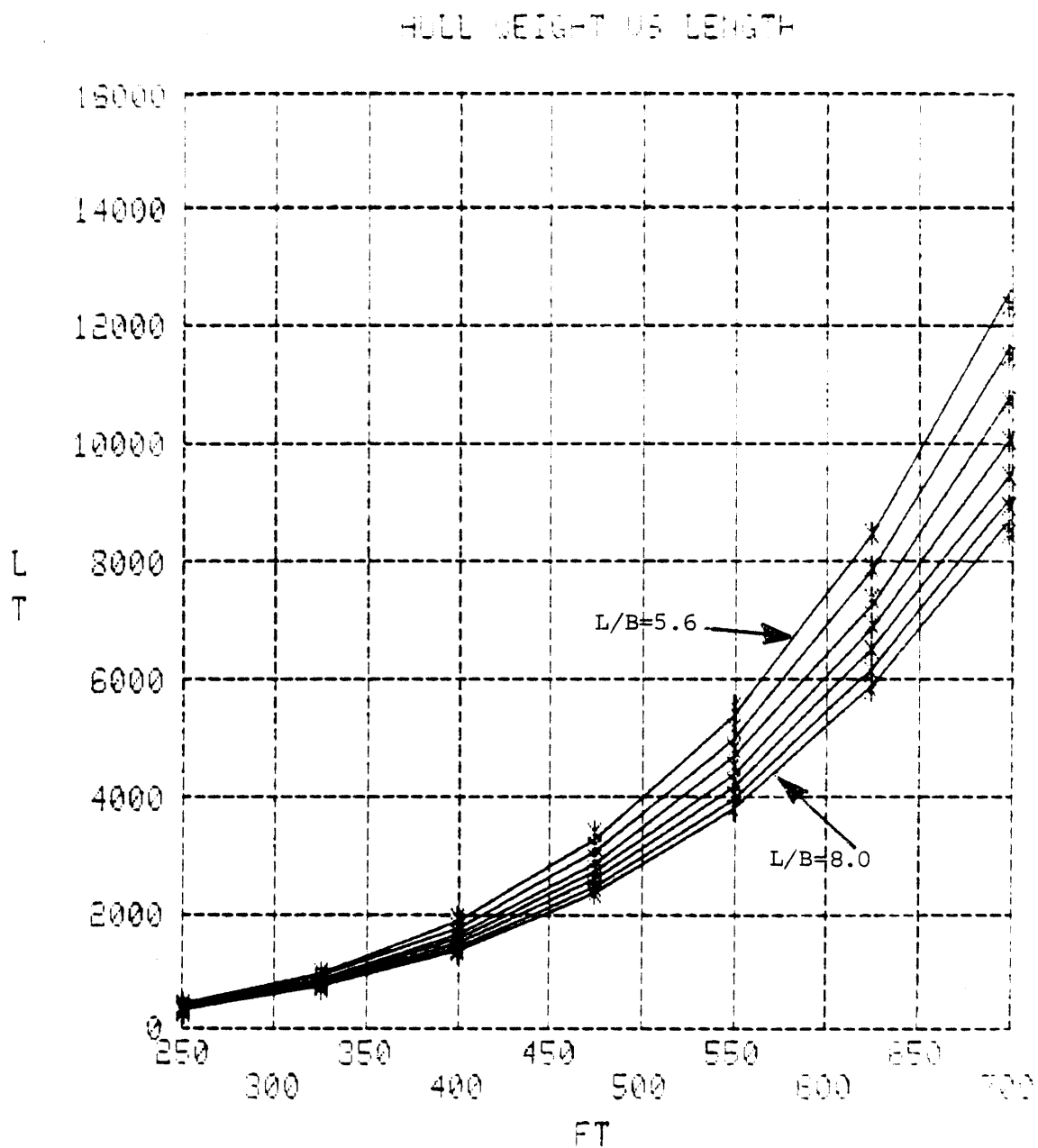


FIGURE 3.12
 PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH
 FOR $C_B=0.85$ AND $B/T=2.8$

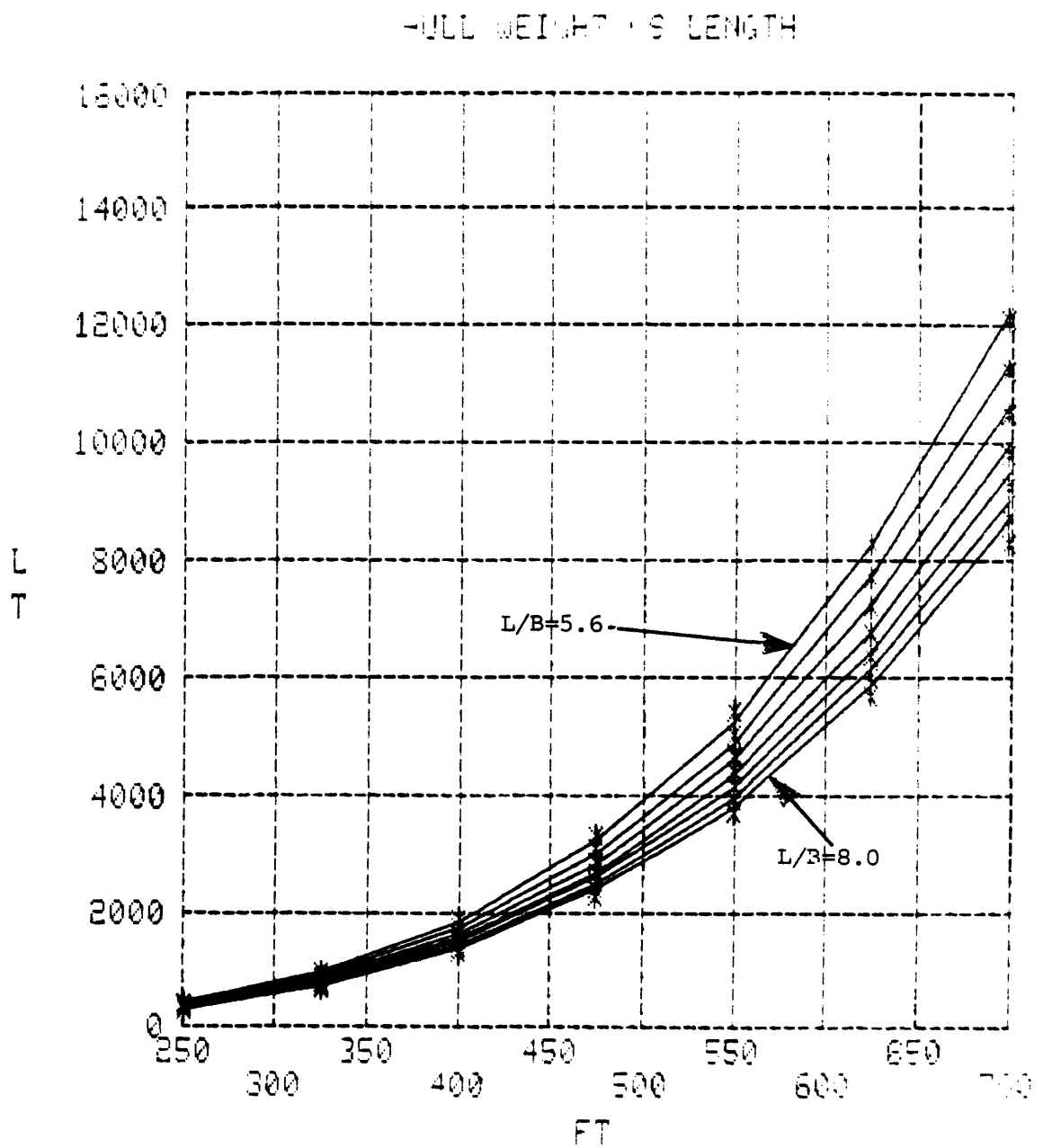


FIGURE 3.12
 PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH
 FOR $C_B=0.85$ AND $B/T=3.0$

TABLE 3.1
VALUES OF "FACT"

Barge Length (Meters)	"Fact"	Barge Length (Meters)	"Fact"
30	42.6	130	1661.0
35	56.0	135	1808.8
40	73.8	140	1961.6
45	95.4	145	2123.6
50	120.5	150	2292.9
55	151.9	155	2467.7
60	185.1	160	2650.8
65	228.4	165	2848.6
70	278.3	170	3049.7
75	338.0	175	3269.1
80	419.1	180	3504.6
85	540.4	185	3741.9
90	666.6	190	3993.9
95	787.2	195	4241.6
100	914.2	200	4499.4
105	1029.0	205	4767.3
110	1141.2	210	5058.9
115	1261.2	215	5348.8
120	1385.9	220	5640.9
125	1523.5	225	5953.2
		230	6269.5

TABLE 3.1...(Continued)

VALUES OF "FACT"

Barge Length (Feet)	"Fact"	Barge Length (Feet)	"Fact"	Barge Length (Feet)	"Fact"
100	6.8	320	132.0	540	439.0
110	8.1	330	144.0	550	458.0
120	9.3	340	155.0	560	477.0
130	11.2	350	165.0	570	498.0
140	13.1	360	176.0	580	519.0
150	15.3	370	186.0	590	542.0
160	17.5	380	199.0	600	564.0
170	20.4	390	210.0	610	587.0
180	23.4	400	223.0	620	611.0
190	26.5	410	236.0	630	635.0
200	29.7	420	249.0	640	658.0
210	34.0	430	262.0	650	682.0
220	38.3	440	276.0	660	707.0
230	43.3	450	291.0	670	732.0
240	48.2	460	305.0	680	759.0
250	55.1	470	320.0	690	787.0
260	62.0	480	336.0	700	814.0
270	74.0	490	352.0	710	842.0
280	85.0	500	368.0	720	869.0
290	97.0	510	385.0	730	899.0
300	109.0	520	402.0	740	928.0
310	120.0	530	420.0	750	958.0

Source: American Bureau of Shipping, Rule for Building and Classing Steel Barges for Offshore Service, 1973, Table 3.2

TABLE 3.2

DEPTH OF LONGITUDINAL BULKHEAD AND SIDE GIRDERS
AND TOTAL NUMBER OF STANCHIONS INSERTED BETWEEN SIDES

"b" Barge Molded Breadth	Depth of Longitudinal Bulkhead and Side Girders	Total No. of Stanchions Inserted Between Sides
< 35'	5'	0
35' - 67'	7'	2
67' - 99'	9'	4
> 99'	11'	6

TABLE 3.3
MINIMUM PLATE THICKNESS

Plate Type	Applicable Rule	Minimum Thickness Calculation (mm and inches)	Valid Range for Equation
Bottom	ABS (1973) para. 4.3.3	$t_{bot} = 0.0463s + 1.7 \text{ mm}$ $= 0.0833s + 5.1 \text{ mm}$ $t_{bot} = 0.000555s + 0.067 \text{ in}$ $= 0.001s + 0.01 \text{ in}$	$s \leq 122 \text{ m}$ $122 < s \leq 228.5 \text{ m}$ $s \leq 400 \text{ ft}$ $400 < s \leq 750 \text{ ft}$
Side	ABS (1973) para 4.3.1	$t_{sid} = 0.06347s + 0.0075s \text{ mm}$ $= 0.02706s + 3.86 \text{ mm}$ $t_{sid} = 0.00082s + 0.0075s \text{ in}$ $= 0.00035s + 0.152 \text{ in}$	$s \leq 152.5 \text{ m}$ $152.5 < s \leq 228.5 \text{ m}$ $s \leq 500 \text{ ft}$ $500 < s \leq 750 \text{ ft}$
Deck	ABS (1973) para 5.1.1	$t_{dk} = 0.01s + 1.8 \text{ mm}$ $= 0.00643s + 4.3 \text{ mm}$ $t_{dk} = 0.01s + 0.07 \text{ in}$ $= 0.00643s + 0.17 \text{ in}$	$s \leq 711 \text{ mm}$ $711 < s < 1016 \text{ mm}$ $s \leq 28 \text{ in}$ $28 < s < 40 \text{ in}$
Longitudinal Bulkhead	ABS (1973) para 9.9.1	$t_{bhd} = \frac{s\sqrt{h}}{254} + 2.5 \text{ mm}$ $t_{bhd} = \frac{s\sqrt{h}}{460} + 0.10 \text{ in}$	$h = "d" + 1.22 - "dbdep"$ $h = "d" + 4 - "dbdep"$

Notes: s: longitudinal frame spacing (12x"lonspac")

h: bulkhead plate head; the distance from the lower edge of longitudinal bulkhead plating to 4' (1.22m) above the deck

TABLE 3.4
MINIMUM REQUIRED SECTION MODULI
FOR LONGITUDINAL FRAMES

Longitudinal frames must have section moduli "secmod_x" greater than the value obtained from the formulae:

$$\text{"secmod}_x = 7.9 \times c \times h \times s \times l^2 \text{ cm}^3$$

or

$$\text{"secmod}_x = 0.0041 \times c \times h \times s \times l^2 \text{ in}^3$$

where the appropriate values of x, c, h, s and l are found in the table below:

Frame Type	Applicable Rule	x	c	h*	s	l
Bottom	ABS (1973) para 6.15.1	1	1.34 1.30***	"h1"	"lonspac"	"xvrspace"
Side	ABS (1973) para 6.15.1	3	1.25	"h2"***	"	"
Deck	ABS (1973) para 6.15.1	5	1.25	"h3"	"	"
Longitudinal Bulkhead	ABS (1973) para 9.9.2	7	1.00	"h2"***	"	"
Double-Bottom	ABS (1973) para 6.19.2	9	1.00	"h5"	"	"

*h: head of the associated member as shown in Figures 3.1 and 3.2

***"h2": Average head experienced by side/bulkhead members. It was used since the required and calculated section moduli are both linear with respect to head

***: C-value for bottom longitudinals when double bottom is installed

TABLE 3.5

MINIMUM REQUIRED SECTION MODULI
FOR TRANSVERSE GIRDERS

Transverse girders must have section moduli "secmod_x"
greater than the value obtained from the formulae:

$$\begin{aligned} \text{or } \text{"secmod}_{\underline{x}} &= 4.74 \times c \times h \times s \times l^2 \text{ cm}^3 \\ \text{"secmod}_{\underline{x}} &= 0.0025 \times c \times h \times s \times l^2 \text{ in}^3 \end{aligned}$$

where the appropriate values of x, c, h, s and l are found in the
table below:

Girder Type	Applicable Rule	x	c	h*	s	l**
Bottom	ABS(1973) para 6.15.2	2	1.75	"h1"	"xvrspac"	"12"
Side	ABS(1973) para 6.15.2	4	1.75	"h2"	"	"14"
Deck	ABS(1973) para 6.15.2	6	2.00	"h3"	"	"12"
Longitudinal Bulkhead	ABS(1973) para 9.9.3	8	1.50	"h2"	"	"14"

*h: head of the associated member as shown
in Figures 3.1 and 3.2

**"l": the unsupported span of the girder as
previously calculated in Section 3.2.3
and shown in Figures 3.1 and 3.2

TABLE 3.6
TRANSVERSE MEMBER WEIGHT PER LONGITUDINAL FOOT CALCULATIONS

Member	Weight Per Longitudinal Foot = [Steel Volume] x $\left[\frac{\text{Longitudinal Weight}}{\text{Per Foot Factor} - \text{lb/in}^2\text{-ft}} \right]$ Program Variable
Bottom Transverse Girder	"xvrs1" = $\left[\left(\frac{\text{cross-sectional area}^*}{\text{of member-"area "}} \right) \times "b" \right] \times \left(\frac{3.4}{\text{"xvrspac"}} \right)$
Side Transverse Girders(2)	"xvrs2" = $\left[2 \times \left(\frac{\text{cross sectional area}^*}{\text{of member-"area4"}} \right) \times \left("d" - \frac{\text{"dbdep"}}{12} \right) \right] \times \left(\frac{3.4}{\text{"xvrspac"}} \right)$
Deck Transverse Girder	"xvrs3" = $\left[\left(\frac{\text{cross sectional area}^*}{\text{of member-"area6"}} \right) \times "b" \right] \times \left(\frac{3.4}{\text{"xvrspac"}} \right)$
Longitudinal Bulkhead Transverse Girder	"xvrs4" = $\left[\left(\frac{\text{cross sectional area}^*}{\text{of member-"area6"}} \right) \times \left("d" - \frac{\text{"dbdep"}}{12} \right) \right] \times \left(\frac{3.4}{\text{"xvrspac"}} \right)$
Transverse Bulkhead Plating	"xvrs5" = $\left[\frac{1}{\text{"ntrbkd"}} \times \left("b" \times \left("d" - \frac{\text{"dbdep"}}{12} \right) \times \text{"bhdplt"} \right) \right] \times \left(\frac{3.4 \times 12}{\text{"xvrspac"}} \right)$
Transverse Bulkhead Longitudinal Stiffeners	"xvrs6" = $\left[\frac{1}{\text{"ntrbkd"}^{**}} \times \left("b" \times \left(\frac{d - \text{"dbdep"}}{12} \right) \times \left(\frac{\text{cross sectional area}}{\text{of member-"area7"}^{**}} \right) \right) \right] \times \left(\frac{3.4}{\text{"xvrspac"}} \right)$
Stanchions	"xvrs7" = $\left[\left(\frac{\text{number of stanchions}}{\text{per transverse girder "stans"}} \right) \times \left(\frac{\text{cross sectional area}}{\text{of stanchion"astans"}} \right) \times \left(\frac{\text{length of}}{\text{stanchion}} \right) \right] \times \left(\frac{3.4}{\text{"xvrspac"}} \right)$ where "astan" = $\pi \times \left(\frac{\text{radius of the cylindrical shaped stanchion}^{****}}{2} \right)^2$

* Cross sectional areas of members are obtained from scantlings calculated in "smact2".

** "ntrbkd" is the number of transverse girders between each transverse bulkhead.

*** $\left(\frac{d - \text{"dbdep"}}{12} \right) / \text{"lonspac"}$ is the number of stiffeners attached to the transverse bulkhead plate.

**** The radius of the stanchion was determined by equating the minimum stanchion load equations found in ABS (1973, par. 6.7.2 and Figure (6)). Specifically,

$$\begin{Bmatrix} 1.07 \\ 0.03 \end{Bmatrix} \times "12" \times "h" \times \text{"xvrspac"} = w = \begin{Bmatrix} [1.232 - 0.00452 ("14"/r)] \pi r^2 \\ [7.83 - 0.345 ("14"/r)] \pi r^2 \end{Bmatrix} \begin{Bmatrix} \text{(MT)} \\ \text{(LT)} \end{Bmatrix}$$

where r is radius of stanchion and "12", "14", "h", "xvrspac" are as shown in Figures 3.1 and 3.2.

TABLE 3.7
INITIAL SCANTLING VALUES USED IN SUBROUTINE "smact2"

Dimension	Reference	Initial Value
Plate thickness		as specified in input "tp"
Plate depth		as specified in input "s"
Web depth	ABS (1973) para 9.9.3c ABS (1978) para 22.27.5	the larger of (1) 2.5 x long. frame and: (2) { 1.5 x "14", for side and bulkhead girders 2.4 x "12", for bottom girders 1.5 x "12", for deck girders and then rounded up to nearest 1/4"
Web thickness	ABS (1978) Table 22.3 ABS (1973) para 9.9.3c reasonable assumption	the larger of: (1) { 0.34" for "1" < 200' 0.34" + ("1" - 200)/3500 for 200' < "1" < 270' 0.36" + ("1" - 270)/3000 for 270' < "1" < 700 0.50" for 700' < "1" for deck, bottom, and side girders or the smaller of: (a) 0.44" and (b) 0.12" + $\frac{\text{"webdepth"}}{100}$ for longitudinal bulkhead girder and: (2) { plate thickness for plate thicker than 5/8" or plate thickness less 1/4" for plate thinner than 5/8" and then rounded up to nearest 1/16"
Flange width	reasonable minimum Horne(1958)	the smaller of: (1) web depth/5 and: (2) 20 x web thickness rounded up to the nearest 1/4"
Flange thickness	reasonable assumption	plate thickness

TABLE 3.8

DOUBLE BOTTOM PARTICULARS

Member	Applicable Rule	Minimum Dimensions (Metric and English units)
Double-Bottom Depth	ABS(1978) para 7.3.2	$\text{"dbdep"} = 0.384 \times \text{"b"} + 4.13\sqrt{\text{"t"}} \text{ m}$ $= 32 \times \text{"b"} + 190 \sqrt{\text{"t"}} \text{ ft}$
Double-Bottom Plate Thickness	ABS(1978) para 7.5	$\text{"dbplt"} = 0.037 \times \text{"l"} + 0.009 \times s + 0.5 \text{ mm}$ $= 0.000445 \times \text{"l"} + 0.009 \times s + 0.02 \text{ in}$ where $s = 12 \times \text{"lonspac"}$
Double-Bottom Center Girder Plate Thickness	ABS(1978) para 7.3.2	$\text{"cendbgirdth"} = 0.056 \times \text{"l"} + 5.5 \text{ mm}$ $= 0.00067 \times \text{"l"} + 0.22 \text{ in}$
Double-Bottom Side Girder and Floor Plate Thickness	ABS(1978) para 7.3.4.	$\text{"sidbgirdth"} = 0.036 \times \text{"l"} + 4.7 \text{ mm}$ $= 0.00043 \times \text{"l"} + 0.24 \text{ in}$
Double-Bottom Head	ABS(1973) para 6.19.2	$\text{"h5"} = \text{"d"} - \text{"dbdep"} + \begin{cases} 1.2192 & \text{"l"} < 60.96\text{m} \\ \text{"l"}/50 & 60.96 \leq \text{"l"} \leq 121.92 \\ 2.4384 & 121.92 < \text{"l"} \end{cases}$ (meters) $\text{"h5"} = \text{"d"} - \text{"dbdep"} + \begin{cases} 4 & \text{"l"} < 200 \\ \text{"l"}/50 & 200 \leq \text{"l"} \leq 400 \\ 8 & 400 < \text{"l"} \end{cases}$ (feet)
Double-Bottom Longitudinal Frame Section Modulus	ABS(1973) para 6.19.2	$\max (\text{"secmod9"}^*, 0.85 \times \text{"secmod1"}^*)$

*"secmod9" and "secmod1" values determined
from expressions found in Table 3.4

TABLE 3.9

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.75

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
2.2	5.6	472	1061	2107	3763	6156	9946	14723
	5.8	452	992	1987	3542	5768	9317	13771
	6.0	431	942	1873	3357	5524	8785	12994
	6.2	412	907	1785	3193	5214	8502	12368
	6.4	396	868	1735	3057	4956	7736	11811
	6.6	380	829	1651	2957	4787	7458	11228
	6.8	364	802	1584	2848	4600	7194	10883
	7.0	351	771	1519	2730	4428	6903	10353
	7.2	360	754	1478	2592	4289	6691	9796
	7.4	346	728	1446	2521	4178	6457	9492
	7.6	335	704	1395	2456	4007	6302	9188
	7.8	321	689	1377	2364	3914	6106	8954
	8.0	307	672	1337	2298	3825	5906	8705
2.4	5.6	447	983	1960	3443	5660	9056	13358
	5.8	427	936	1861	3295	5345	8542	12774
	6.0	406	891	1774	3164	5188	8144	12021
	6.2	389	858	1701	3030	4918	7762	11570
	6.4	374	818	1640	2907	4717	7337	11062
	6.6	360	800	1592	2810	4581	7113	10633
	6.8	347	769	1526	2701	4382	6808	10235
	7.0	335	748	1482	2628	4239	6678	9937
	7.2	344	727	1450	2513	4106	6429	9441
	7.4	329	704	1402	2439	4003	6226	9145
	7.6	316	686	1376	2349	3892	6056	8929
	7.8	302	676	1328	2318	3779	5888	8710
	8.0	293	660	1304	2275	3696	5762	8469

TABLE 3.9 --Continued

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.75

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
2.6	5.6	423	937	1853	3289	5345	8461	12591
	5.8	406	895	1792	3160	5090	8075	11968
	6.0	389	861	1714	3026	4860	7778	11422
	6.2	372	831	1649	2900	4703	7419	10952
	6.4	358	801	1595	2804	4521	7145	10584
	6.6	346	776	1551	2725	4404	6892	10224
	6.8	327	753	1494	2618	4260	6615	9890
	7.0	315	738	1460	2530	4121	6433	9607
	7.2	326	717	1417	2458	4008	6275	9230
	7.4	315	706	1378	2412	3900	6091	8993
	7.6	304	687	1345	2347	3800	5930	8728
	7.8	291	669	1305	2290	3735	5770	8519
	8.0	282	653	1289	2253	3631	5663	8323
2.8	5.6	407	908	1795	3158	5138	8129	11909
	5.8	390	872	1732	3050	4917	7852	11432
	6.0	374	835	1678	2919	4755	7512	11018
	6.2	358	813	1619	2809	4581	7257	10601
	6.4	338	796	1572	2705	4423	6933	10238
	6.6	325	768	1502	2647	4304	6678	9991
	6.8	313	759	1490	2572	4165	6518	9641
	7.0	302	733	1433	2515	4055	6337	9378
	7.2	316	712	1406	2443	3930	6128	9079
	7.4	305	693	1368	2408	3815	6001	8857
	7.6	295	699	1333	2343	3742	5847	8664
	7.8	282	686	1314	2287	3651	5696	8453
	8.0	273	666	1273	2258	3590	5573	8280

TABLE 3.9 --Continued

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.75

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
3.0	5.6	395	887	1761	3096	4981	7908	11541
	5.8	370	854	1699	2944	4822	7592	11100
	6.0	354	835	1634	2853	4657	7289	10794
	6.2	339	803	1594	2746	4515	7099	10429
	6.4	326	796	1545	2698	4366	6810	10097
	6.6	314	768	1501	2607	4230	6604	9799
	6.8	302	745	1472	2551	4112	6460	9503
	7.0	289	750	1430	2487	4021	6300	9298
	7.2	308	736	1393	2433	3897	6086	9045
	7.4	298	711	1355	2405	3828	5944	8760
	7.6	291	694	1345	2347	3752	5796	8611
	7.8	281	684	1316	2314	3688	5667	8424
	8.0	275	669	1299	2258	3617	5556	8253

TABLE 3.10

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.80

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
2.2	5.6	473	1064	2122	3793	6223	10076	14948
	5.8	452	996	2012	3630	6016	9666	14434
	6.0	432	961	1894	3384	5547	8976	13262
	6.2	413	909	1820	3250	5317	8530	12652
	6.4	397	870	1737	3138	5115	7917	12002
	6.6	382	835	1669	3003	4875	7558	11620
	6.8	365	809	1624	2893	4702	7307	11039
	7.0	352	786	1549	2789	4512	7039	10701
	7.2	361	757	1505	2645	4404	6883	10120
	7.4	347	738	1471	2586	4243	6651	9771
	7.6	335	714	1436	2499	4103	6408	9453
	7.8	321	696	1397	2404	3992	6243	9177
	8.0	308	679	1362	2349	3891	6056	8917
2.4	5.6	448	996	1989	3504	5748	9254	13653
	5.8	428	943	1885	3365	5435	8724	12952
	6.0	407	903	1792	3206	5201	8259	12328
	6.2	389	873	1737	3087	5003	8036	11853
	6.4	374	842	1673	2972	4812	7491	11332
	6.6	360	805	1623	2890	4661	7298	10866
	6.8	348	779	1559	2755	4461	6962	10462
	7.0	336	756	1519	2679	4320	6772	10254
	7.2	344	734	1468	2551	4198	6618	9800
	7.4	329	725	1431	2488	4089	6359	9427
	7.6	317	705	1389	2417	3989	6203	9185
	7.8	303	695	1363	2367	3871	6072	8895
	8.0	294	680	1329	2330	3791	5871	8673

TABLE 3.10 --Continued

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.80

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
2.6	5.6	427	952	1898	3343	5457	8660	12770
	5.8	406	916	1832	3208	5201	8250	12260
	6.0	389	871	1744	3070	4966	7897	11704
	6.2	373	842	1684	2951	4785	7638	11262
	6.4	359	809	1629	2863	4648	7250	10936
	6.6	346	784	1575	2770	4477	7096	10612
	6.8	328	771	1516	2667	4384	6809	10149
	7.0	316	745	1479	2596	4259	6597	9861
	7.2	326	745	1450	2500	4100	6405	9485
	7.4	315	720	1401	2466	3991	6282	9221
	7.6	305	700	1376	2415	3902	6088	9002
	7.8	292	691	1350	2367	3835	5915	8735
	8.0	283	680	1305	2303	3738	5780	8523
2.8	5.6	408	920	1829	3224	5263	8345	12337
	5.8	391	880	1776	3082	5069	7941	11712
	6.0	375	849	1697	2964	4843	7687	11444
	6.2	359	844	1644	2879	4678	7415	10952
	6.4	340	803	1588	2792	4541	7089	10610
	6.6	326	796	1555	2682	4398	6863	10198
	6.8	314	773	1506	2643	4266	6731	10035
	7.0	303	749	1472	2562	4154	6514	9689
	7.2	316	735	1438	2529	4077	6321	9296
	7.4	306	720	1404	2461	3957	6166	9078
	7.6	296	713	1365	2382	3827	6003	8900
	7.8	284	698	1341	2339	3766	5867	8660
	8.0	280	678	1321	2294	3706	5758	8454

TABLE 3.10 --Continued

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.80

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
3.0	5.6	395	896	1785	3163	5088	8070	11822
	5.8	378	902	1711	2998	4957	7803	11497
	6.0	355	867	1670	2917	4760	7520	11096
	6.2	340	837	1631	2829	4616	7240	10745
	6.4	327	807	1582	2757	4475	6974	10370
	6.6	315	776	1547	2664	4339	6795	10080
	6.8	302	774	1494	2626	4226	6591	9767
	7.0	293	765	1467	2555	4116	6434	9503
	7.2	311	747	1433	2501	4035	6275	9281
	7.4	304	724	1399	2436	3953	6142	9067
	7.6	299	714	1376	2396	3840	5979	8796
	7.8	285	698	1342	2358	3793	5895	8611
	8.0	276	689	1331	2313	3700	5740	8468

TABLE 3.11

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.85

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
2.2	5.6	475	1073	2141	3874	6382	10275	15128
	5.8	453	1015	2051	3610	5964	9682	14433
	6.0	434	964	1930	3472	5730	9124	13569
	6.2	414	926	1832	3325	5398	8716	12924
	6.4	398	881	1754	3164	5165	8030	12343
	6.6	382	859	1702	3054	5003	7821	11858
	6.8	366	826	1656	2958	4781	7411	11355
	7.0	353	797	1585	2847	4608	7168	10895
	7.2	361	769	1532	2692	4463	6973	10232
	7.4	348	748	1506	2629	4329	6777	10013
	7.6	336	726	1450	2535	4199	6583	9810
	7.8	323	715	1419	2457	4087	6373	9370
	8.0	308	704	1374	2403	3980	6259	9255
2.4	5.6	449	1001	2001	3584	5880	9382	14033
	5.8	429	950	1914	3389	5578	8946	13266
	6.0	408	917	1826	3262	5294	8495	12620
	6.2	390	881	1765	3144	5132	8052	12059
	6.4	376	849	1692	3017	4930	7749	11636
	6.6	361	822	1641	2905	4766	7412	11037
	6.8	348	793	1579	2817	4568	7171	10695
	7.0	336	783	1542	2712	4445	6941	10416
	7.2	345	761	1494	2613	4321	6740	9887
	7.4	330	744	1482	2548	4198	6501	9644
	7.6	317	729	1421	2501	4092	6364	9477
	7.8	304	711	1388	2452	3978	6204	9144
	8.0	294	692	1357	2388	3878	6046	8917

TABLE 3.11 --Continued

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.85

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
2.6	5.6	427	962	1915	3402	5576	8843	13238
	5.8	407	918	1846	3259	5338	8509	12634
	6.0	390	889	1760	3128	5135	8114	11937
	6.2	374	855	1700	3018	4890	7801	11473
	6.4	360	841	1651	2930	4758	7425	11126
	6.6	347	812	1595	2829	4575	7234	10824
	6.8	329	795	1562	2723	4448	6909	10401
	7.0	316	772	1515	2660	4338	6752	10121
	7.2	327	753	1472	2607	4242	6565	9647
	7.4	316	730	1437	2530	4095	6410	9429
	7.6	306	730	1401	2458	4002	6243	9190
	7.8	293	707	1370	2398	3898	6080	9002
	8.0	283	695	1342	2382	3825	5938	8764
2.8	5.6	409	931	1870	3266	5365	8478	12607
	5.8	391	888	1808	3152	5147	8228	11991
	6.0	376	880	1730	3059	4975	7869	11670
	6.2	360	860	1671	2936	4818	7588	11189
	6.4	341	833	1627	2854	4662	7259	10835
	6.6	327	808	1582	2766	4512	7050	10436
	6.8	315	782	1540	2709	4372	6843	10081
	7.0	304	782	1500	2629	4276	6637	9903
	7.2	317	751	1466	2565	4157	6523	9496
	7.4	310	740	1429	2549	4079	6329	9296
	7.6	302	723	1396	2450	3956	6190	9088
	7.8	291	707	1377	2426	3868	6030	8875
	8.0	287	705	1356	2346	3780	5889	8734

TABLE 3.11 --Continued

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.85

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
3.0	5.6	396	927	1826	3222	5235	8282	12258
	5.8	378	909	1761	3067	5055	7961	11698
	6.0	357	874	1706	2993	4871	7740	11378
	6.2	341	846	1652	2886	4752	7423	10923
	6.4	328	825	1619	2832	4590	7245	10622
	6.6	316	809	1571	2757	4457	6962	10326
	6.8	304	797	1539	2673	4323	6779	9971
	7.0	304	774	1498	2616	4223	6627	9759
	7.2	314	756	1462	2600	4132	6431	9506
	7.4	309	751	1442	2517	4043	6284	9266
	7.6	300	732	1416	2461	3956	6148	9038
	7.8	286	713	1381	2408	3876	6003	8872
	8.0	277	698	1353	2399	3792	5870	8701

TABLE 3.12
DOUBLE BOTTOM TANK BARGE HULL WEIGHTS FOR CB=0.80

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
2.2	5.6	552	1189	2312	4039	6539	10430	15184
	5.8	526	1135	2198	3867	6184	9863	14382
	6.0	505	1091	2052	3678	5877	9295	13580
	6.2	483	1049	1980	3508	5555	8896	12839
	6.4	464	1009	1889	3362	5317	8260	12324
	6.6	447	970	1817	3211	5087	8001	11765
	6.8	432	932	1718	3104	4908	7678	11267
	7.0	414	903	1684	2977	4737	7291	10789
	7.2	401	890	1620	2819	4565	6998	10213
	7.4	390	858	1573	2729	4392	6779	9839
	7.6	378	828	1528	2641	4267	6576	9505
	7.8	368	804	1477	2568	4145	6339	9142
	8.0	357	786	1422	2477	4019	6149	8884
2.4	5.6	524	1138	2189	3781	6064	9577	14111
	5.8	501	1089	2084	3607	5779	9062	13303
	6.0	481	1045	1951	3437	5493	8721	12601
	6.2	462	1004	1864	3309	5237	8326	11964
	6.4	442	983	1823	3173	5051	7804	11487
	6.6	426	947	1745	3054	4833	7530	11037
	6.8	412	912	1664	2930	4690	7234	10583
	7.0	399	881	1615	2835	4512	6926	10197
	7.2	389	855	1547	2726	4369	6695	9696
	7.4	374	843	1499	2633	4211	6491	9413
	7.6	363	815	1462	2563	4102	6252	9077
	7.8	353	790	1427	2488	3963	6084	8800
	8.0	342	772	1390	2391	3858	5901	8547

TABLE 3.12 --Continued
DOUBLE BOTTOM TANK BARGE HULL WEIGHTS FOR CB=0.80

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
2.6	5.6	504	1100	2058	3595	5744	9061	13143
	5.8	481	1066	1989	3433	5523	8614	12473
	6.0	460	1024	1893	3289	5275	8218	11820
	6.2	443	982	1817	3171	5032	7888	11354
	6.4	428	946	1757	3041	4825	7484	10898
	6.6	413	915	1680	2937	4663	7224	10519
	6.8	397	903	1610	2854	4503	6980	10119
	7.0	385	870	1562	2764	4355	6683	9757
	7.2	375	840	1523	2650	4219	6438	9382
	7.4	363	816	1466	2561	4092	6284	9069
	7.6	351	796	1430	2499	3966	6108	8817
	7.8	340	786	1398	2447	3845	5936	8577
	8.0	331	762	1372	2350	3763	5808	8416
2.8	5.6	484	1076	2005	3457	5532	8602	12508
	5.8	465	1036	1922	3297	5314	8247	11966
	6.0	447	994	1836	3179	5095	7908	11345
	6.2	431	976	1761	3082	4845	7613	10917
	6.4	414	938	1707	2974	4687	7262	10502
	6.6	399	904	1647	2860	4533	6993	10153
	6.8	384	877	1577	2762	4380	6760	9771
	7.0	370	852	1530	2677	4236	6536	9477
	7.2	363	839	1488	2610	4091	6333	9230
	7.4	352	812	1460	2551	4030	6169	8990
	7.6	341	789	1432	2456	3920	6011	8755
	7.8	332	771	1390	2437	3817	5909	8567
	8.0	333	750	1375	2352	3753	5778	8347

TABLE 3.12 --Continued
DOUBLE BOTTOM TANK BARGE HULL WEIGHTS FOR CB=0.80

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
3.0	5.6	472	1066	1932	3340	5382	8325	12066
	5.8	454	1030	1864	3229	5161	7980	11548
	6.0	434	985	1797	3110	4962	7702	10970
	6.2	416	952	1736	3000	4723	7393	10587
	6.4	401	923	1676	2897	4590	7086	10186
	6.6	386	907	1628	2815	4406	6871	9963
	6.8	373	874	1570	2740	4308	6650	9619
	7.0	360	847	1531	2647	4182	6439	9413
	7.2	355	824	1486	2591	4090	6308	9192
	7.4	355	799	1467	2523	4000	6195	8960
	7.6	343	780	1427	2487	4930	6053	8794
	7.8	332	763	1420	2431	3817	5899	8594
	8.0	323	742	1380	2388	3761	5789	8462

TABLE 3.13

REGRESSION EQUATIONS FOR BARGE LENGTH, DEADWEIGHT,
AND HULL STEEL WEIGHT

Regression Type	Barge Type	C_B	REGRESSION EQUATION			
Barge Hull Steel Weight $f(L, L/B, B/T)$	Single-Skin	0.75	$e^{-9.921}$	$(L/B)^{-1.091}$	$(B/T)^{-0.372}$	$(L)^{3.290}$
	"	0.80	$e^{-10.063}$	$(L/B)^{-1.076}$	$(B/T)^{-0.350}$	$(L)^{3.308}$
	"	0.85	$e^{-10.188}$	$(L/B)^{-1.057}$	$(B/T)^{-0.331}$	$(L)^{3.323}$
	Double-Bottom	0.80	$e^{-8.784}$	$(L/B)^{-1.175}$	$(B/T)^{-0.389}$	$(L)^{3.147}$
Barge Hull Steel Weight $f(L/B, B/T, DWT)$	Single-Skin	0.75	$e^{-6.206}$	$(L/B)^{1.348}$	$(B/T)^{0.884}$	$(DWT)^{1.104}$
	"	0.80	$e^{-6.396}$	$(L/B)^{1.368}$	$(B/T)^{0.909}$	$(DWT)^{1.111}$
	"	0.85	$e^{-6.569}$	$(L/B)^{1.391}$	$(B/T)^{0.930}$	$(DWT)^{1.117}$
	Double-Bottom	0.80	$e^{-5.150}$	$(L/B)^{1.126}$	$(B/T)^{0.800}$	$(DWT)^{1.049}$
Barge Length $f(L/B, B/T, DWT)$	Single-Skin	0.75	$e^{1.128}$	$(L/B)^{0.742}$	$(B/T)^{0.382}$	$(DWT)^{0.336}$
	"	0.80	$e^{1.108}$	$(L/B)^{0.739}$	$(B/T)^{0.381}$	$(DWT)^{0.336}$
	"	0.85	$e^{1.088}$	$(L/B)^{0.737}$	$(B/T)^{0.379}$	$(DWT)^{0.336}$
	Double-Bottom	0.80	$e^{1.154}$	$(L/B)^{0.731}$	$(B/T)^{0.378}$	$(DWT)^{0.333}$
Barge Deadweight $f(L/B, B/T, L)$	Single-Skin	0.75	$e^{-3.359}$	$(L/B)^{-2.209}$	$(B/T)^{-1.137}$	$(L)^{2.978}$
	"	0.80	$e^{-3.295}$	$(L/B)^{-2.199}$	$(B/T)^{-1.133}$	$(L)^{2.976}$
	"	0.85	$e^{-3.237}$	$(L/B)^{-2.191}$	$(B/T)^{-1.129}$	$(L)^{2.975}$

NOTES: Barge depth is assumed to be the minimum allowed by the Freeboard rules presented in Table 2.5.
DWT refers to the barge cargo deadweight.

TABLE 3.14

COMPARISONS OF BARGE DESIGN MODEL'S HULL STEEL WEIGHT ESTIMATES
WITH ACTUAL BARGE HULL STEEL WEIGHTS

Barge Particulars L X B X D Draft DWT	Remarks	Assumed Scanting Length	Assumed Block Coefficient	Actual Hull Steel Weight	Calculated Hull Steel Weight	% Error
546'0"x85'0"x38'9" 32'0" 31,500	Single-Skin 2nd Generation Tank Barge	546'0"	0.86	4180 LT	4558 LT	+ 9%
		520'0"	0.90	"	4154	- 1%
581'0"x95'0"x46'0" 36'0" 40,500	Single-Skin Rigid Linked 3rd Generation Tank Barge	581'0"	0.83	5398 LT	6123 LT	+13%
		525'0"	0.90	"	5156	- 4%
450'0"x80'0"x32'6" 26'3" 20,000	Single-Skin 2nd Generation Tank Barge	450'0"	0.85	2181 LT	2668 LT	+22%
		420'0"	0.90	"	2330	+ 6%
269'3"x54'0"x17'6" 13'8" 4,200	Single-Skin 1st Generation Tank Barge	269'3"	0.85	588 LT	521 LT	-11%
		250'0"	0.90	"	462	-21%

CHAPTER 4

TUG-BARGE POWERING PROGRAM TO ESTIMATE DHP

4.1 Introduction

The tug-barge powering program "power" has been developed to provide an estimate of the horsepower required to be delivered to the propeller to drive an integrated tug-barge unit of given size and form at a specified speed. This delivered horsepower (DHP) estimate is used by the drop-and-swap program to estimate the horsepower of the machinery to be installed onboard the tug (IHP) and the horsepower required to be delivered to the shaft (SHP). The IHP estimate is used in a regression equation formulated by Sharp (1975) to estimate cost of propulsion machinery. The SHP estimate is used to determine fuel costs. This is done by simply multiplying SHP by cost per unit fuel, by voyage duration, and by specific fuel consumption. Thus, a reasonably accurate estimate of DHP is necessary to ensure that tug capital and operating costs used by the drop-and-swap program are reasonable.

The barge powering program is basically a table lookup and interpolation/extrapolation routine that utilizes the

tankship resistance and propulsion data of Tsuchida (1969) which is based on hull lines shown in Figure 4.1. From the resistance data the program determines the effective horsepower (EHP) needed to propel the tug-barge unit at the specified speed. Then the EHP, self-propulsion factors, and some assumed propeller particulars are fed into the Triantafyllou (1977) preliminary propeller design program to obtain estimates of the propeller's propulsive efficiency and consequently the horsepower required to be delivered to the propeller.

The tankship data of Tsuchida (1969) is used since, unfortunately, no series tests of large ocean going barge forms have been conducted. However, the tankship model tests use forms similar to the full-bodied integrated tug-barge hulls currently in operation today. Thus, no significant error should result from using this data for tug-barge data.

The propeller design program of Triantafyllou (1977) is used since this program uses a mathematical formulation of Wageningen B-series that is corrected for laminar flow effects. It provides very good agreement with model tests of commercial type hulls and propellers and thus is appropriate for use in the barge powering estimate. Since this program is documented in Triantafyllou (1977), the details of its logic will not be presented here.

In the text that follows, the logical structure of the powering program is presented. This is followed by the pre-

sentation and discussion of the table data and sample program outputs.

4.2 Logical Structure

The logical structure of the program is shown in flow chart form in Figure 4.2. In the text following, each block of the flow chart is explained (except for the propeller efficiency program explained in Triantafyllou (1977)) so that the tables and assumptions used are enumerated.

4.2.1 Input of Parameter Values

The barge powering program requires as input the tug-barge unit's principal dimensions, block coefficient, and speed. Specifically, the following values must be provided for the tug-barge unit design:

- "l": length at the waterline
- "b": molded breadth
- "dr": summer waterline draft
- "cb": barge block coefficient
- "v": speed

Note, except for "l" which corresponds to the length of the tug integrated with the barge, barge values are used above. This is because the tug is normally designed to integrate smoothly with the barge's lines so that the combined unit's dimensions and block coefficient (except for "l") are not significantly different from the barge alone.

These values are immediately converted to ratios--length breadth ("lb"="l"/"b"), breadth draft ("bt"="b"/"dr"), and Froude number ("fn"=0.2976x"v"/sqrt("l"))--which are used as inputs into the subroutines "resist" and "propfactors" described later. Limitations on the ranges of values that these ratios and the block coefficient may take are as follows:

$$0.75 \leq "cb" \leq 0.85$$

$$6.0 \leq "lb" \leq 8.0$$

$$"fn" \leq 0.22$$

$$2.0 \leq "bt" \leq 3.25$$

Either of the first two and both of the last two limitations must hold for the program to operate without significant error. The rationale of these limitations will be explained in the following text. Following the calculation of these ratios the program calls the subroutine "propfactors".

4.2.2 Calculation of Self-Propulsion Factors

The subroutine "propfactors" is a simple linear interpolation and quadratic extrapolation routine that provides values for wake fraction (W_T ="wa"), thrust deduction fraction (t ="th"), and relative rotative efficiency (η_R ="hr") as a function of "lb" and "cb". These self-propulsion factor values are interpolated or extrapolated from the tables of $(1-t)$, $(1-W_T)$, and (η_R) shown in Table 4.1 which were derived

from graphs presented in Tsuchida (1969, Figs. 60 and 63). These tables provide a range of "lb" values from 6.2 to 7.6 by 0.1 increments and a range of "cb" values from 0.770 to 0.840 by 0.005 increments for a Froude number of 0.16.

For "lb" and "cb" values that fall within the table ranges, the subroutine "propfactors" obtains the self-propulsion factors through a simple two dimensional linear interpolation of table values. If either "lb" or "cb" fall outside of the table ranges, then the subroutine utilizes quadratic extrapolation to extend the two nearest table entries for the variable that exceeds the table range and then uses linear interpolation between these two values for the other variable. For example, if "cb" is 0.75 and "lb" is 7.15 then the subroutine will use quadratic extrapolation to find the table extrapolation of "cb" to 0.75 for "lb" of 7.1 and 7.2. Then the subroutine will linearly interpolate between these two extrapolations to obtain the value for "lb" of 7.15.

To ensure that the extrapolation will not yield gross errors, table extensions have been limited to:

$$0.75 \leq \text{"cb"} \leq 0.85$$

or

$$6.0 \leq \text{"lb"} \leq 8.0$$

If either of these ranges are exceeded or if the values of both "lb" and "cb" exceed table ranges, the program indicates this error has occurred ("iprop"=1) to the calling program "power" and does not calculate the self-propulsion factors.

After obtaining the self-propulsion factors for the specified value of "lb" and "cb", the program corrects the wake fraction for Froude numbers greater than 0.16. The other factors do not need such correction according to Tsuchida (1969, pp. 7-8). Following this, the subprogram "power" calls the subroutine "resist" to obtain the residual resistance coefficient.

4.2.3 Calculation of Residual Resistance Coefficient

The subroutine "resist" is a four dimensional linear and quadratic extrapolation/interpolation routine that provides values for the residual resistance coefficient ("cr") as a function of "lb", "cb", "fn", and "bt". These residual resistance coefficients are interpolated and extrapolated from tables shown in Table 4.2 which were derived from the graphs presented in Tsuchida (1969, Figs. 7-32). These tables provide a range of "cb" values from 0.775 to 0.835 by 0.01 increments, a range of "lb" values from 6.2 to 7.6 by 0.2 increments, and a range of "fn" values from 0.14 to 0.22 by 0.01 increments (except between 0.14 and 0.16) for "bt" values of 2.46 and 2.76. These data are rerepresented in graphic form with "cr" as a function of "fn" for a range of "lb" values, with separate graphs for different "cb" and "bt" values in Figure 4.3.

After the subroutine reconstructs the residual resistance coefficient array by creating linearly interpolated

entries for a Froude number of 0.15, it proceeds with quadratic interpolation or extrapolation with respect to Froude number. That is, for the four "cb" and "lb" table values that bracket the inputted "cb" and "lb" value (for each "bt" value), the subroutine fits a parabola through the three closest Froude number table entries and then quadratically interpolates or extrapolates for the inputted Froude number. It then does a two dimensional linear interpolation among these four interpolated/extrapolated values with respect to "cb" and "lb". After this is done for both "bt" tables, the subroutine finishes by linearly interpolating or extrapolating between the two "bt" values for the inputted "bt" value.

To insure that the quadratic extrapolation routine does not yield gross errors, Froude number inputs are limited to a value of 0.22. No lower bound is specified since the values are almost constant for low Froude numbers so that extrapolation will cause little error.

After obtaining the residual resistance coefficient for the specified values of "lb", "bt", "cb", and "fn", the subprogram "power" proceeds to calculate the inputs for the subroutine "prop".

4.2.4 Calculation of Inputs for Subroutine "prop"

The propeller efficiency subroutine "prop" requires as input the self-propulsion factors previously calculated as well as some particulars concerning the propeller, i.e., its

diameter, number of blades, and expanded area ratio. It also requires an estimate of effective horsepower needed to drive the tug-barge hull at the specified speed (EHP) through still water.

The latter value is proportional to the unit's speed and hull resistance. The hull resistance is, in turn, proportional to the vessel's wetted surface area, square of its speed, and its total resistance coefficient. The wetted surface area is estimated by the formula given in Tsuchida (1969, p.11):

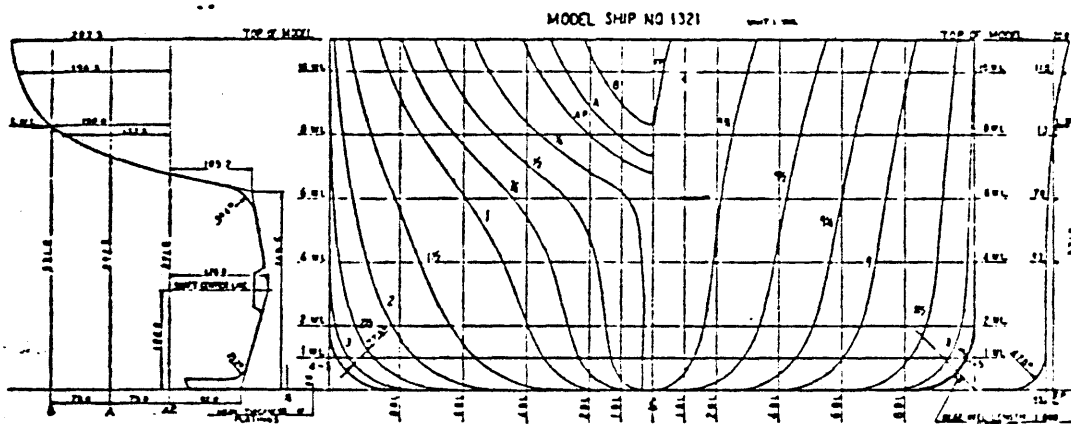
$$"wsa" = 1.81 \times "l" \times "dr" + "cb \times "l" \times "b"$$

And, the total resistance coefficient is determined by summing the residual resistance coefficient (obtained in the previous section), the frictional resistance coefficient (calculated by the ATTC formula via an iterative routine), and assumed roughness allowance of 0.0004.

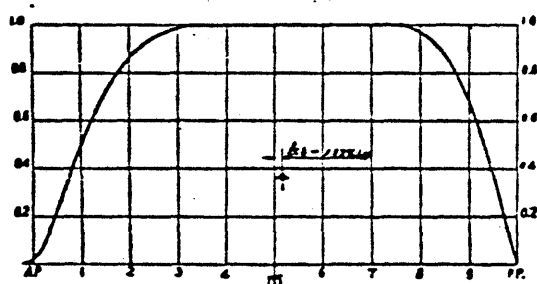
As for the propeller particulars, these have been assumed to agree with the propeller used in the tanker series given in Tsuchida (see Fig. 4.1). Specifically, the propeller is assumed to have four blades, a diameter equal to 0.636 of the barge draft, and an expanded area ratio estimated by a semi-empirical formula recommended by Wageningen for his B-screw series.

Following the determination of the above inputs, subroutine "prop" is called. As stated previously, this subroutine determines via a mathematical formulation of the

propeller series curves the propulsive efficiency of the propeller (η) and the horsepower required to be delivered to propeller ($DHP = EHP/\eta$). The subprogram "power" then terminates upon returning the value of "dhp" to the main program "drop-and-swap".

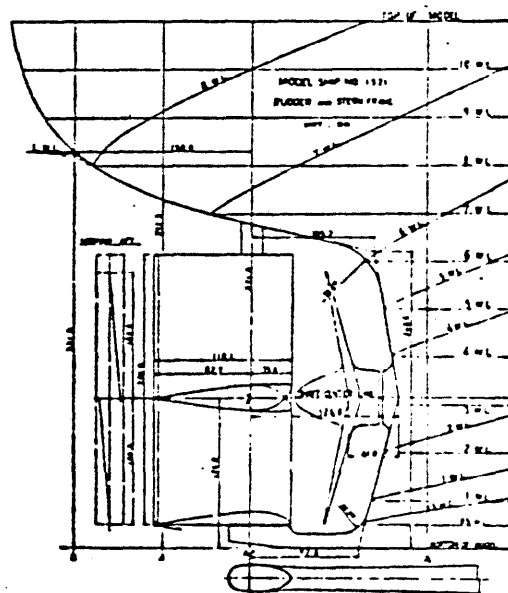


Body Plan, Stem & Stern Contour [M.S. No. 1321]



Prismatic Curve

Prismatic Curve of
M.S. No. 1321



Stern Frame & Rudder
of M.S. No. 1321

SOURCE: Tsuchida (1969)

FIGURE 4.1

TANK-SHIP HULL LINES USED IN OBTAINING RESIDUAL RESISTANCE
COEFFICIENT AND SELF-PROPULSION FACTOR DATA

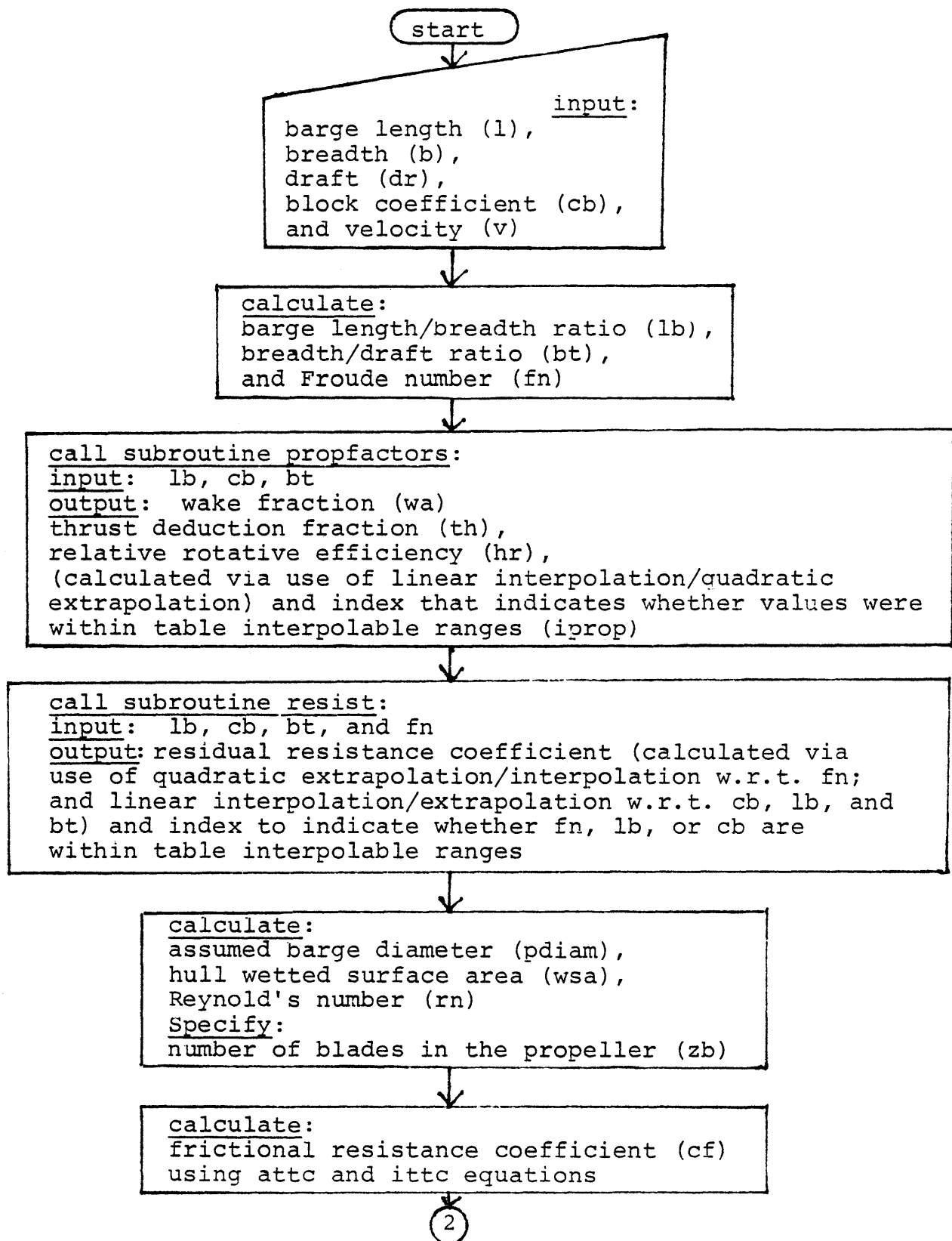


FIGURE 4.2

SUMMARY FLOW CHART FOR POWERING SUBPROGRAM

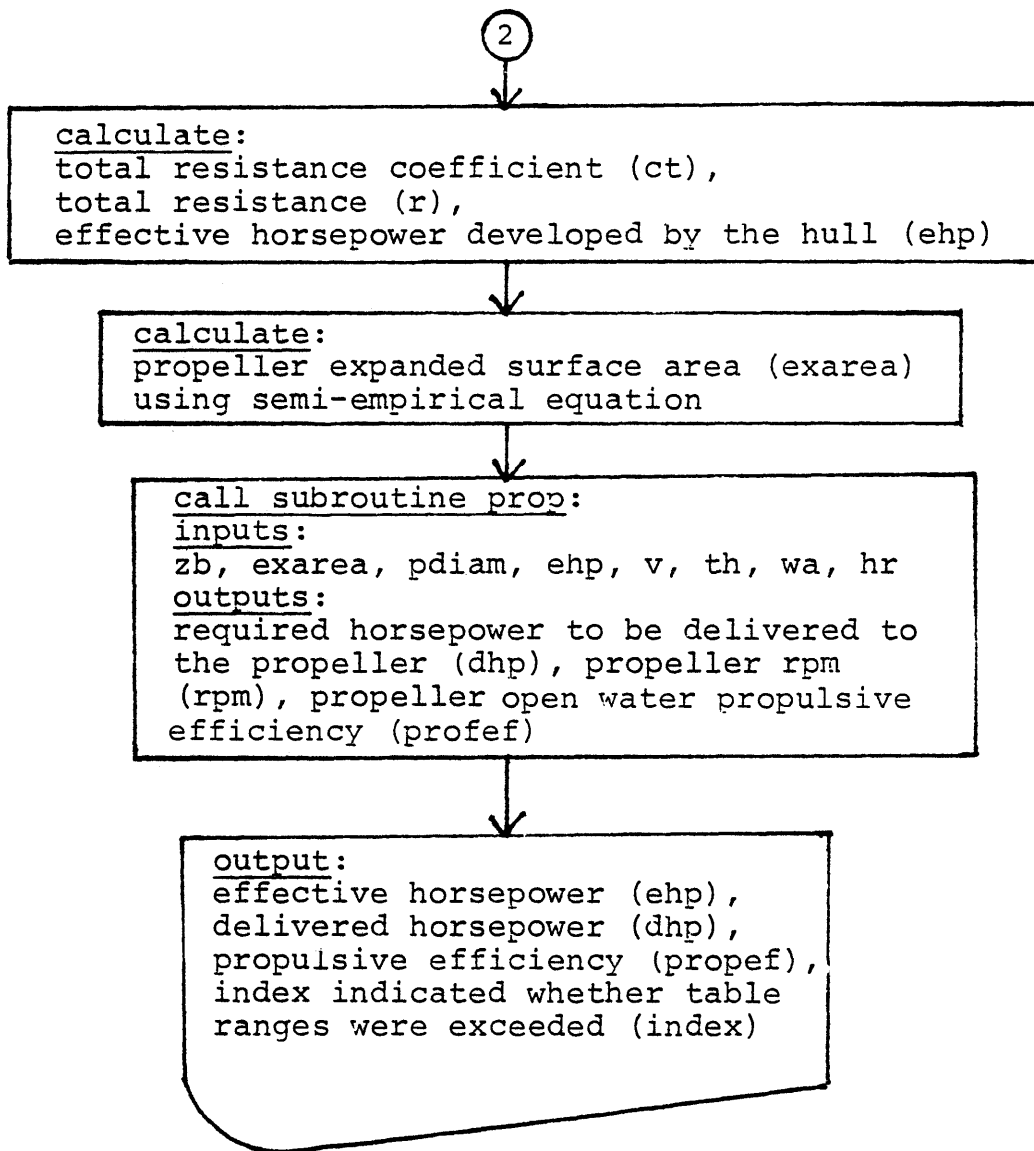


FIGURE 4.2...(Continued)

RESIDUAL RESISTANCE COEFFICIENT VS FRIC FRICTION NO

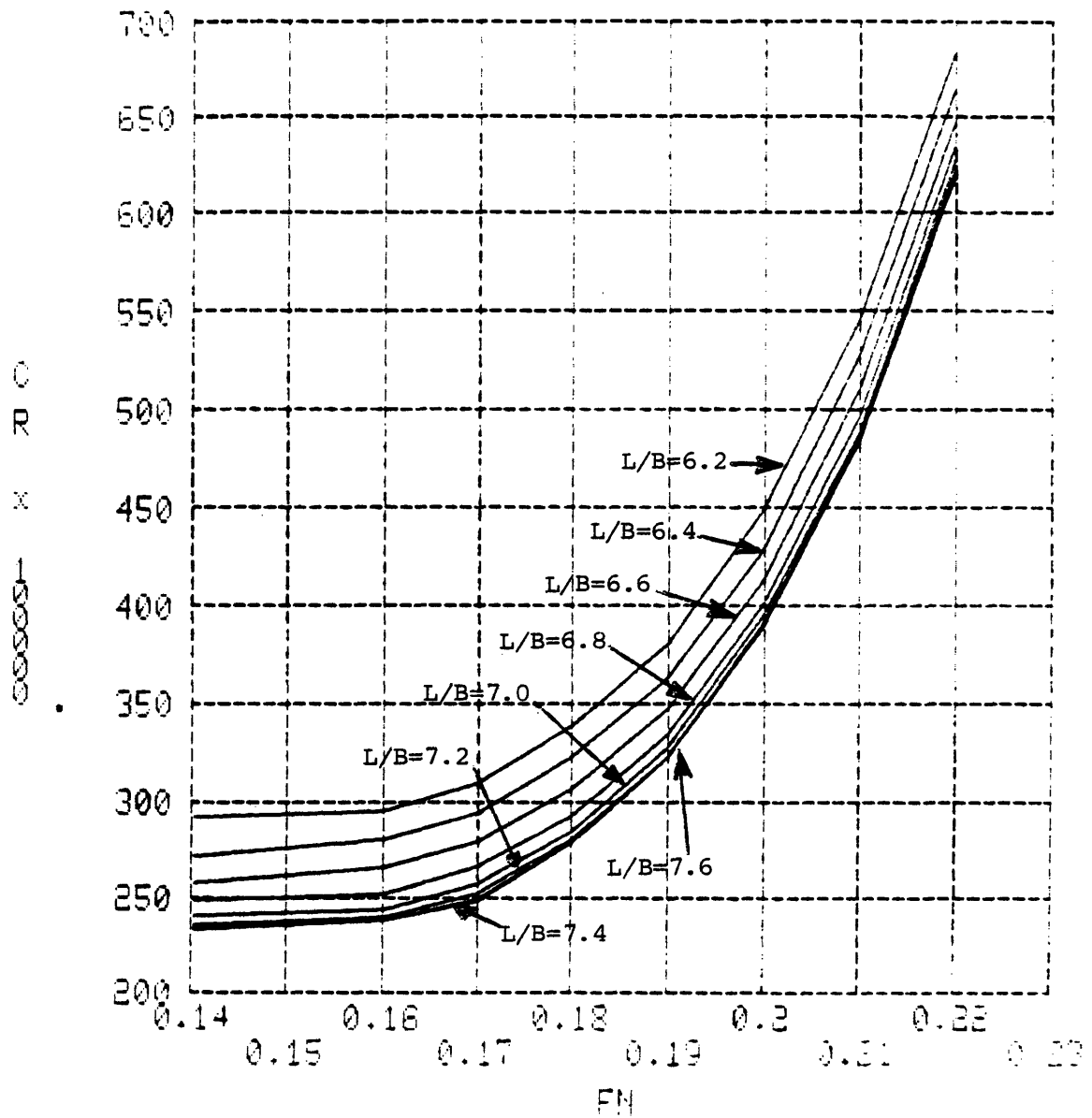


FIGURE 4.3

PLOT OF C_R VS. FN FOR $C_B=0.775$ AND $B/T=2.46$

RESIDUAL RESISTANCE COEFFICIENT C_R VS. FN

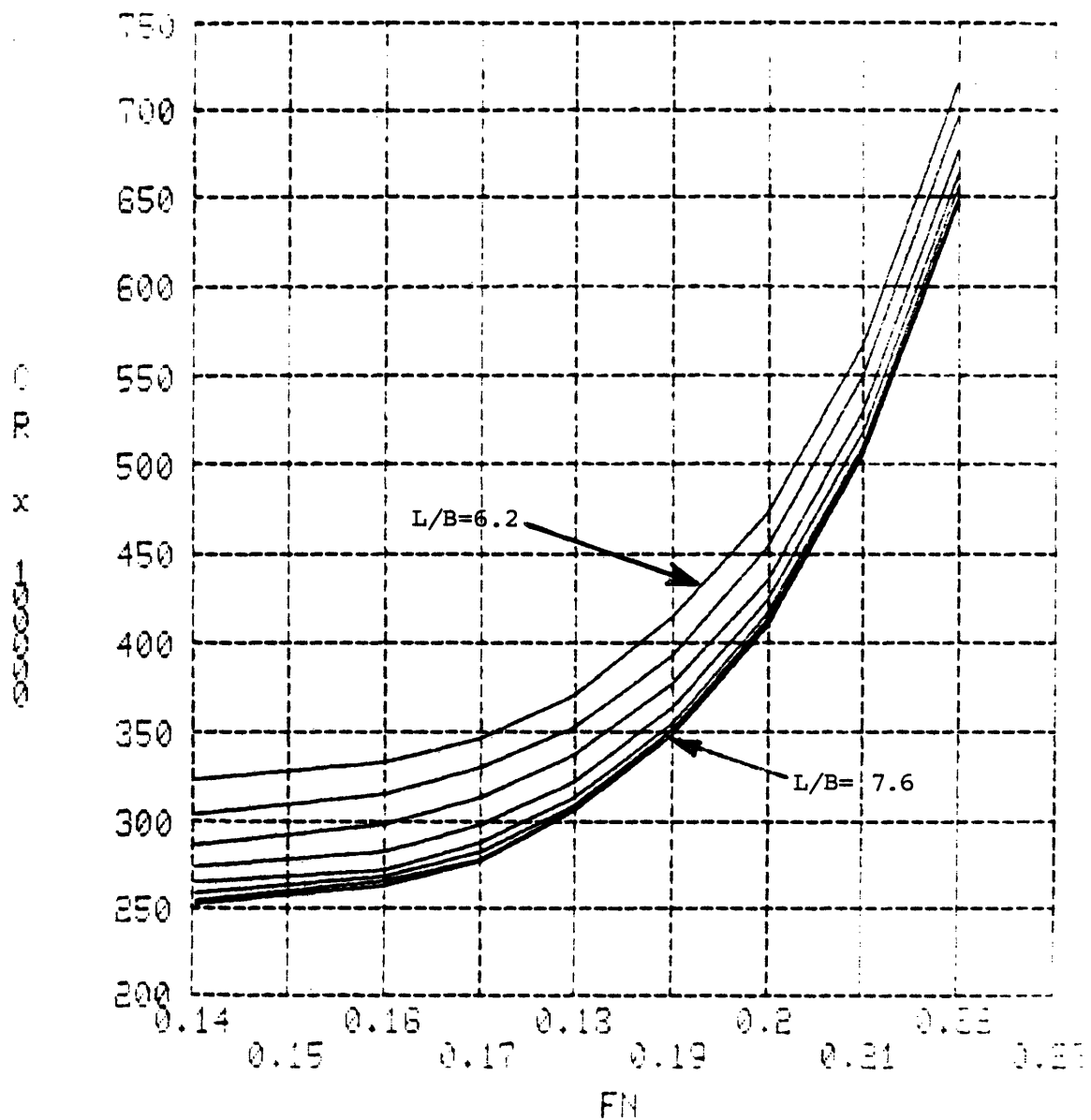


FIGURE 4.3

PLOT OF C_R VS. FN FOR $C_B=0.785$ AND $B/T=2.46$

RESIDUAL RESISTANCE COEFFICIENT C_R VS. FN

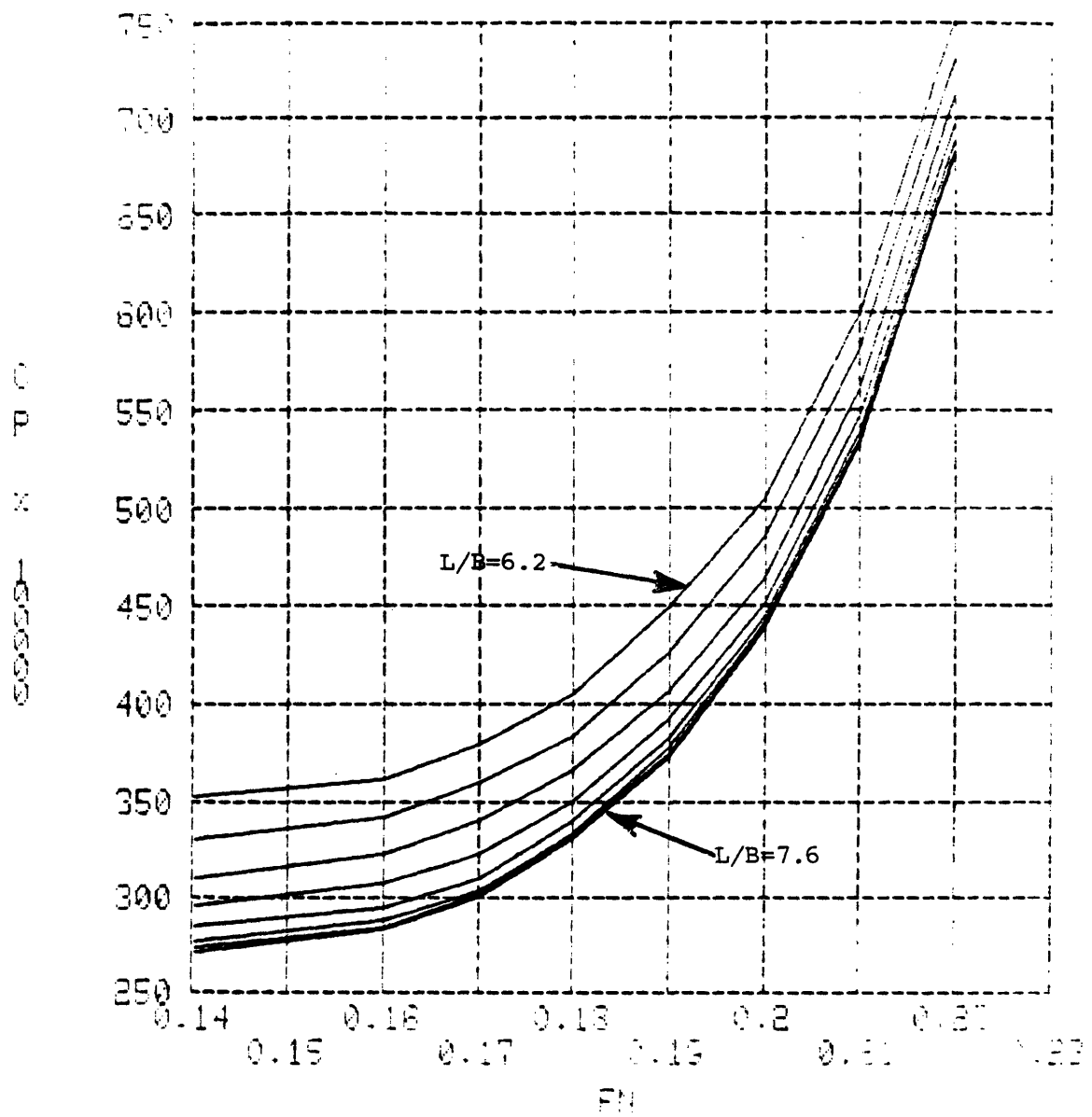


FIGURE 4.3

PLOT OF C_R VS. FN FOR $C_B=0.795$ AND $B/T=2.46$

PERIODAL RESISTANCE COEFFICIENT C_R VS. FN

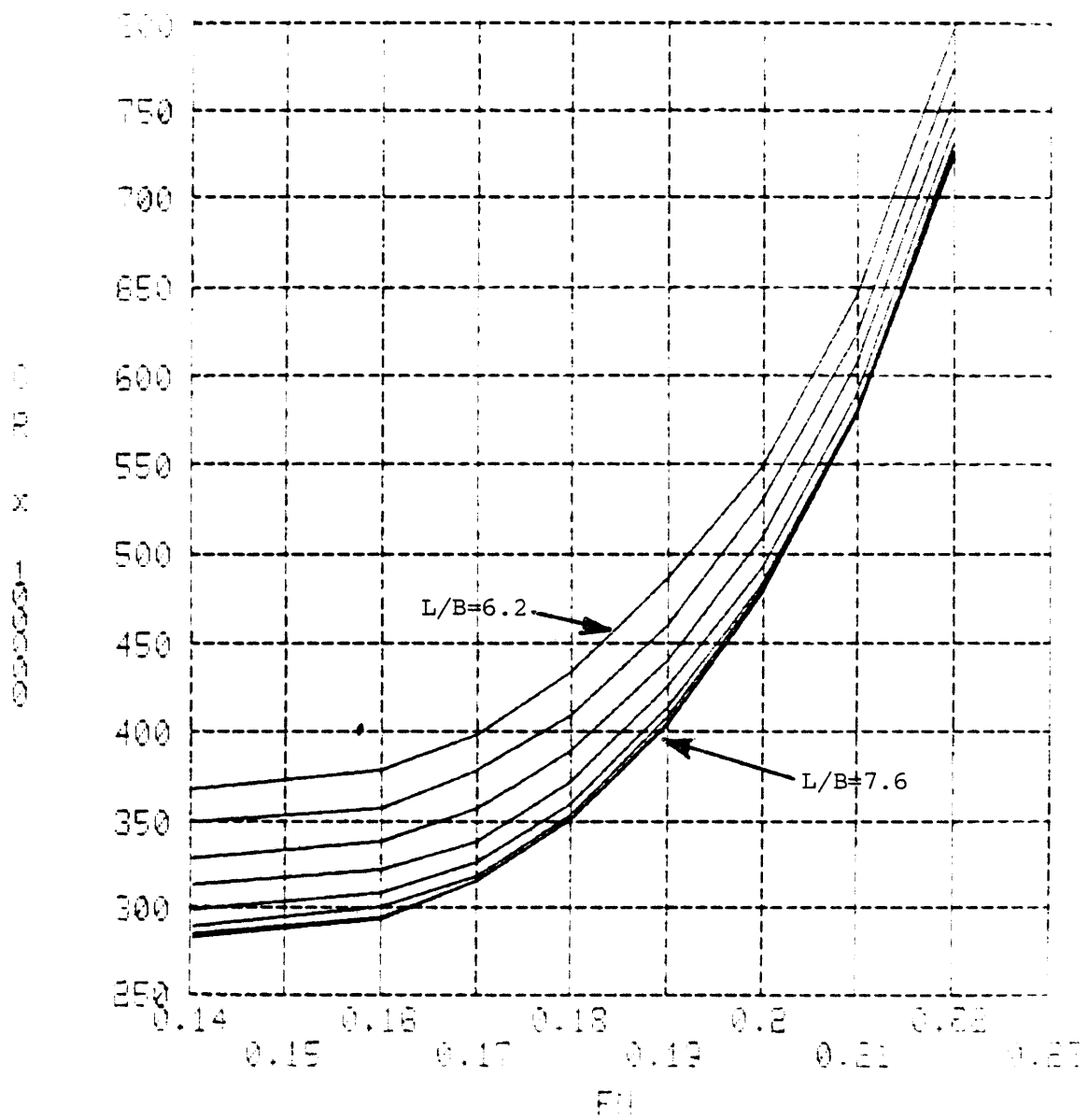


FIGURE 4.3

PLOT OF C_R VS. FN FOR $C_B=0.805$ AND $B/T=2.46$

PERIODAL RESISTANCE COEFFICIENT VS. FREQUENCY

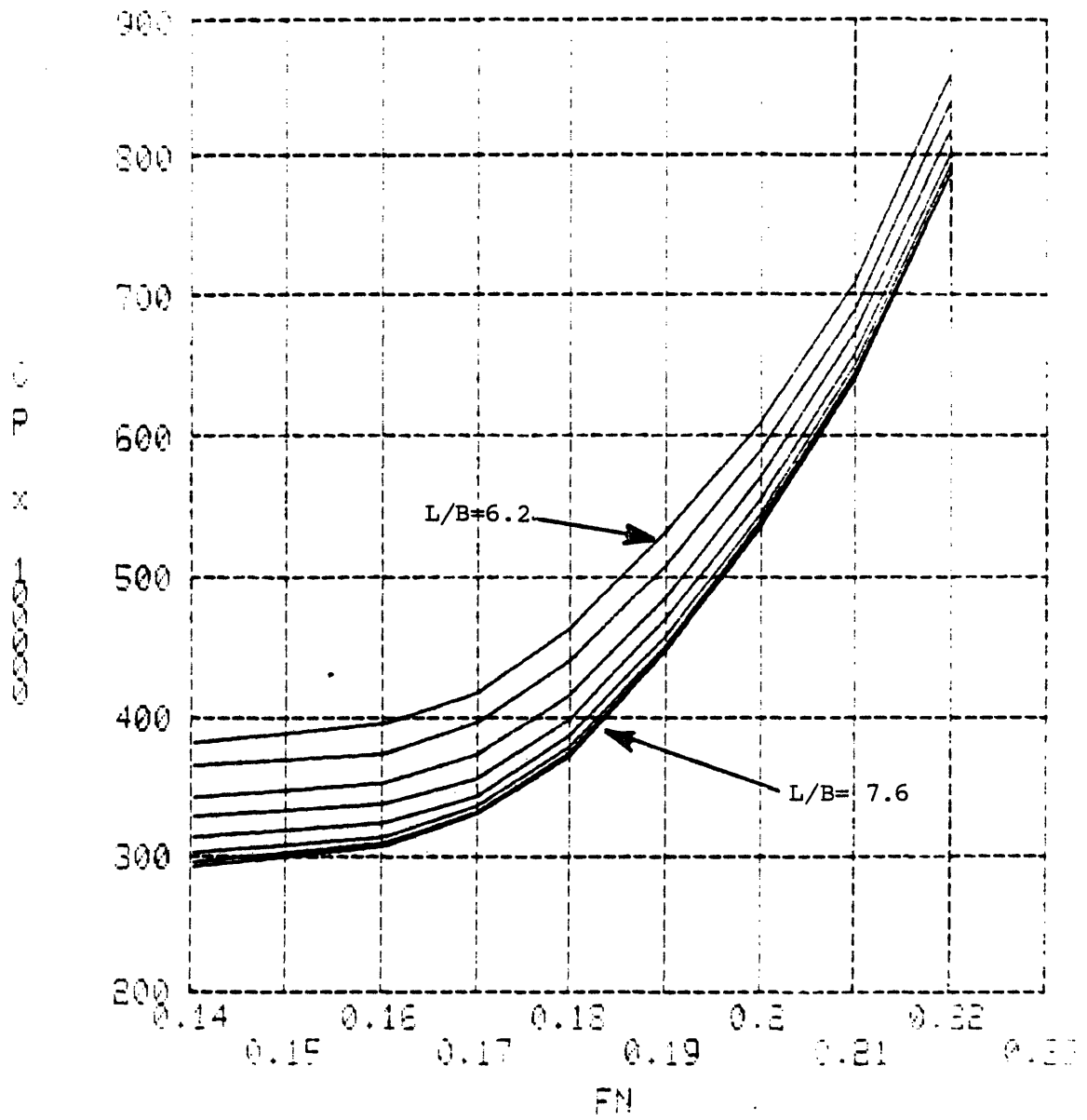


FIGURE 4.3

PLOT OF C_R VS. FN FOR $C_B=0.815$ AND $B/T=2.46$

RESIDUAL RESISTANCE COEFFICIENT VS FOLD DEPTH

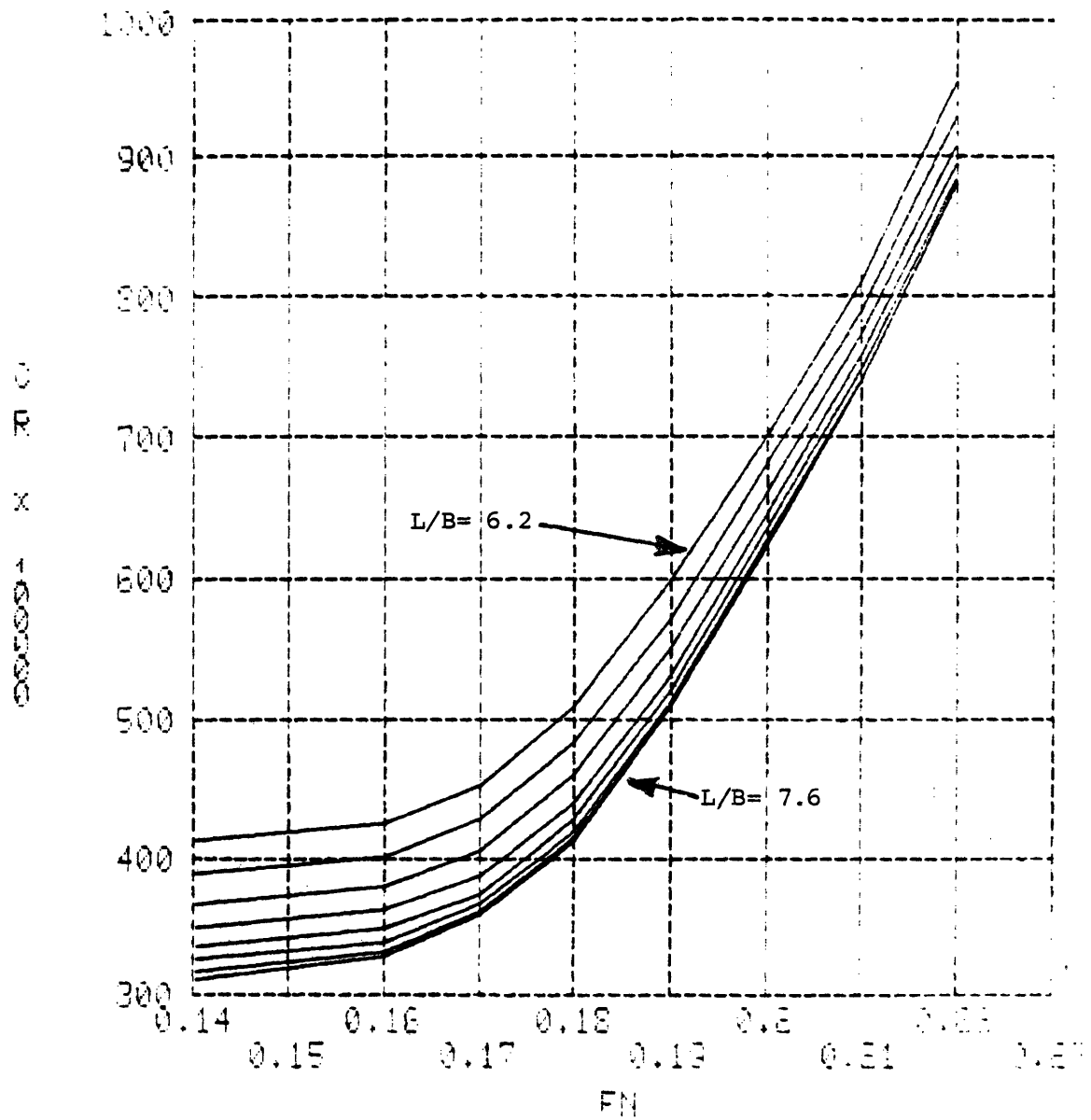


FIGURE 4.3

PLOT OF C_R VS. FN FOR $C_B=0.825$ AND $B/T=2.46$

RESIDUAL RESISTANCE COEFFICIENT VS. FRICTION

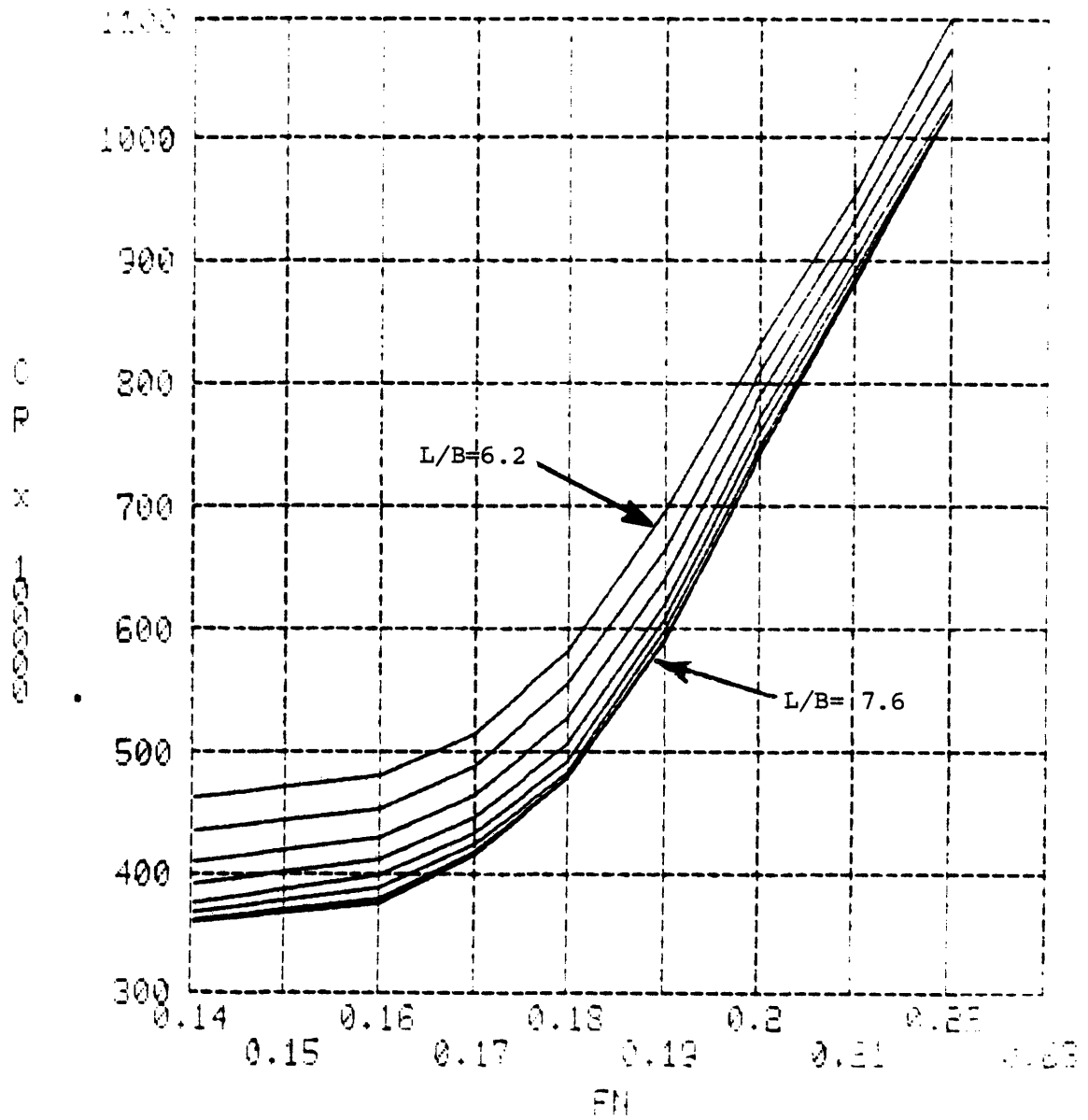


FIGURE 4.3

PLOT OF C_R VS. FN FOR $C_B=0.835$ AND $B/T=2.46$

RESIDUAL RESISTANCE COEFFICIENT VS. FRICTION NO.

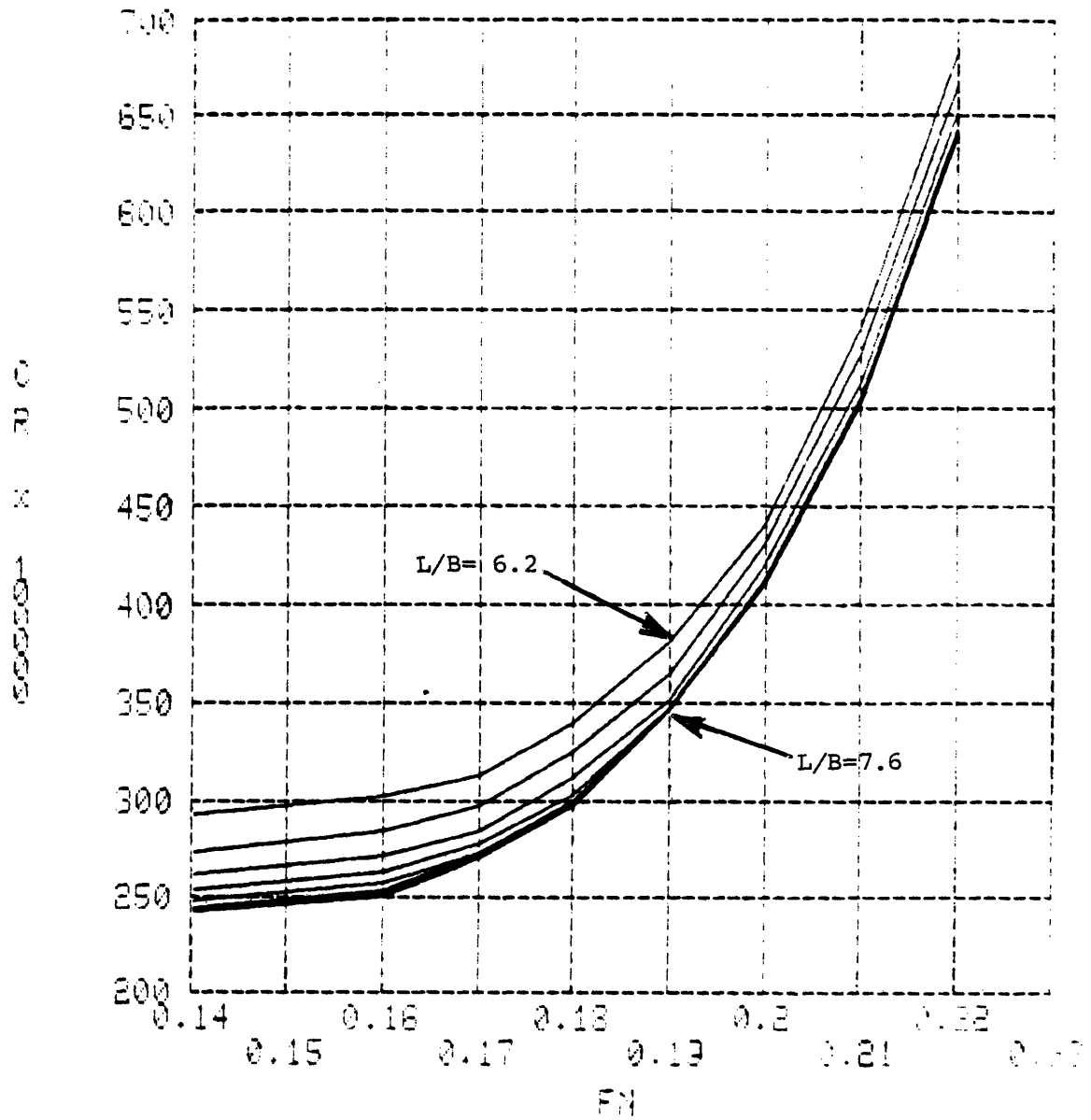


FIGURE 4.3

PLOT OF C_R VS. FN FOR $C_B=0.775$ AND $B/T=2.76$

RESIDUAL RESISTANCE COEFFICIENT VS. FOLDE NO

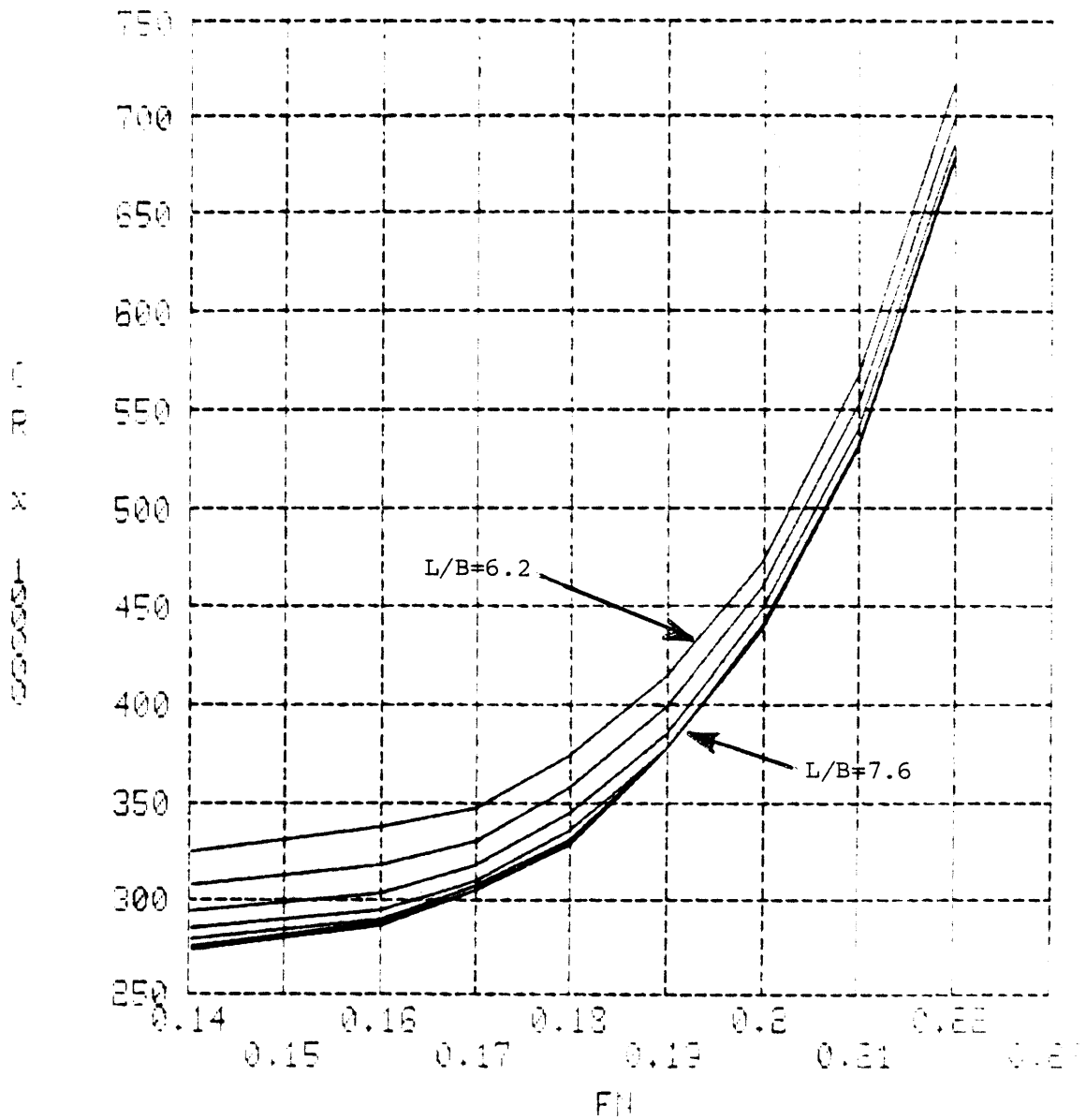


FIGURE 4.3

PLOT OF C_R VS. FN FOR $C_B=0.785$ AND $B/T=2.76$

RESIDUAL RESISTANCE COEFFICIENT S PROUDE NO

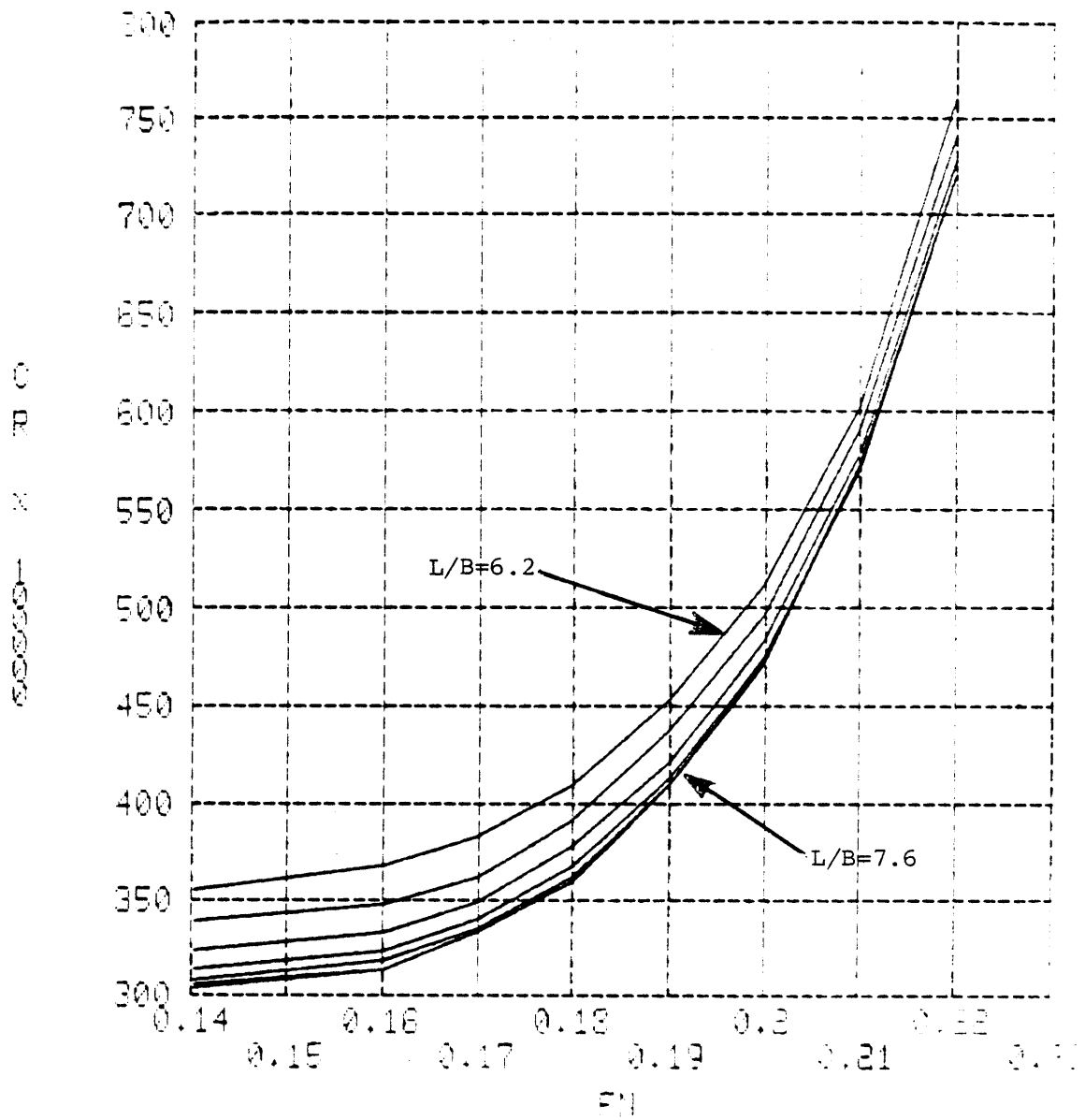


FIGURE 4.3

PLOT OF C_R VS. FN FOR $C_B=0.795$ AND $B/T=2.76$

RESIDUAL RESISTANCE COEFFICIENT VS. FREQUENCY

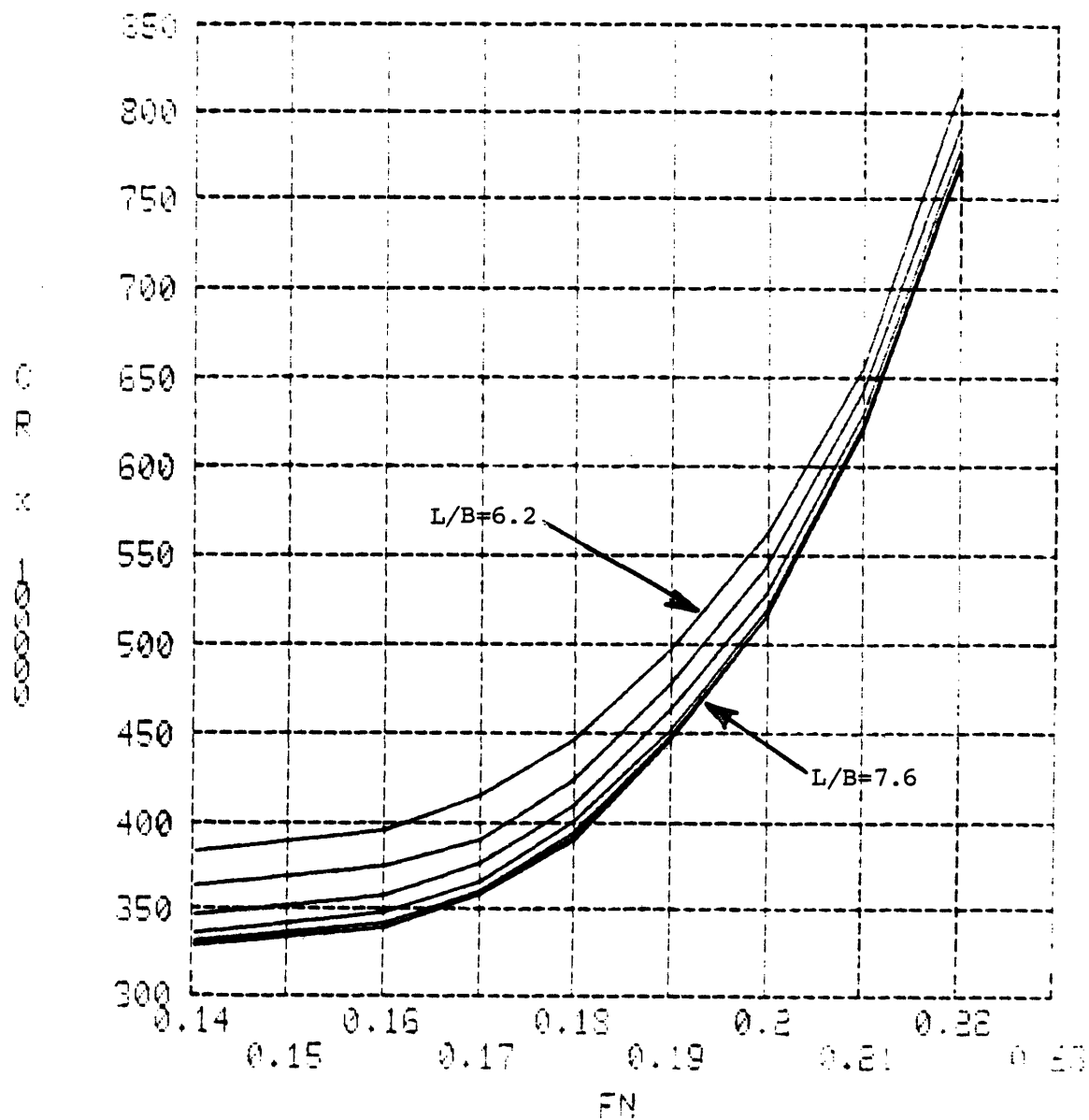


FIGURE 4.3

PLOT OF C_R VS. FN FOR $C_B=0.805$ AND $B/T=2.76$

RESIDUAL RESISTANCE COEFFICIENT S FOLDED 10

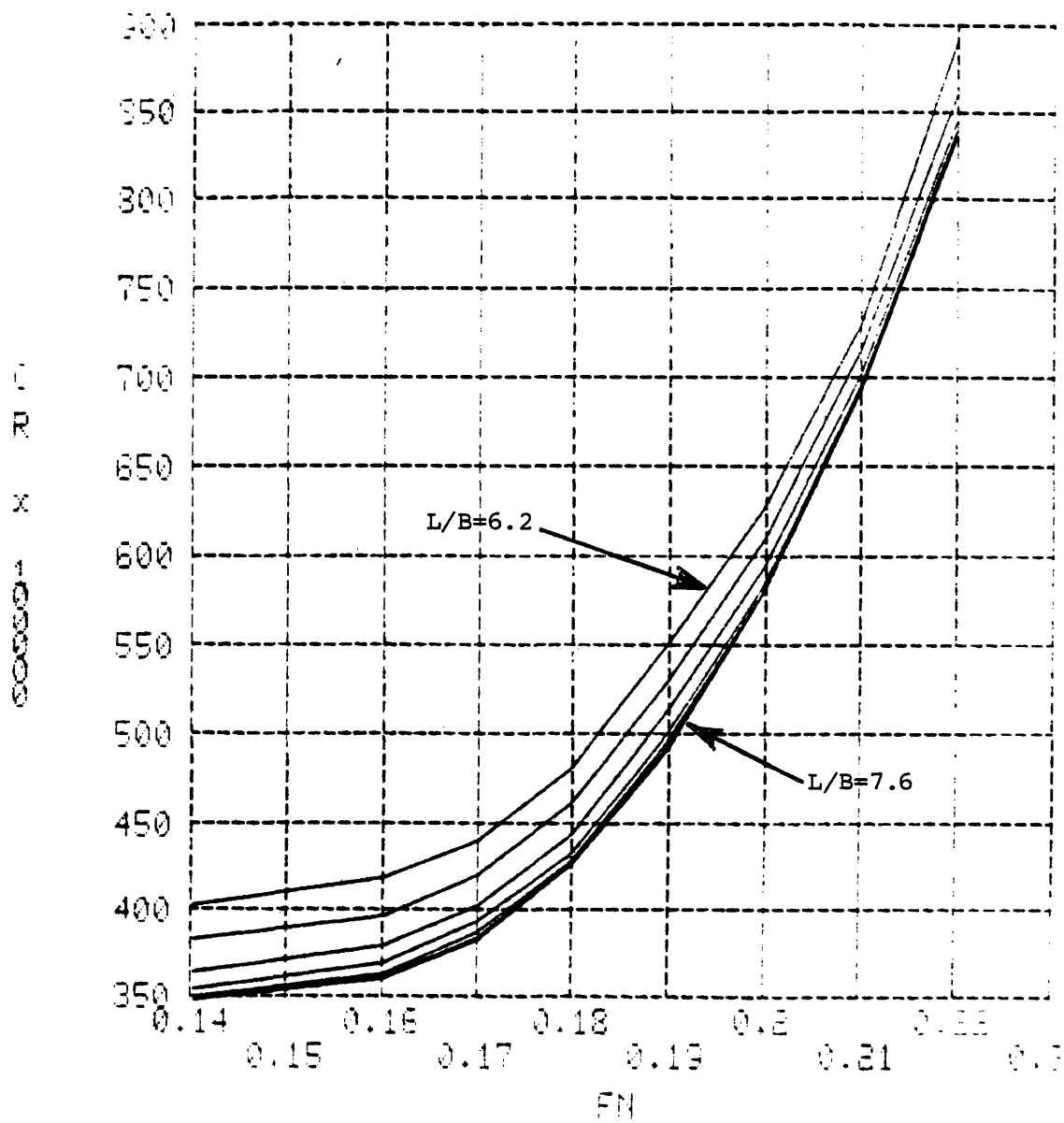


FIGURE 4.3

PLOT OF C_R VS. FN FOR $C_B=0.815$ AND $B/T=2.76$

RESIDUAL RESISTANCE COEFFICIENT VS FROUDE NO

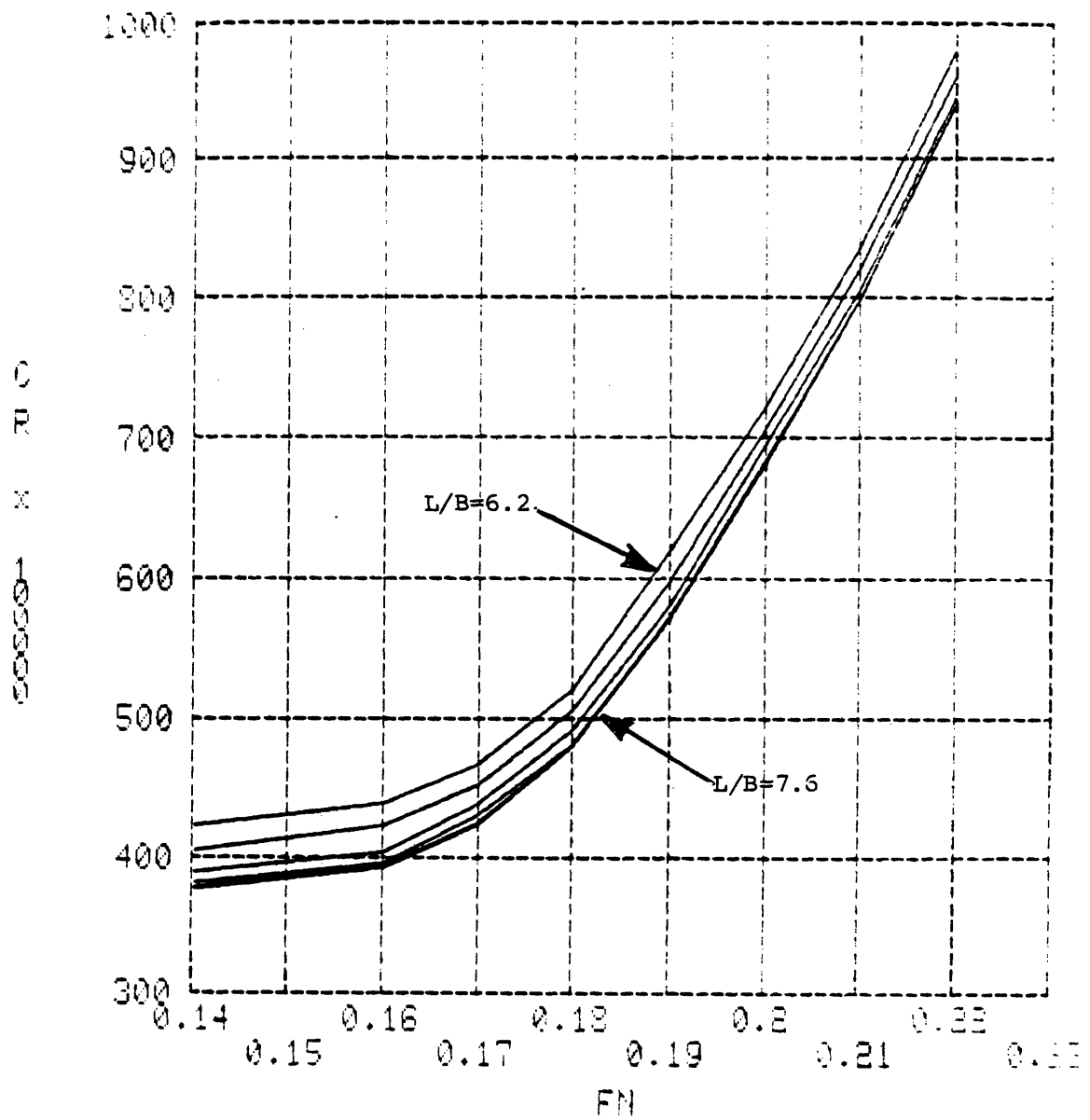


FIGURE 4.3

PLOT OF C_R VS. FN FOR $C_B=0.825$ AND $B/T=2.76$

RESIDUAL RESISTANCE COEFFICIENT VS FROUDE NO

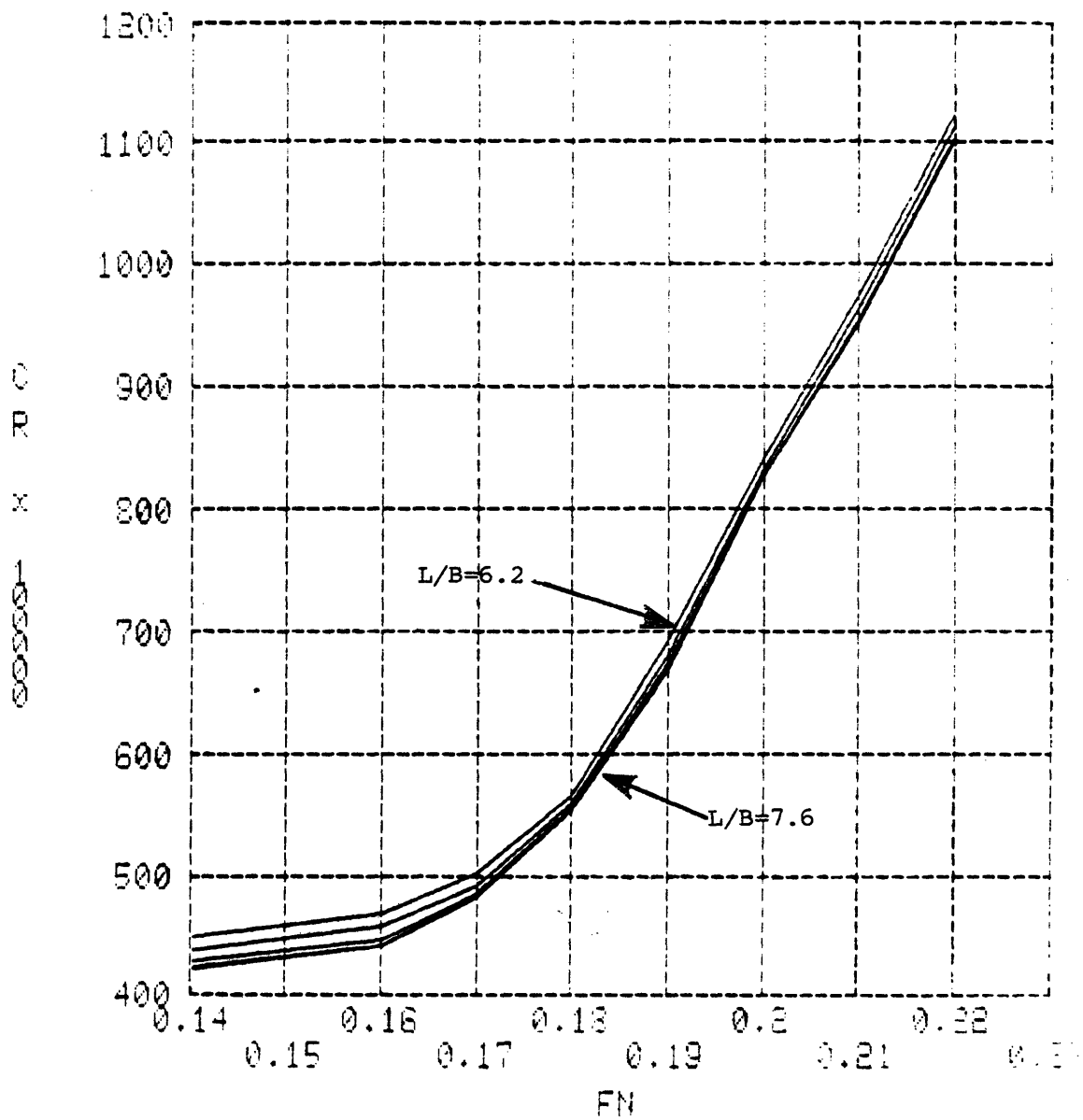


FIGURE 4.3

PLOT OF C_R VS. FN FOR $C_B = 0.835$ AND $B/T = 2.76$

TABLE 4.1

SELF PROPULSION FACTOR- W_T
WAKE FRACTION $((1-W_T) \times 1000)$

Block Coefficient	6.2	6.3	6.4	6.5	6.6	6.7	6.8	6.9	7.0	7.1	7.2	7.3	7.4	7.5	7.6
0.770	543	550	558	565	572	580	587	594	601	608	615	622	630	637	644
0.775	540	547	554	561	568	575	582	589	596	603	610	617	624	631	638
0.780	536	543	550	557	563	570	577	584	591	598	605	612	619	626	633
0.785	533	540	547	553	560	566	572	579	586	592	599	606	613	619	626
0.790	530	536	542	548	555	561	568	574	580	587	593	600	606	612	619
0.795	526	532	538	544	550	557	563	569	575	581	588	594	600	606	612
0.800	523	529	535	540	546	552	558	564	570	576	582	588	594	600	606
0.805	520	526	531	536	542	548	553	559	565	570	576	582	588	593	599
0.810	518	523	528	532	538	543	549	555	560	565	570	576	582	587	592
0.815	516	520	525	529	533	538	543	549	554	560	565	570	576	581	586
0.820	515	518	522	526	530	534	539	543	549	554	560	565	570	575	580
0.825	514	517	520	523	527	531	535	539	544	549	553	558	563	568	573
0.830	513	515	517	520	523	527	531	535	539	543	548	553	558	562	567
0.835	512	513	515	518	521	524	528	531	535	538	542	546	551	556	560
0.840	511	512	513	515	518	521	524	527	530	534	538	541	545	550	554

TABLE 4.1

SELF PROPULSION FACTOR-t
THRUST DEDUCTION FRACTION ((1-t)x1000)

Block Coefficient	6.2	6.3	6.4	6.5	6.6	6.7	6.8	6.9	7.0	7.1	7.2	7.3	7.4	7.5	7.6
0.770	769	772	775	778	782	784	787	790	793	797	800	803	806	809	812
0.775	769	772	775	778	781	784	787	790	793	796	799	802	805	808	811
0.780	769	772	775	778	781	784	787	790	793	796	799	802	805	808	811
0.785	769	772	775	777	781	784	787	790	792	796	798	801	804	807	810
0.790	768	771	774	777	780	783	786	789	792	795	798	801	804	807	810
0.795	768	771	774	777	780	783	786	788	792	795	797	800	803	806	809
0.800	768	770	773	776	779	782	785	788	791	794	797	800	802	805	808
0.805	767	770	773	776	778	782	785	787	790	793	796	799	802	805	807
0.810	766	769	772	775	778	781	784	787	789	792	795	798	801	804	807
0.815	765	768	771	774	777	780	782	785	788	791	794	797	800	802	806
0.820	764	767	770	773	775	778	781	784	787	790	793	795	798	801	804
0.825	763	765	768	771	774	777	780	782	785	788	791	794	797	800	802
0.830	761	763	766	769	772	775	778	780	783	786	789	792	795	797	800
0.835	758	761	764	767	770	773	775	778	781	784	787	789	792	795	798
0.840	755	759	761	764	767	770	772	775	778	781	784	786	789	792	795

TABLE 4.1

SELF PROPULSION FACTOR- η_R
RELATIVE ROTATING EFFICIENCY (X1000)

Block Coefficient	6.2	6.3	6.4	6.5	6.6	6.7	6.8	6.9	7.0	7.1	7.2	7.3	7.4	7.5	7.6
0.770	998	1003	1007	1010	1013	1016	1017	1017	1017	1016	1015	1013	1011	1008	1004
0.775	1000	1005	1009	1013	1016	1018	1019	1020	1020	1019	1018	1016	1014	1011	1007
0.780	1003	1007	1011	1015	1018	1021	1022	1023	1023	1022	1021	1019	1017	1014	1010
0.785	1005	1009	1014	1018	1021	1023	1024	1025	1025	1025	1024	1022	1020	1017	1013
0.790	1007	1012	1016	1020	1023	1026	1027	1028	1028	1028	1027	1025	1023	1020	1015
0.795	1009	1014	1018	1022	1026	1028	1030	1031	1031	1031	1030	1028	1026	1023	1018
0.800	1011	1016	1021	1025	1028	1031	1033	1033	1034	1033	1032	1031	1029	1025	1021
0.805	1013	1018	1023	1027	1031	1034	1035	1036	1037	1036	1035	1034	1032	1028	1024
0.810	1015	1020	1025	1030	1034	1036	1038	1039	1039	1039	1038	1037	1035	1031	1027
0.815	1017	1022	1027	1033	1036	1039	1041	1042	1042	1042	1041	1039	1037	1034	1030
0.820	1020	1025	1030	1035	1039	1042	1043	1044	1045	1045	1044	1042	1040	1037	1033
0.825	1022	1027	1032	1037	1041	1044	1046	1047	1048	1048	1047	1045	1043	1040	1036
0.830	1024	1029	1035	1040	1043	1046	1048	1050	1050	1050	1050	1048	1046	1043	1039
0.835	1026	1031	1037	1042	1046	1049	1051	1053	1053	1053	1052	1051	1049	1046	1042
0.840	1028	1033	1039	1045	1048	1051	1054	1055	1056	1056	1055	1054	1052	1048	1045

TABLE 4.2
FULL-BODIED SINGLE-SCREW RESIDUAL RESISTANCE COEFFICIENTS
FOR BREADTH/DRAFT = 2.46 (x 10000)

Froude Number	Block Coefficient	Length/Breadth							
		6.2	6.4	6.6	6.8	7.0	7.2	7.4	7.6
0.14	0.775	291	271	257	248	240	235	233	232
	0.785	323	303	286	273	264	258	254	252
	0.795	352	330	310	295	284	276	272	270
	0.805	368	349	328	312	298	288	284	282
	0.815	381	365	342	327	313	301	294	291
	0.825	412	388	364	348	334	324	315	310
	0.835	463	435	409	390	375	367	360	359
0.16	0.775	295	280	265	251	243	239	237	237
	0.785	333	315	298	285	272	268	265	263
	0.795	361	341	322	307	294	287	283	282
	0.805	378	357	338	322	308	300	294	292
	0.815	395	373	352	337	323	313	308	306
	0.825	425	400	378	361	348	337	330	327
	0.835	481	453	429	411	398	387	379	374
0.17	0.775	309	294	279	266	257	252	249	248
	0.785	346	330	313	298	288	282	278	277
	0.795	379	360	340	322	310	303	301	300
	0.805	398	378	357	338	326	318	316	315
	0.815	417	396	373	355	343	336	332	330
	0.825	452	428	405	387	373	366	360	358
	0.835	515	488	465	447	433	424	418	415
0.18	0.775	338	322	306	292	284	280	278	278
	0.785	371	353	338	322	313	309	307	306
	0.795	405	383	366	350	340	333	332	331
	0.805	433	409	389	372	360	353	351	350
	0.815	462	439	415	398	386	378	374	372
	0.825	509	483	460	440	427	419	414	412
	0.835	582	555	528	507	492	483	480	479

TABLE 4.2
FULL-BODIED SINGLE SCREW RESIDUAL RESISTANCE COEFFICIENTS
FOR BREADTH/DRAFT = 2.46 (x 1000)

Froude Number	Block Coefficient	Length/Breadth							
		6.2	6.4	6.6	6.8	7.0	7.2	7.4	7.6
0.19	0.775	380	363	347	334	328	323	322	322
	0.785	414	393	377	363	355	351	348	348
	0.795	449	426	406	392	382	377	374	373
	0.805	486	460	440	425	414	408	405	404
	0.815	531	507	484	469	457	450	447	446
	0.825	598	571	550	531	520	511	509	508
	0.835	696	665	641	620	608	599	591	590
0.20	0.775	448	428	412	400	394	390	388	388
	0.785	472	453	434	423	416	412	410	409
	0.795	504	484	463	450	443	440	438	437
	0.805	549	530	509	491	482	479	477	477
	0.815	609	590	570	554	543	538	535	534
	0.825	700	680	660	644	633	626	623	621
	0.835	830	808	788	770	758	746	740	740
0.21	0.775	545	527	510	496	488	486	485	484
	0.785	567	549	530	518	510	507	505	505
	0.795	599	580	560	546	538	534	532	532
	0.805	645	624	607	591	582	579	578	578
	0.815	709	690	673	659	650	644	642	641
	0.825	810	790	773	758	748	741	740	740
	0.835	952	933	915	900	890	884	882	880
0.22	0.775	684	664	647	635	628	623	620	619
	0.785	715	696	677	663	656	650	648	647
	0.795	752	731	712	697	688	683	682	681
	0.805	797	774	756	740	731	727	724	723
	0.815	859	838	818	801	793	788	788	788
	0.825	955	930	909	895	885	881	880	880
	0.835	1100	1075	1052	1032	1026	1025	1025	1025

TABLE 4.2
FULL-BODIED SINGLE-SCREW RESIDUAL RESISTANCE COEFFICIENTS
FOR BREADTH/DRAFT = 2.76 (x 10000)

Froude Number	Block Coefficient	Length/Breadth							
		6.2	6.4	6.6	6.8	7.0	7.2	7.4	7.6
0.14	0.775	292	273	261	253	247	243	242	241
	0.785	324	307	293	284	278	274	273	272
	0.795	355	338	323	313	307	304	303	303
	0.805	383	363	346	335	330	328	327	327
	0.815	402	382	363	353	348	347	347	347
	0.825	422	404	388	379	376	375	375	375
	0.835	448	437	427	422	420	420	420	420
0.16	0.775	302	284	271	263	257	253	251	250
	0.785	337	318	303	294	289	287	286	286
	0.795	368	347	332	323	317	313	312	312
	0.805	395	374	357	347	341	338	338	338
	0.815	418	396	378	368	362	360	359	359
	0.825	439	422	403	395	392	391	391	391
	0.835	468	457	445	440	440	440	440	440
0.17	0.775	313	297	284	277	273	272	271	270
	0.785	347	330	317	310	307	305	304	304
	0.795	383	362	349	340	335	333	333	333
	0.805	415	390	376	366	360	358	358	358
	0.815	440	420	402	392	386	383	382	382
	0.825	467	452	438	430	425	424	423	423
	0.835	502	492	485	483	482	482	482	482
0.18	0.775	340	325	312	303	300	298	297	297
	0.785	374	358	345	336	331	329	328	328
	0.795	410	391	378	368	362	360	359	359
	0.805	446	424	410	400	394	392	390	390
	0.815	481	462	443	433	428	427	426	426
	0.825	520	505	491	481	480	480	480	480
	0.835	566	560	556	554	554	554	554	554

TABLE 4.2
FULL-BODIED SINGLE-SCREW RESIDUAL RESISTANCE COEFFICIENTS
BREADTH/DRAFT = 2.76 (x 10000)

Froude Number	Block Coefficient	Length/Breadth							
		6.2	6.4	6.6	6.8	7.0	7.2	7.4	7.6
0.19	0.775	381	364	351	347	346	346	346	346
	0.785	414	399	385	378	377	377	377	377
	0.795	453	437	421	413	410	410	410	410
	0.805	497	478	463	452	448	447	447	447
	0.815	551	530	513	501	496	493	492	492
	0.825	618	596	580	572	570	570	570	570
	0.835	692	681	674	670	669	669	669	669
0.20	0.775	440	430	419	412	411	410	410	410
	0.785	472	460	448	440	438	437	437	437
	0.795	511	496	483	474	472	472	471	471
	0.805	561	543	528	519	515	515	515	515
	0.815	627	609	593	583	580	580	580	580
	0.825	721	706	694	683	680	680	680	680
	0.835	840	830	825	825	825	825	825	825
0.21	0.775	541	527	513	505	503	502	502	502
	0.785	568	553	540	532	530	530	530	530
	0.795	602	590	578	570	572	572	572	572
	0.805	655	642	630	623	620	620	620	620
	0.815	729	715	701	695	693	692	692	692
	0.825	835	820	805	799	798	798	798	798
	0.835	972	962	952	950	950	950	950	950
0.22	0.775	683	665	651	643	640	639	639	639
	0.785	717	699	686	679	677	677	677	677
	0.795	760	741	727	720	720	720	720	720
	0.805	813	791	779	772	769	769	769	769
	0.815	891	860	845	839	836	836	836	836
	0.825	980	960	945	940	940	940	940	940
	0.835	1123	1112	1103	1100	1100	1100	1100	1100

CHAPTER 5

DROP-AND-SWAP MODEL RESULTS, OBSERVATIONS, AND SUGGESTED REFINEMENTS

5.1 Base Case Results

At this point, the structure and details of the drop-and-swap model, including the barge powering and design subprograms, have been discussed. It is now time to consider what can be learned from the model. To do this, the model was run for five one-way trades with annual cargo flows of 100,000; 600,000; 1,000,000; 6,000,000; and 10,000,000 long tons. The user specified inputs and the values of the semi-fixed parameters used in these runs are shown in Table 5.1 and Figure 2.5, respectively. Essentially, in these base case runs the model finds the optimum barge size, speed, and form within the system parameter ranges specified in Table 2.1 for port pair trades with port separation distances of 500, 2000, 3500 and 5000 nautical miles (NM) and for a reasonable range of loading/discharging rates. In these base case trades, it is assumed there are no draft or beam restrictions. Barge length, however, is limited to 750 feet since the ABS

rules used in the barge design model are applicable only to barges less than this length. Additionally, barge size is limited to 100,000 DWT since this is the maximum size that designers have even considered for barge construction. (1) Finally, it is assumed that there are no costs associated with terminal loading/discharging or storage facilities. The value of the cargo is assumed to be \$200 per long ton and the other cost parameters are as discussed in Chapter 2.

Graphical output from these runs of the required freight rate as a function of loading/discharging (L-D) rate for four port separation distances (500, 2000, 3500 and 5000 nautical miles) is shown in Figures 5.1-5.5. The printed outputs from these runs are voluminous and so cannot be included here. However, some of the results, including the range of optimum barge deadweights and tug speeds, have been extracted and presented in Table 5.2. In addition, some observations that can be made concerning all the runs are presented in the next subsection. This is followed by a discussion of points of interest that pertain to an individual run. And, this is followed by some conclusions that can be drawn from the whole series of base runs.

(1) The largest barge currently in operation is the Breit/Ingram tug-barge system Presque Isle of 53,000DWT operating on the Great Lakes. The largest ocean going barges will be the 47,000 DWT CATUG tank barges being built for Aramada Hess.

5.1.1 General Observations Pertaining to All Base Case Runs

The following general observations can be made after examining the printed output of the base case runs:

1. The optimum barge form for all the port pair trades has the smallest allowable length-breadth (6.0) and breadth-draft (2.0) ratios. These ratios result in the shortest barge with the least amount of hull steel for a given deadweight capacity. (1) Apparently, at the slow speeds that these tug-barge systems operate, the capital cost savings achieved by constructing short, blunt barges outweigh any cost penalties associated with the higher residual resistance of these forms.

2. The optimum block coefficient for all the port pair trades varies from 0.78 to 0.81. The 0.78 value is usually associated with tug-barge units advancing at eleven knots or faster while the 0.81 value is usually found with systems advancing at nine knots or slower. However, for small tug-barge units, less than 15,000 DWT, a block coefficient of as low as 0.78 may be found for speeds as low as nine knots.

3. Port pair trades with longer port separation distances and faster L-D rates usually have optimum systems with larger barges.

4. Port pair trades with longer port separation distances and lower L-D rates usually have optimum systems with a

(1) This result can be verified by examining the elasticity exponents of the regression equations presented in Table 3.13.

greater number of tug and barge units.

Observations that are peculiar to an individual base case and not already summarized in Table 5.2 are presented in the following four subsections.

5.1.2 Base Case Results: Annual Cargo Flows of 100,000 LT

When annual cargo flows are as low as in this case, all of the cargo can usually be moved less expensively in a single small barge operating in the ship-like integral mode. The drop-and-swap mode is not competitive due to the extra barge or barges needed to be stationed in port. As can be seen from Figure 5.1, the only time that the drop-and-swap mode can be of advantage is when L-D rates are very low (less than 2000 LT/day) in trades with long port separation distances. In these trades, the extensive in port time forces integral mode operations to require more than one tug unit to handle the annual cargo flows. This makes integral mode operation uncompetitive compared to drop-and-swap mode operation with only one tug.

5.1.3 Base Case Results: Annual Cargo Flows of 600,000 and 1,000,000 LT

Both of these cases are similar in that for port pair trades with low L-D rates, the drop-and-swap mode will be less expensive. Conversely, for those trades having high L-D rates, the integral mode will be less expensive. As seen in

Figures 5.2 and 5.3, the major difference between these cases is that for the 1,000,000 LT case, the L-D rate tradeoff points are usually larger than for the 600,000 LT case. It was observed from both cases that for a given port separation distance, the L-D rate tradeoff point between the two modes usually occurs when the optimal integral mode begins to require only one more tug than the optimal drop-and-swap mode. Then, above the tradeoff L-D rate, the extra cost of the barges needed to remain in port for drop-and-swap mode operation begin to outweigh the capital cost savings resulting from the one fewer tug.

5.1.4 Base Case Results: Annual Cargo Flows of 6,000,000 LT

As seen in Figure 5.4, for this case the drop-and-swap mode will be the most economical method of operation for all port pair trades with port separation distance greater than 3500 NM. This is because with large cargo flows, the savings resulting from economies of scale push the optimum barge size to the upper constraint of 100,000 DWT. Due to this constraint, the optimum number of integral tug-barge units cannot be reduced (at economical speeds of operation) sufficiently fewer than the number of drop-and-swap tugs to make the integral mode economically competitive. Thus, when barge size is constrained, the higher tug utilization at a given speed achieved by the drop-and-swap mode of operation outweighs any barge capital cost savings inherent in the inte-

gral mode. For the shorter port separation distance (500 and 2000 nautical miles) the barge size is not binding so that the integral mode does become more economical, but at comparatively high L-D rates. It should be noted that when barge size is not binding the optimal system speeds vary between seven and eleven knots. However, when the barge size approaches the upper limit, the optimum integral mode speeds become as high as thirteen knots.

5.1.5 Base Case Results: Annual Cargo Flows of 10,000,000 LT

In this case the barge size constraint causes the drop-and-swap mode to be the operating mode of choice for all port pair trades except for some with port separation distance of 500 NM. For those trades, the integral mode will be favored when L-D rates exceed 45,000 LT/day. Again, it should be noted that barge size constraints have forced the optimum integral mode systems to operate with speeds of up to fifteen knots.

5.1.6 Conclusions From Base Case Runs

After reviewing all the output from the base case runs, a few general conclusions can be drawn concerning tug-barge systems operating on a port pair trade in either the drop-and-swap or integral mode. The major conclusion is that port pair trades can be broken up into three groups, primarily based on the amount of annual cargo flow and secondarily based

on the port separation distance. These groups are 1) trades for which the integral mode will be the operating method of choice for all L-D rates, 2) trades for which the drop-and-swap mode will be the operating method of choice for all L-D rates, and 3) trades for which the drop-and-swap mode will be optimum for low L-D rates and the integral mode will be optimum for high L-D rates.

The first group consists of trades with annual cargo flow requirements less than the ton-mile capacity of a single small (less than 25,000 DWT) tug-barge unit operating at an economical speed (6 to 9 knots). In these trades, since all the cargo can be easily transported in a single tug-barge unit, the higher tug utilization available from drop-and-swap operation is unnecessary. Thus, the integral mode will always be the operating method of choice in these trades.

The second group consists of trades with annual cargo flow requirements much greater than the ton mile capacity of two or more tug-barge units of maximum carrying capacity. The constraint on barge size prevents full use of economies of scale in the integral mode of operation. This, in turn, prevents operation in the integral mode at economical speeds with sufficiently less capital equipment to make up for the increased tug utilization savings inherent in the drop-and-swap mode of operation. Thus, the drop-and-swap mode will always be the operational mode of choice in deadweight constrained trades.

The third group consists of trades not falling within the first two groups. That is, trades for which annual cargo flows are too large to be carried in one tug-barge unit but too small to be carried in several maximum sized units. It is for these trades that there will be a L-D tradeoff point below which the drop-and-swap mode and above which the integral mode will be the operational method of choice.

In general, for trades with greater annual cargo flows and longer port separation distances, this tradeoff point is at greater L-D rates. This is because these trades demand more ton-mile transport capacity which can be met by either increasing the number or size of the tug-barge units. If the number of units is increased, then the additional barges that must remain in port in the drop-and-swap operations become less significant. If the barge size is increased, the increase for the drop-and-swap mode operation will be less than for integral mode operations due to the higher tug utilization efficiency. In either case the drop-and-swap mode is favored in these trades more than the integral mode.

It should be mentioned that trades with very short port separation distances (i.e. 500 NM) have L-D tradeoff points at higher values than for longer distance trades. This is because the percentage of the total voyage time spent in port (about 40% at the L-D tradeoff point) is large for these trades, favoring the higher tug utilization efficiencies achieved by drop-and-swap mode of operation.

From the base case runs, certain conclusions can also be drawn concerning the form and speed of the optimum barge. It appears that since the Froude number is so low (less than 0.16) at the optimum tug-barge speeds of seven to eleven knots that residual resistance does not have much significance. Thus, optimum barge forms have the greatest draft and breadth possible to reduce the length; and, consequently, the barge capital cost. This saving outweighs any resistance penalty caused by the blunt barge forms. The only concession made for the effect of resistance is with respect to block coefficient. For tug-barge speeds greater than ten knots finer lines are required while blunter lines are suitable at slower speeds.

5.2 Sensitivity Runs

Sensitivity runs were made to see the effect on required freight rates and L-D rate tradeoff points of changes in some of the semi-fixed parameter values used in the base case runs. Specifically, changes to the value of the cargo, the shoreside storage costs, and the maximum barge draft were investigated.

Due to the cost of these runs, an exhaustive set, including a wide range of variation of a single parameter or combination of parameters, could not be made. However, the runs whose inputs are given in Table 5.3 should give an indication, although not conclusive, of the effects of changes to their values. In the three sections that follow, an analysis is made of the output from the sensitivity runs.

5.2.1 Sensitivity Run: Change in Cargo Value

A sensitivity run was made to see what effects a change in the cargo value from \$200 to \$0 per long ton would have on the 1,000,000 LT annual cargo flow base case. As seen in Figure 5.6 and Table 5.4, this reduction in cargo value reduces the required freight rate by less than \$.20 for trades with port separation distance of 500 NM to over \$1.00 for trades with a port separation distance of 5000 NM. As to be expected, the reduction is in proportion to the port separation distance, which is indicative of the sea time and the time value of the cargo. It was also noticed that in the base case run, that the optimum system speeds would sometimes be one, two, or even three knots faster than the sensitivity run. This confirms what is to be expected, that higher cargo values result in higher optimum system speeds. Thus, it is seen that unless the time value of cargo is included in any transportation system optimization model, the transport vehicles will be optimized at too slow a speed.

5.2.2 Sensitivity Runs: Changes to Shoreside Storage Costs

Two sensitivity runs were made to determine what effects the inclusion of shoreside storage capital construction costs of \$48 per long ton storage capacity (1) would have on the

(1) This value for storage capacity capital costs was obtained from a major oil company and is indicative of oil tankage construction costs for large tanks as of January 1979. Also, the model assumes that the amount of shoreside storage capacity

1,000,000 and 6,000,000 long ton annual cargo flow base cases. In addition, a run was made to determine the effect on the 1,000,000 long ton base case when a \$1.00 per long ton storage facility throughput charge was also included.

It would be expected that since shoreside storage costs are applicable to the integral mode only, the inclusion of such costs would favor the drop-and-swap mode of operation. This expected result is confirmed by the graphical and tabular output of these runs presented in Figures 5.7-5.9 and Table 5.4. From both cases it is seen that the effect of the capital cost of shoreside storage on the integral mode becomes more pronounced with longer port separation distances. This is because these port pair trades demand the larger barges and consequently, larger and more expensive storage tanks. (1) Also, the effect of storage is greater in trades with low L-D rates since these trades require more shoreside loading and discharging terminal facilities to handle the annual cargo flows.

The combined results of both of these effects is that the L-D tradeoff point is shifted to higher values, especially for trades with longer port separation distances and that the

for a given port pair trade is equal to the product of the number of terminal facilities in the port times the deadweight of the optimum barge operating in the integral mode for that port pair trade.

(1) For the 1,000,000 LT annual cargo flow trades, the present valued annual capital charges for storage amount to approximately \$0.0564 per 10,000 DWT tankage and LT of cargo flow.

advantage the drop-and-swap mode has for trades with low L-D rates becomes more pronounced.

The addition of the \$1.00 storage operating cost favors the drop-and-swap mode even more than the storage capital costs. This is because it essentially increases all the integral mode trades' rfr by \$1.00, equivalent to the annualized capital cost for over 175,000 DWT of tankage. As seen by comparing Figures 5.8 and 5.9, this results in a shift of the integral mode curves by \$1.00 so that the L-D tradeoff point moves to a much higher value. For the port pair trades with a separation distance of 500 NM, where the integral mode has rfr's only marginally lower than the drop-and-swap mode at high L-D rates, the inclusion of the operating storage costs increases the integral mode rfr so much that the drop-and-swap mode dominates for all L-D rates. Since the differences between the drop-and-swap and integral rfr curves is more pronounced with large port separation distance, the tradeoff point shift is not as great for these trades. However, it is apparent that if shoreside storage costs are included in the economic analysis, the trades for which the drop-and-swap mode will be optimal will be extended to those of significantly higher loading/discharging rates.

5.2.3 Sensitivity Runs: Inclusion of Port Draft Limit

A sensitivity run was made to investigate the effects that a draft limitation would have on the 1,000,000 LT annual

cargo flow base case. Additionally, another run was made to see if the same effect occurred to the 1,000,000 LT trades that also included a \$48 per long ton storage terminal capital cost. The graphical and tabular results of these runs are shown in Figures 5.10 and 5.11 and Table 5.4.

Comparing these figures with the unrestricted draft cases shown in Figures 5.3 and 5.7, it was observed that there was little effect on trades which were optimized with small barges, those of less than 450 feet or 25,000 DWT. However, the effect was very pronounced in those trades which in the unrestricted draft case would be optimized with large barges. In those trades, both the drop-and-swap and integral mode rfr's were increased by several dollars. This is because the draft restrictions forced the optimum feasible barges of a given deadweight capacity to have greater lengths and consequently, greater capital cost. Additionally, since barge length was constrained to be less than 750 feet, the maximum barge size was limited to less than 65,000 DWT. This forced those trades which were optimized in the unrestricted draft cases to obtain the necessary ton-mile capacity by either increasing the speed of the tugs or the number of tug-barge units in operation--both requiring the rfr to increase even further. The draft limitation affects both the drop-and-swap and integral mode trades. However, since the majority of unlimited draft port pair trades are optimized with larger barges in the integral than the drop-and-swap mode, the effect

is somewhat greater for the integral mode. Thus, the L-D trade-off point is shifted to somewhat higher values in the longer distance trades where larger barges are predominant. Nevertheless, the shift is not very great.

Generally, it is seen that port draft restrictions are very costly in trades which have large annual cargo flows and/or long port separation distances and thus would normally take advantage of the economies of scale inherent in large barges. These trades, however, are slightly more favorable to the drop-and-swap mode of operation.

5.3 Suggestions for Model Refinement and Future Research

The simplest refinement that can be made to the drop-and-swap model is to verify and calibrate the barge hull steel weight regression equations obtained from the barge design subprogram with hull weight data from single-skin tank-barges currently in operation. This requires only the acquisition of detailed design data (especially block coefficient, percentage high strength steel used, scantling length, and hull steel weight) for a series of barges of different size and form. However, since this data is of proprietary nature, it was not obtained.

Another major refinement could be achieved by using tank-barge resistance and propulsion data instead of the single-screw, full-bodied data that was used in the powering subprogram. The tank-ship data, although adequate for the

preliminary economic analysis accomplished in the drop-and-swap model, is probably not accurate enough to be used for preliminary design purposes. However, other than results of some towing tank model tests presented in Robinson (1976) there is no published literature on the hydrodynamic performance of large push-towed tug-barge forms, single or twin screw. Therefore, there is a definite need for a series of towing tank model tests of tug-barge forms similar to those currently in use. Suggested ranges of forms that should be investigated are length-breadth ratios from 4.0 to 7.0 and breadth-draft ratios from 2.0 to 4.0 and block coefficients from 0.80 to 0.90. Evaluation is particularly needed of the drag caused by current and proposed linkage designs and by draft differentials between the tug and barge.

Another refinement could be achieved by updating, calibrating, and validating the tug capital cost regression equations. This would require researching tug machinery and linkage costs, and possibly developing a preliminary tug design subprogram. However, the latter may not be necessary if sufficient tug and supply boat data can be obtained to develop reliable regression equations for tug hull and outfit weights and/or costs.

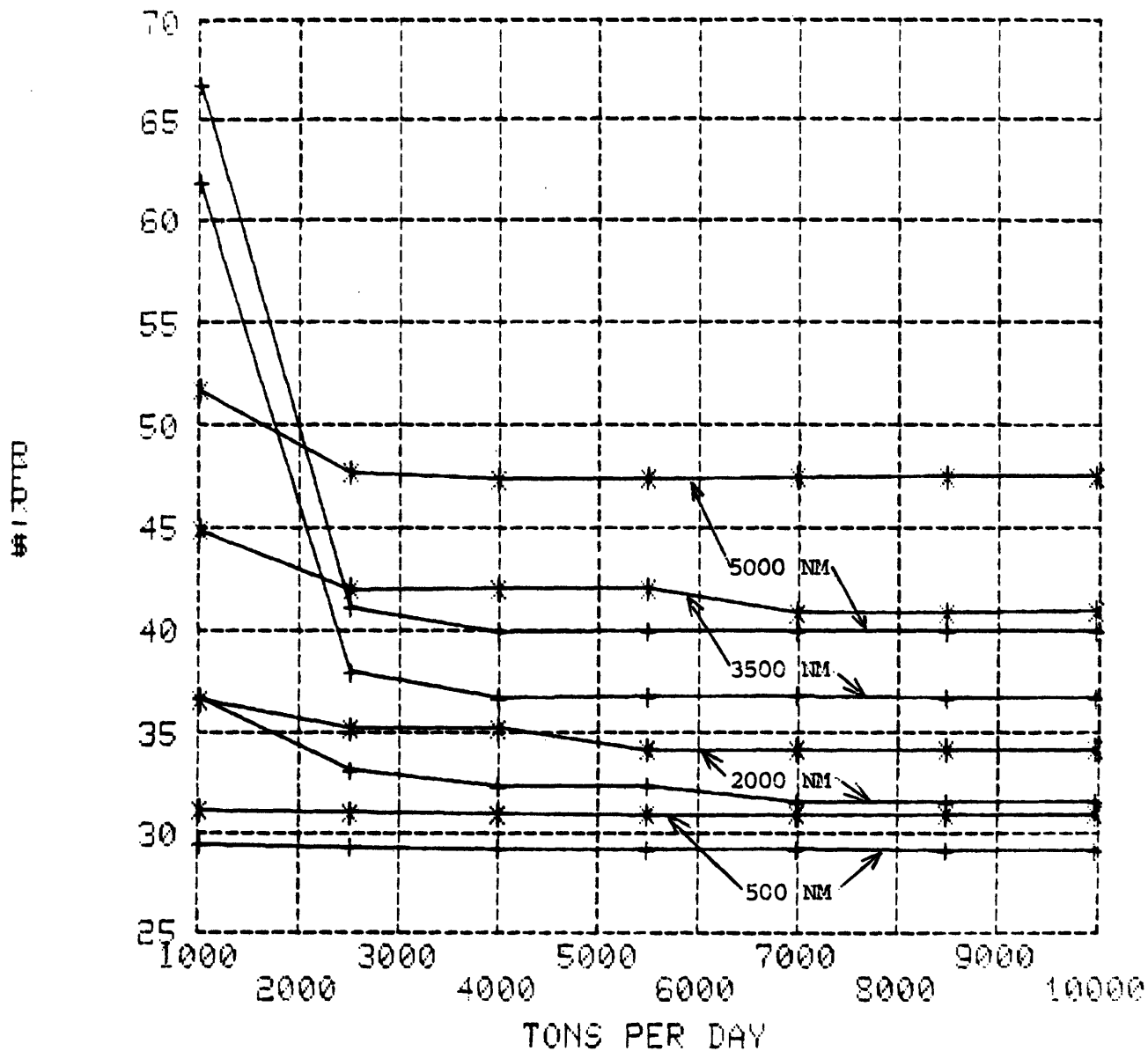
The model could also be improved by taking into account the stochastic nature of shipping operations. Tug-barge systems can not operate on precise time schedules on account of delays due to weather, equipment malfunction, strikes, port

congestion, etc. Since the drop-and-swap model is deterministic, and only uses average process times, it can not simulate the unpredictable nature of tug-barge operations. Nevertheless, for systems with a large number of tugs and barges, the stochastic nature will not be very important due to system averaging. Only in the small systems might these effects be of concern. Therefore, it would be useful to develop a simulation model that incorporates stochastic operations for a small system, say one tug and three barges. A comparison of the stochastic model results with that from the deterministic drop-and-swap model would then indicate how important is the stochastic nature of the system.

The stochastic nature of shipping schedules would also have an important effect on the amount of shoreside storage that would be needed for the integral mode of operation. However, given an estimate of the stochastic parameters of the system which could be obtained from a simulation model, the amount of shoreside storage can be determined from basic inventory theory.

Finally, the drop-and-swap model could be improved by extending it to multiport trades. Such an extension would complicate the problem significantly, especially if barges of different sizes can be used. Most likely network theory would need to be utilized to find the optimum tug-barge systems. However, since the stochastic nature of the problem would probably be very significant in multiport cases where

scheduling is of utmost importance, it may be that only simulation models could be used for multiport trades.

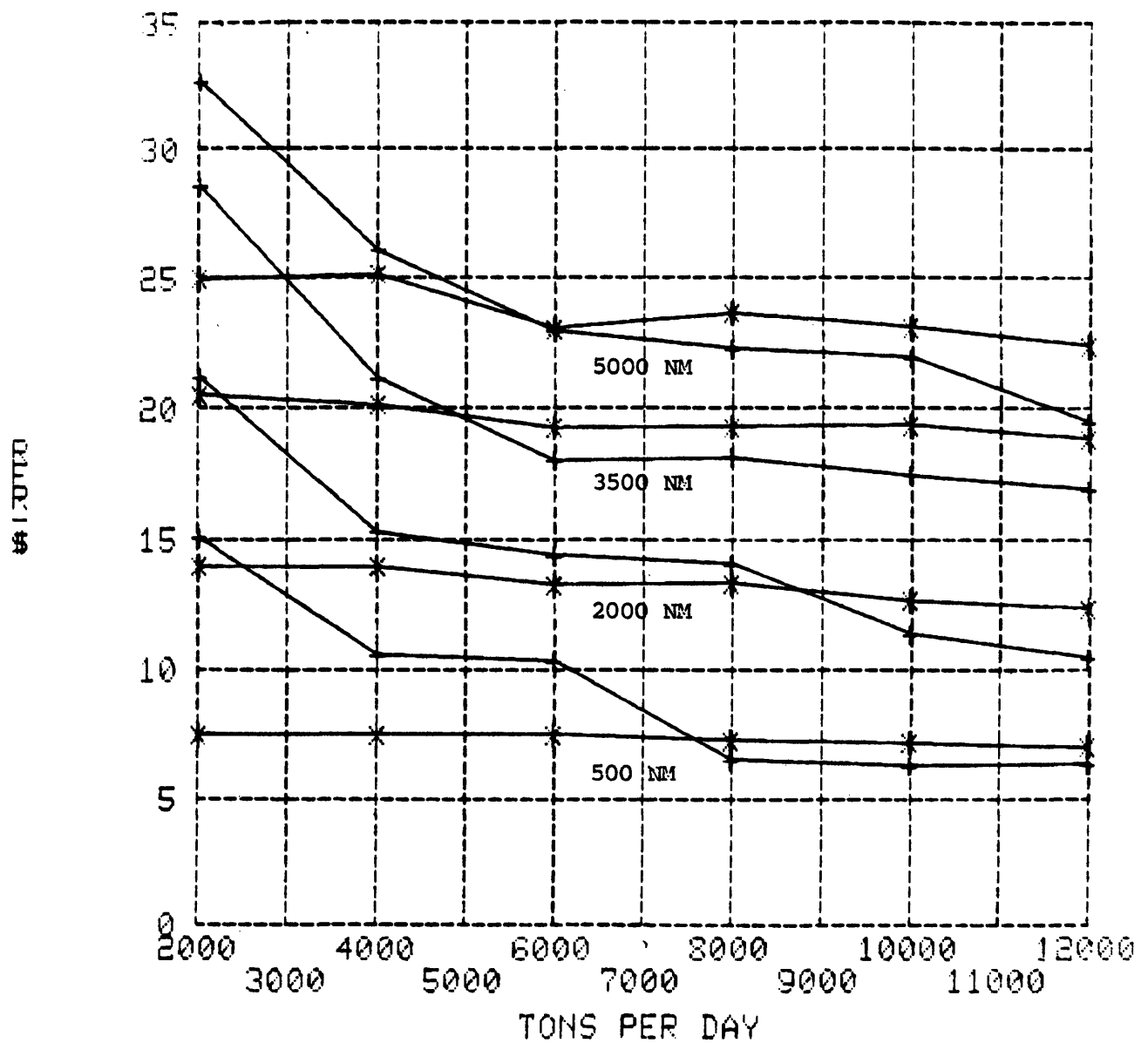


KEY: * -- DROP-AND-SWAP MODE
 + -- INTEGRAL MODE

FIGURE 5.1

PLOT OF RFR VS. TERMINAL L-D RATE

BASE CASE RUN: ANNUAL CARGO FLOW OF 100,000 LT

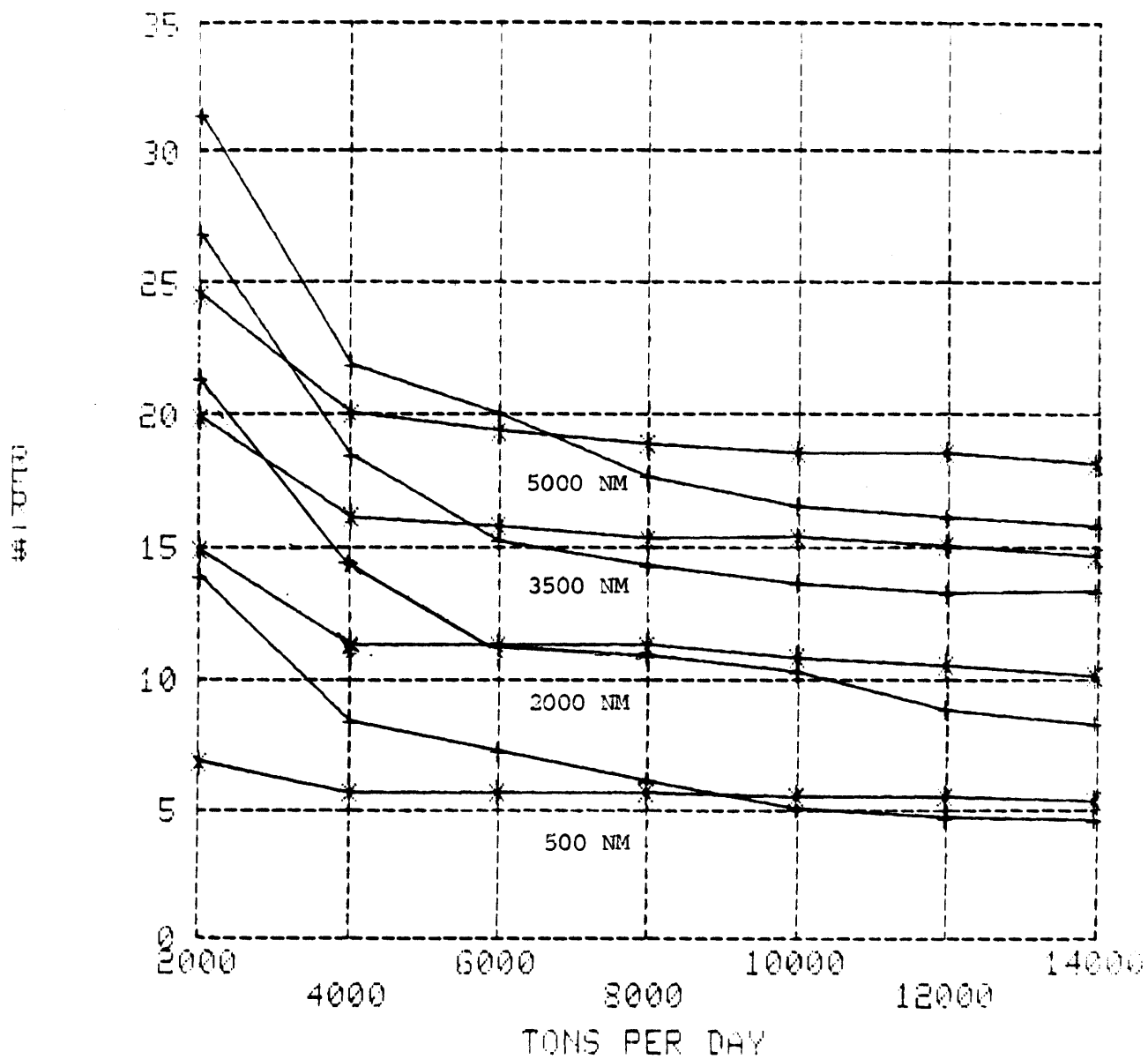


KEY: * -- DROP-AND-SWAP MODE
 + -- INTEGRAL MODE

FIGURE 5.2

PLOT OF RFR VS. TERMINAL L-D RATE

BASE CASE RUN: ANNUAL CARGO FLOW OF 600,000 LT



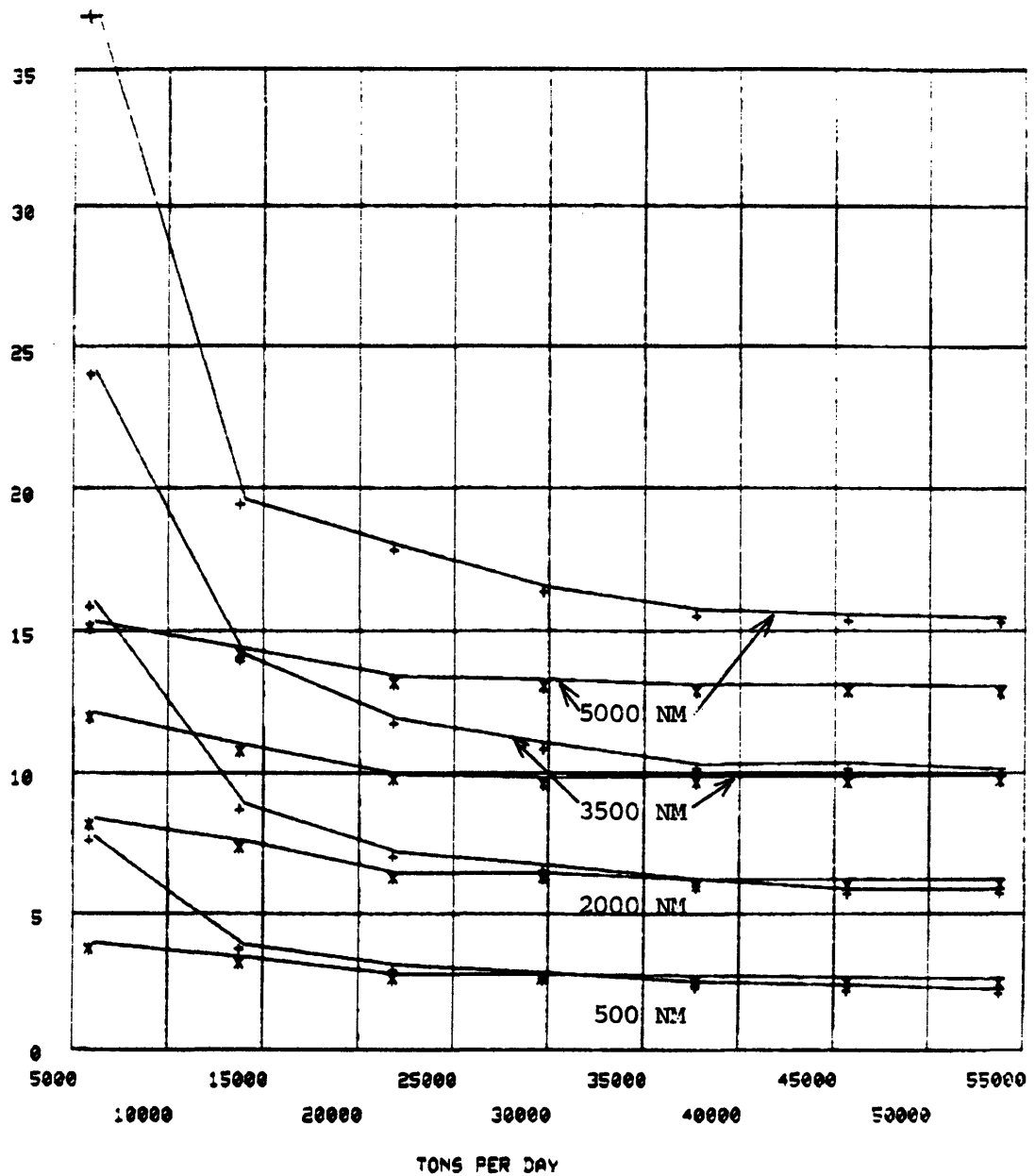
KEY: x -- DROP-AND-SWAP MODE
 + -- INTEGRAL MODE

FIGURE 5.3

PLOT OF RFR VS. TERMINAL L-D RATE

BASE CASE RUN: ANNUAL CARGO FLOW OF 1,000,000 LT

RFR



KEY: x -- DROP-AND-SWAP MODE
+ -- INTEGRAL MODE

FIGURE 5.4

PLOT OF RFR VS. TERMINAL L-D RATE

BASE CASE RUN: ANNUAL CARGO FLOW OF 6,000,000 LT

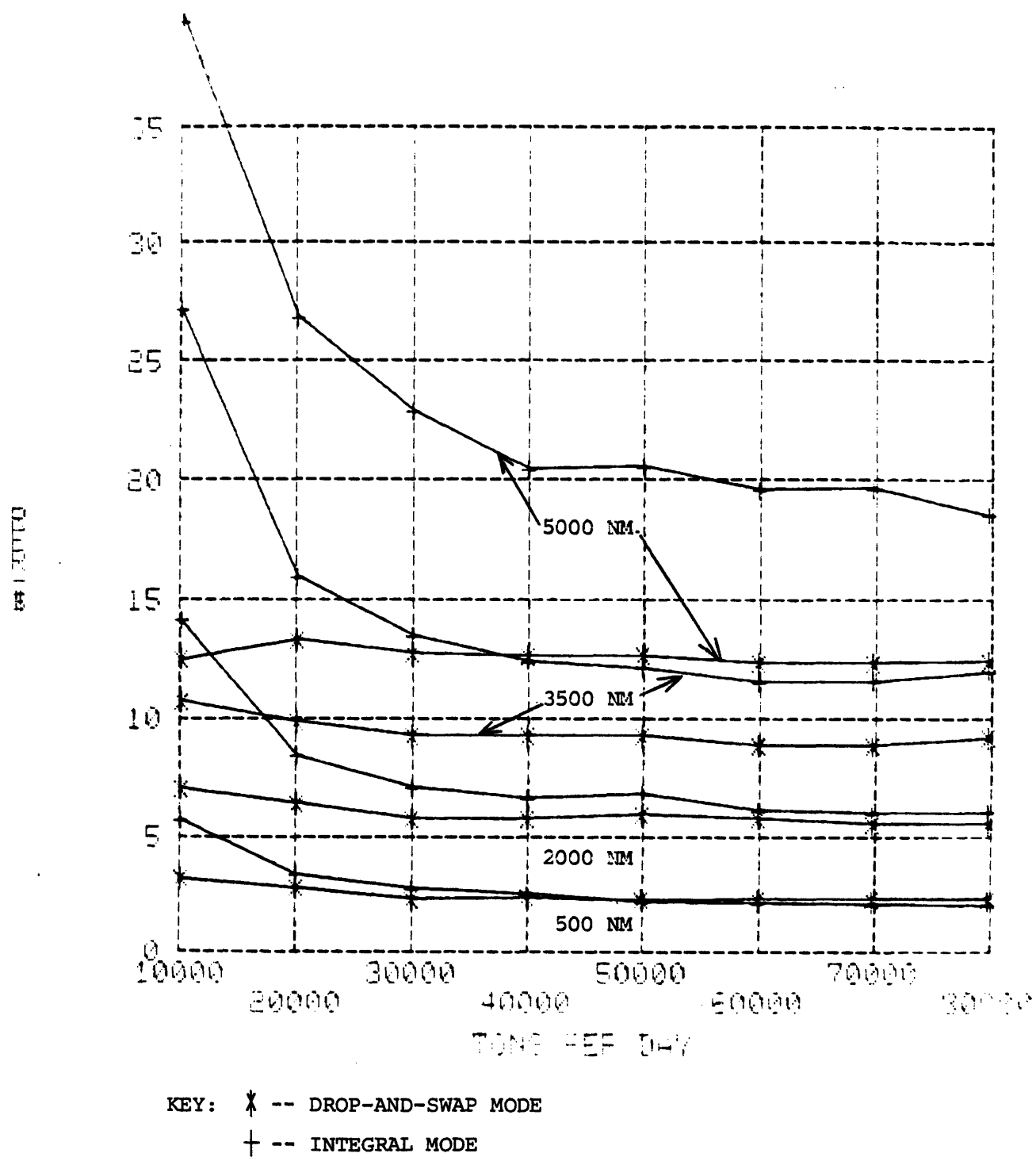
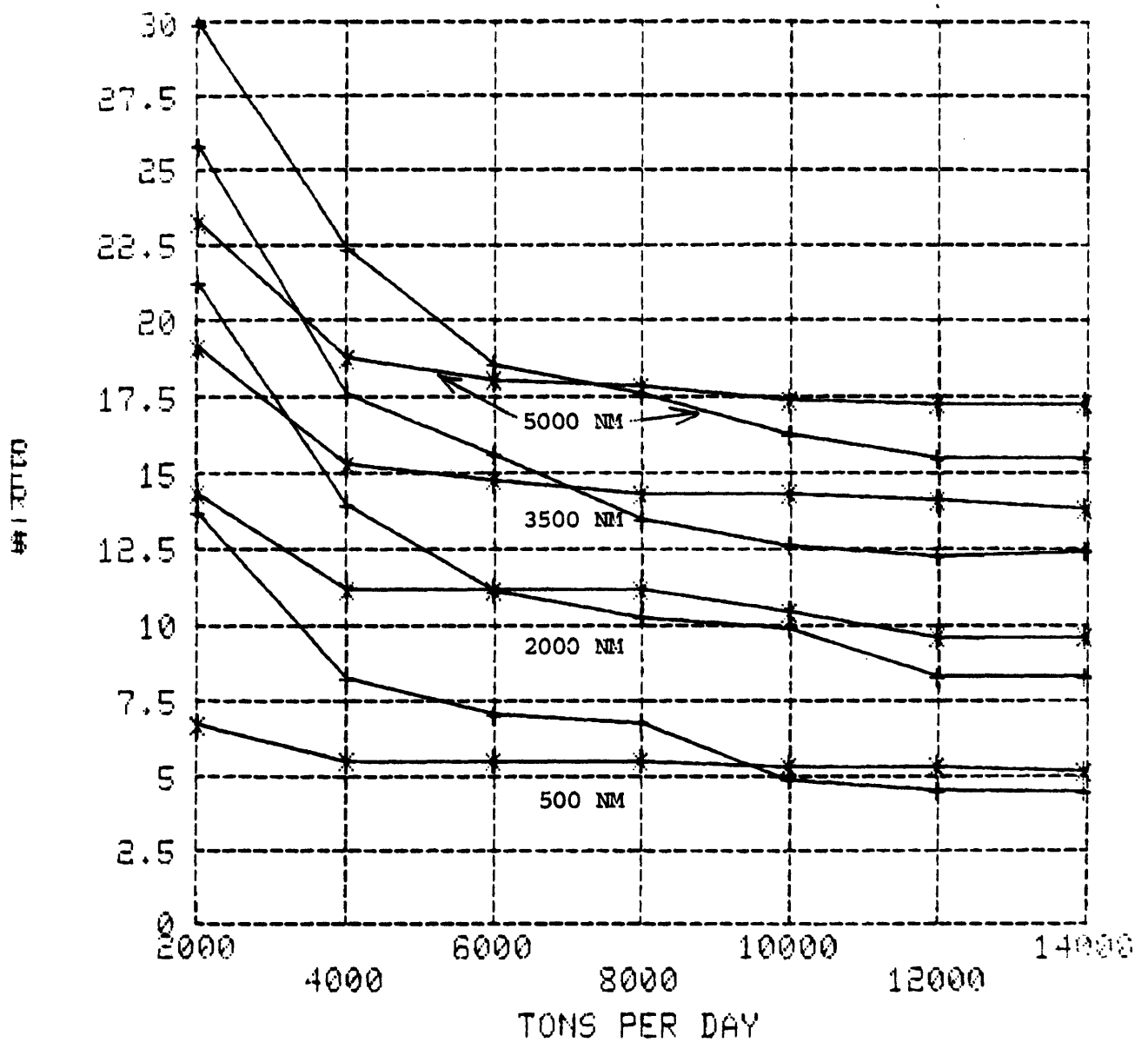


FIGURE 5.5

PLOT OF RFR VS. TERMINAL L-D RATE

BASE CASE RUN: ANNUAL CARGO FLOW OF 10,000,000 LT

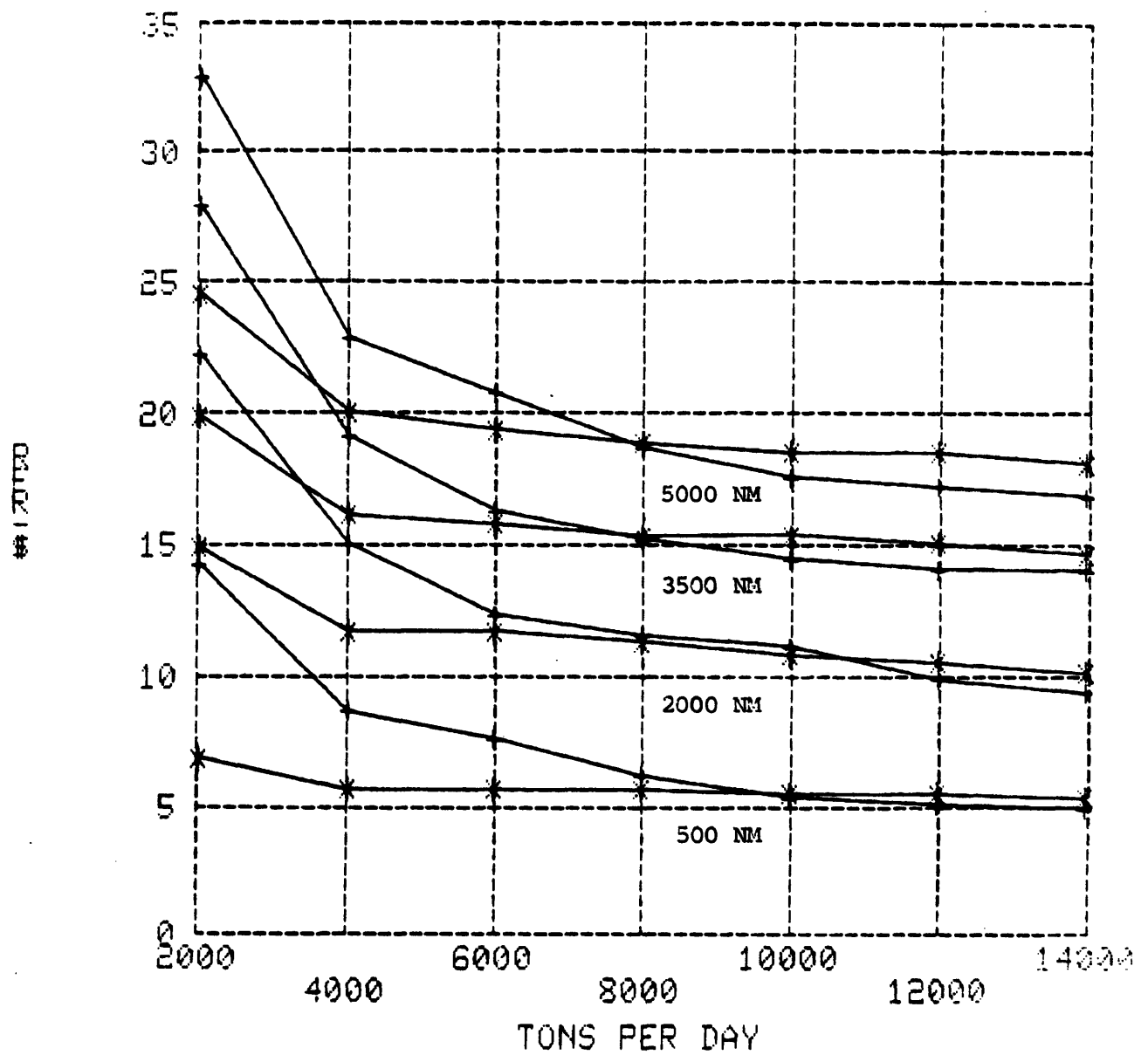


KEY: x -- DROP-AND-SWAP MODE
 + -- INTEGRAL MODE

FIGURE 5.6

PLOT OF RFR VS. TERMINAL L-D RATE

SENSITIVITY RUN: ANNUAL CARGO FLOW OF 1,000,000 LT
 WITH CARGO VALUE OF \$0

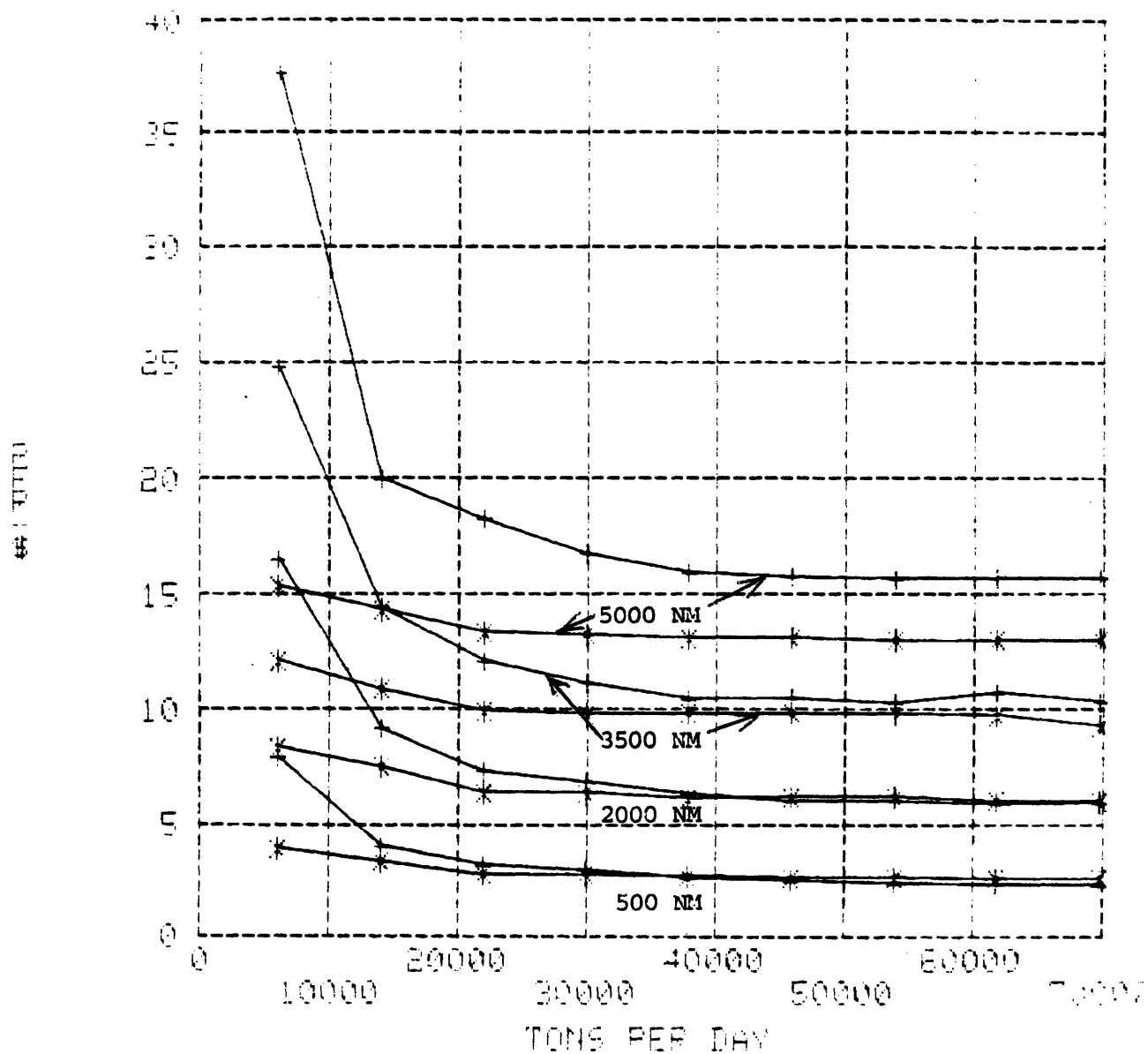


KEY: x -- DROP-AND-SWAP MODE
 + -- INTEGRAL MODE

FIGURE 5.7

PLOT OF RFR VS. TERMINAL L-D RATE

SENSITIVITY RUN: ANNUAL CARGO FLOW OF 1,000,000 LT
 WITH STORAGE FACILITY CAPITAL COST OF \$48/LT

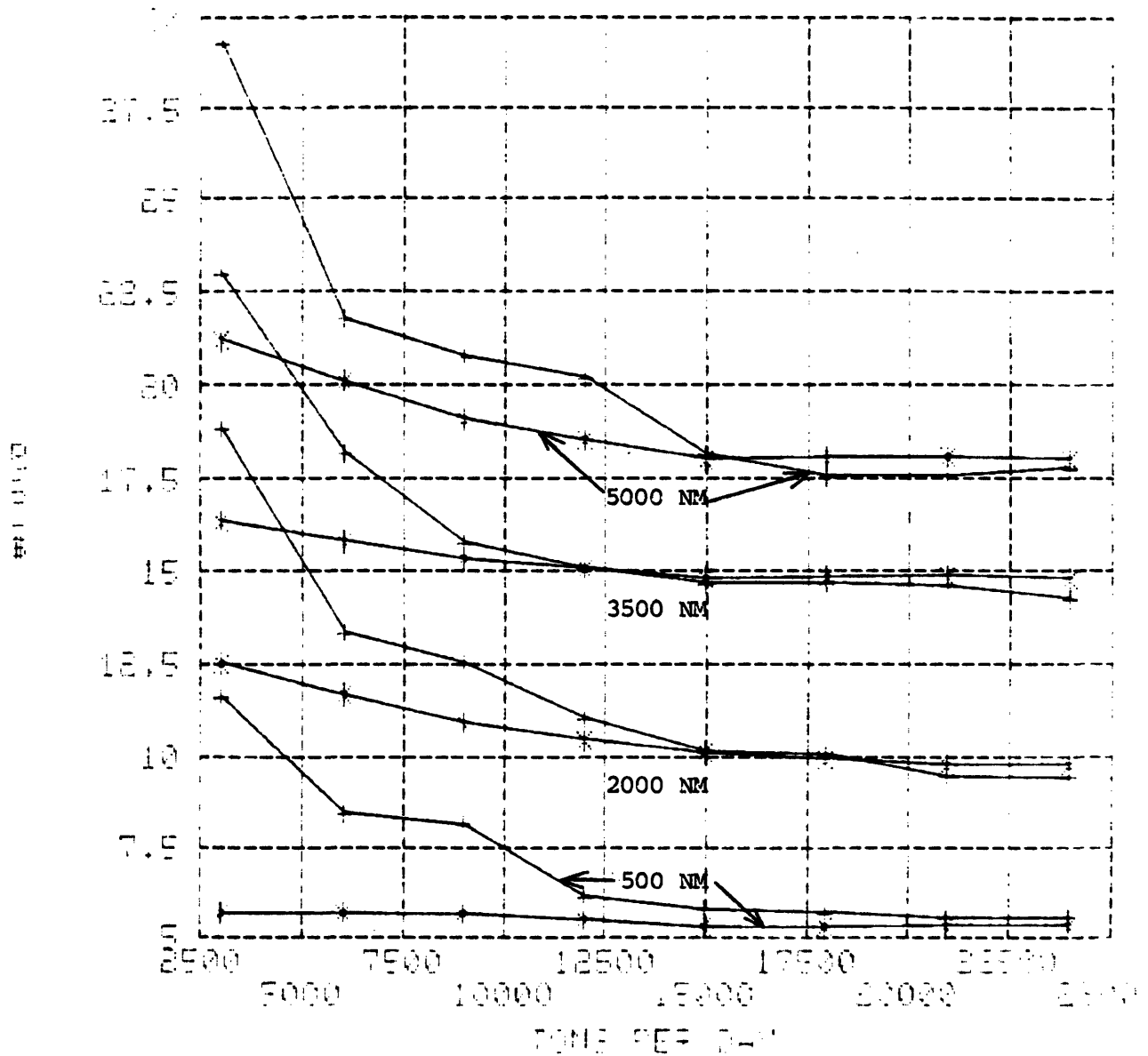


KEY: x -- DROP-AND-SWAP MODE
 + -- INTEGRAL MODE

FIGURE 5.8

PLOT OF RFR VS. TERMINAL L-D RATE

SENSITIVITY RUN: ANNUAL CARGO FLOW OF 6,000,000 LT
 WITH STORAGE FACILITY CAPITAL COST OF \$48/LT

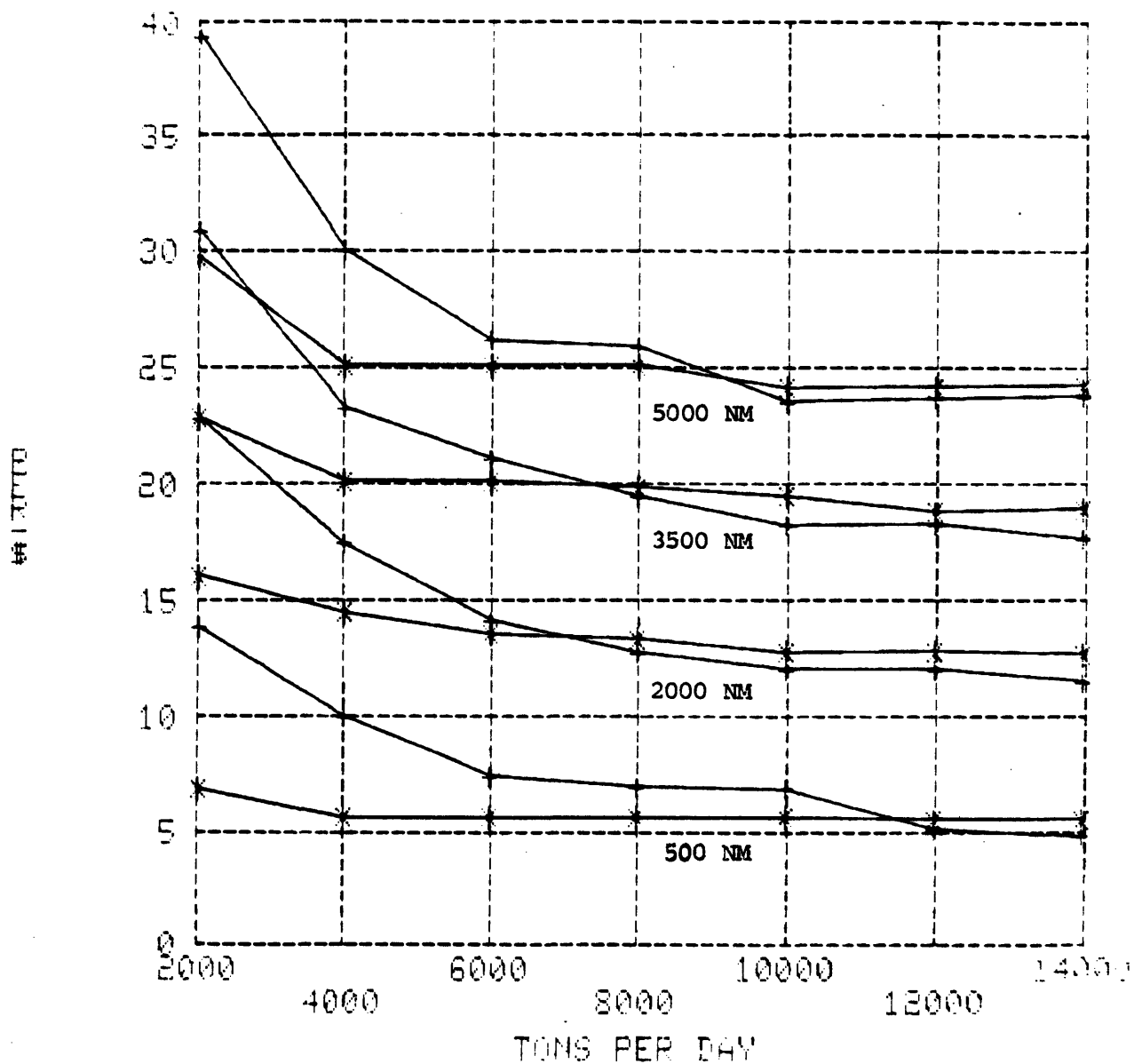


KEY: x -- DROP-AND-SWAP MODE
+ -- INTEGRAL MODE

FIGURE 5.9

PLOT OF RFR VS. TERMINAL L-D RATE

SENSITIVITY RUN: ANNUAL CARGO FLOW OF 1,000,000 LT
WITH STORAGE FACILITY CAPITAL COST OF \$48/LT
WITH STORAGE FACILITY OPERATING COST OF \$1/LT



KEY: x -- DROP-AND-SWAP MODE
 + -- INTEGRAL MODE

FIGURE 5.10

PLOT OF RFR VS. TERMINAL L-D RATE

SENSITIVITY RUN: ANNUAL CARGO FLOW OF 1,000,000 LT
 WITH BARGE DRAFT LIMITED TO 38'

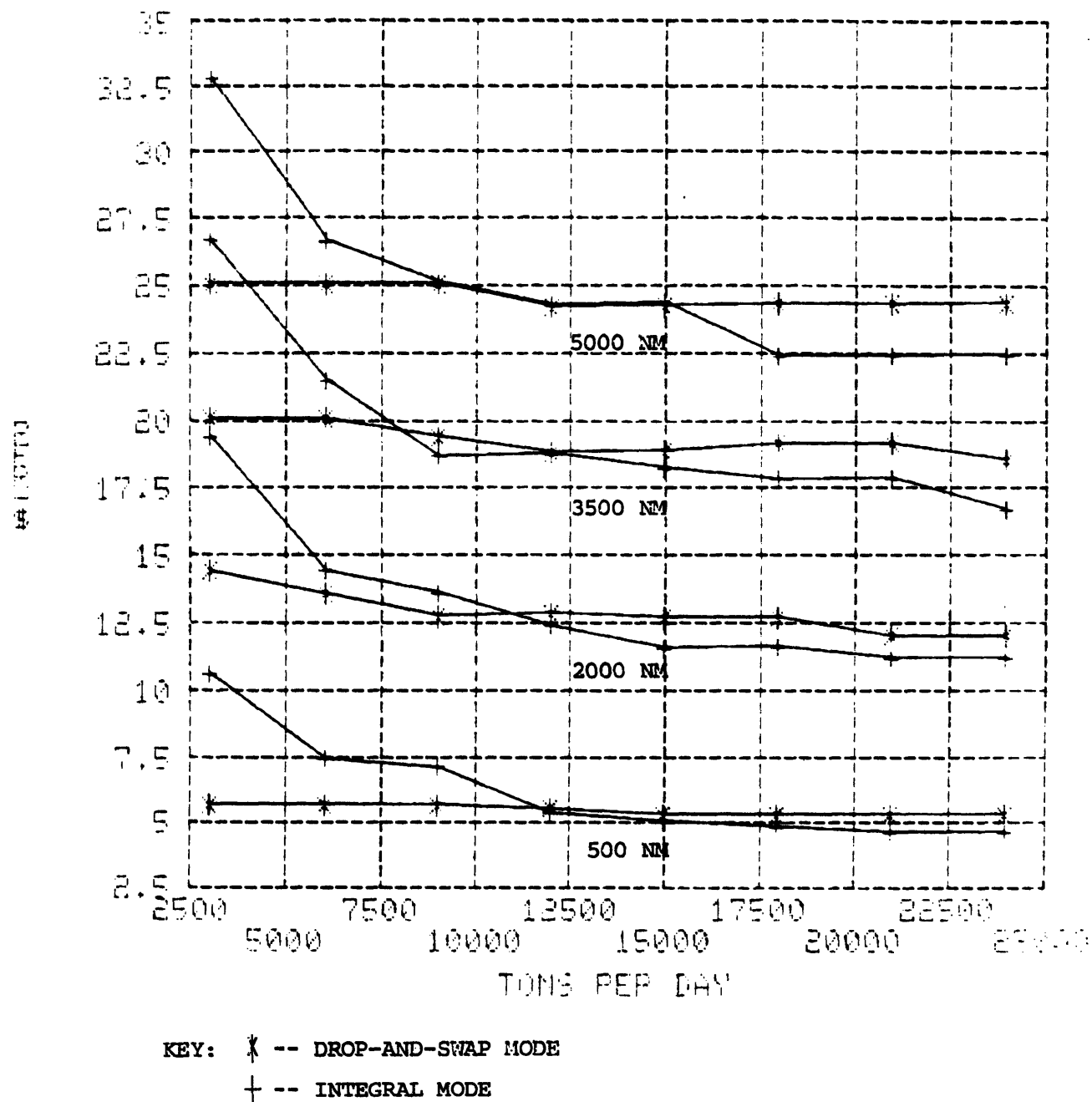


FIGURE 5.11

PLOT OF RFR VS. TERMINAL L-D RATE

SENSITIVITY RUN: ANNUAL CARGO FLOW OF 1,000,000 LT
 WITH BARGE DRAFT LIMITED TO 38'
 WITH STORAGE FACILITY CAPITAL COST OF \$48/LT

TABLE 5.1

SUMMARY OF INPUTS USED IN BASE CASE RUNS

aflowavel	minrate	mindist	mindwt	mincb	minlb	minbt	minspeed	Changes to Semifixed Parameters
aflowave2	maxrate	maxdist	maxdwt	maxcb	maxlb	maxbt	maxspeed	
	delrate	deldist	deldwt	delcb	dellb	delbt	delspeed	
100,000	1,000	500	5,000	0.77	6.0	2.0	6.0	None
0	10,000	5,000	50,000	0.84	6.6	3.0	12.0	
	1,500	1,500	2,500	0.01	0.2	0.5	1.0	
600,000	2,000	500	5,000	0.77	6.0	2.0	6.0	None
0	12,000	5,000	70,000	0.84	6.6	3.0	13.0	
	2,000	1,500	2,500	0.01	0.2	0.5	1.0	
1,000,000	2,000	500	5,000	0.77	6.0	2.0	6.0	None
0	14,000	5,000	100,000	0.84	6.6	3.0	13.0	
	2,000	1,500	5,000	0.01	0.2	0.5	1.0	
6,000,000	6,000	500	5,000	0.77	6.0	2.0	7.0	None
0	54,000	5,000	100,000	0.84	6.6	3.0	14.0	
	8,000	1,500	5,000	0.01	0.2	0.5	1.0	
10,000,000	10,000	500	30,000	0.77	6.0	2.0	8.0	None
0	80,000	5,000	100,000	0.84	6.6	3.0	16.0	
	10,000	1,500	5,000	0.01	0.2	0.5	1.0	

TABLE 5.2
TABULAR OUTPUT SUMMARY FOR BASE CASE RUNS

Port Separation Distance (Nautical miles)	OUTPUT PARAMETER	a f l o w a v e l (LT)				
		100,000	600,000	1,000,000	6,000,000	10,000,000
500	D&S Mode Speed Range (Kts)	6	7-9	7-10	8-10	8-10
	Integral Mode Speed Range (Kts)	6	6-9	6-12	7-10	8-10
	D&S Mode DWT Range (1000 LT)	5	10-15	15-25	30-70	55-90
	Integral Mode DWT Range (1000 LT)	5	7.5-20	15-40	40-70	60-85
	D&S Integral Mode L-D Rate Tradeoff Point (LT/day)	none: Integral mode always favorable	Between 6000 & 8000	Between 8000 & 10000	Between 30,000 & 38000	Between 40,000 & 50,000
	%Porttime is of Total Voyage Time at L-D Rate Tradeoff Point	--	44-31	72-59	45-39	37-38
2000	D&S Mode Speed Range (Kts)	7-8	8-12	8-10	8-9	9-10
	Integral Mode Speed Range (Kts)	6-9	7-12	7-12	9-11	10-12
	D&S Mode DWT Range (1000 LT)	7.5	32.5-42.5	30-80	75-95	85-100
	Integral Mode DWT Range (1000 LT)	7.5 - 12.5	27.5-42.5	50-95	75-100	85-100
	D&S Integral Mode L-D Rate Tradeoff Point (LT/day)	near 1000	Between 8000 & 10,000	Near 6000	Near 38000	None: D&S Mode always favorable
	%Porttime is of Total Voyage Time at L-D Rate Tradeoff Point	57	35-22	49	23	--
3500	D&S Mode Speed Range (Kts)	10-13	8-9	8-11	9-10	9
	Integral Mode Speed Range (Kts)	6-9	8-12	8-9	10-12	12-13
	D&S Mode DWT Range (1000 LT)	10 - 12.5	32.5-37.5	40-100	95-100	90-100
	Integral Mode DWT Range (1000 LT)	10 - 12.5	27.5-37.5	50-100	95-100	90-100
	D&S Integral Mode L-D Rate Tradeoff Point (LT/day)	Between 1000 & 2500	Between 4000 & 6000	Between 4000 & 6000	None: D&S Mode always favorable	None: D&S Mode always favorable
	%Porttime is of Total Voyage Time at L-D Rate Tradeoff Point	29	43-29	48	--	--
5000	D&S Mode Speed Range (Kts)	8-10	8-10	8-10	9-10	9-12
	Integral Mode Speed Range (Kts)	6-7	8-11	8-10	10-13	13-15
	D&S Mode DWT Range (1000 LT)	12.5-17.5	37.5-50	45-90	90-100	60-100
	Integral Mode DWT Range (1000 LT)	12.5-22.5	37.5-52.5	70-1000	90-100	60-100
	D&S Integral Mode L-D Rate Tradeoff Point (LT/day)	Between 1000 & 2500	Near 6000	Between 6000 & 8000	None: D&S Mode always favorable	None: D&S Mode always favorable
	%Porttime is of Total Voyage Time at L-D Rate Tradeoff Point	29	29	36-32	--	--

TABLE 5.3
SUMMARY OF INPUTS USED IN SENSITIVITY CASE-RUNS

aflowave1	minrate	mindist	mindwt	mincb	minlb	minbt	minspeed	Changes to
aflowave2	maxrate	maxdist	maxdwt	maxcb	maxlb	maxbt	maxspeed	Semixed
	delrate	deldist	deldwt	delcb	dellb	delbt	delspeed	Parameter
1,000,000	2,000	500	5,000	0.78	6.0	2.0	6.0	vcargo = 0
0	14,000	5,000	100,000	0.82	6.0	2.0	13.0	
	2,000	1,500	5,000	0.01			1.0	
6,000,000	6,000	500	30,000	0.78	6.0	2.0	7.0	cfixstor = 48
0	70,000	5,000	100,000	0.81	6.0	2.0	15.0	
	8,000	1,500	5,000	0.01			1.0	
1,000,000	4,000	500	5,000	0.78	6.0	2.0	6.0	cfixstor = 48
	16,000	5,000	100,000	0.81	6.0	2.0	13.0	
	2,000	1,500	5,000	0.01			1.0	
1,000,000	3,000	500	5,000	0.78	6.0	2.0	6.0	cfixstor = 48 cvarstor = 1
0	24,000	5,000	100,000	0.81	6.0	2.0	13.0	
	3,000	1,500	5,000	0.01			1.0	
1,000,000	3,000	500	5,000	0.78	6.0	2.0	5.0	cfixstor = 48 maxt1 = maxt2 = 38
0	24,000	5,000	100,000	0.81	6.6	3.0	12.0	
	3,000	1,500	5,000	0.01	0.2	0.5	1.0	
1,000,000	2,000	500	5,000	0.75	6.0	2.0	6.0	maxt1 = maxt2 = 38
0	14,000	5,000	100,000	0.83	6.4	3.0	13.0	
	2,000	1,500	5,000	0.01	0.2	0.5	1.0	

TABLE 5.4
TABULAR SUMMARY FOR SENSITIVITY RUNS

Port Separation Distance (Nautical Miles)	O U T P U T P A R A M E T E R S	S E N S I T I V I T Y R U N P A R A M E T E R S			
		aflowavel=1,000,000 Base Case	aflowavel=1,000,000 vcargo = 0	aflowavel=6,000,000 Base Case	aflowavel=6,000,000 cfixstor=48
500	RFR at Specified L-D rate, D&S mode RFR at Specified L-D rate, Integral mode RFR at Specified L-D rate, D&S mode RFR at Specified L-D rate, Integral mode RFR at L-D Rate Tradeoff Point D&S/Integral mode L-D rate Tradeoff Point (LT/day) % Port Time is of Total Voyage Time at L-D rate Tradeoff Point	\$ 6.85 @2000 13.84 " 5.49 @12000 4.65 " About \$5.55 Between 8000 & 10000 72-59	\$ 6.71 @2000 13.69 " 5.32 @12000 4.47 " About \$5.35 Between 8000 & 10000 59-37	\$ 3.87 @6000 7.71 " 2.55 @54000 2.17 " About \$2.68 Between 30000 & 38000 45-39	\$ 3.87 @6000 7.94 " 2.55 @54000 2.30 " About \$2.65 Between 30000 & 38000 45-39
2000	RFR at Specified L-D rate, D&S mode RFR at Specified L-D rate, Integral mode RFR at Specified L-D rate, D&S mode RFR at Specified L-D rate, Integral mode RFR at L-D Rate Tradeoff Point D&S/Integral mode L-D rate Tradeoff Point (LT/day) % Port Time is of Total Voyage Time at L-D rate Tradeoff Point	\$14.85 @2000 21.48 " 10.46 @12000 8.79 " 11.67 Near 6,000 49	\$14.31 @2000 21.23 " 9.56 @12000 8.27 " 11.13 Near 6,000 49	\$ 8.35 @6000 16.03 " 6.17 @54000 5.84 " 6.14 Near 38,000 23	\$ 8.35 @6000 16.45 " 6.17 @54000 6.03 " 6.15 Between 38,000 & 46000 23-19
3500	RFR at Specified L-D rate, D&S mode RFR at Specified L-D rate, Integral mode RFR at Specified L-D rate, D&S mode RFR at Specified L-D rate, Integral mode RFR at L-D Rate Tradeoff Point D&S/Integral mode L-D rate Tradeoff Point (LT/day) % Port Time is of Total Voyage Time at L-D rate Tradeoff Point	\$19.91 @2000 26.82 " 15.03 @12000 13.25 " 15.85 Between 4000 & 6000 48	\$19.16 @2000 25.82 " 14.10 @12000 12.24 " about \$14.50 Between 6000 & 8000 48-36	\$12.08 @6000 24.28 " 9.85 @54000 10.08 " --- None: D&S mode always favorable ---	\$12.08 @6000 24.84 " 9.85 @54000 10.26 " --- None: D&S mode always favorable ---
5000	RFR at Specified L-D rate, D&S mode RFR at Specified L-D rate, Integral mode RFR at Specified L-D rate, D&S mode RFR at Specified L-D rate, Integral mode RFR at L-D Rate Tradeoff Point D&S/Integral mode L-D rate Tradeoff Point (LT/day) % Port Time is of Total Voyage Time at L-D rate Tradeoff Point	\$24.56 @2000 31.42 " 18.55 @12000 16.07 " About \$19.25 Between 6000 & 8000 36-32	\$23.26 @2000 29.99 " 17.31 @12000 15.50 " About \$17.95 Between 6000 & 8000 32- 24	\$15.31 @6000 37.28 " 12.97 @54000 15.43 " --- None: D&S mode always favorable ---	\$15.31 @6000 37.79 " 12.97 @54000 15.61 " --- None: D&S mode always favorable ---

TABLE 5.4... (Continued)

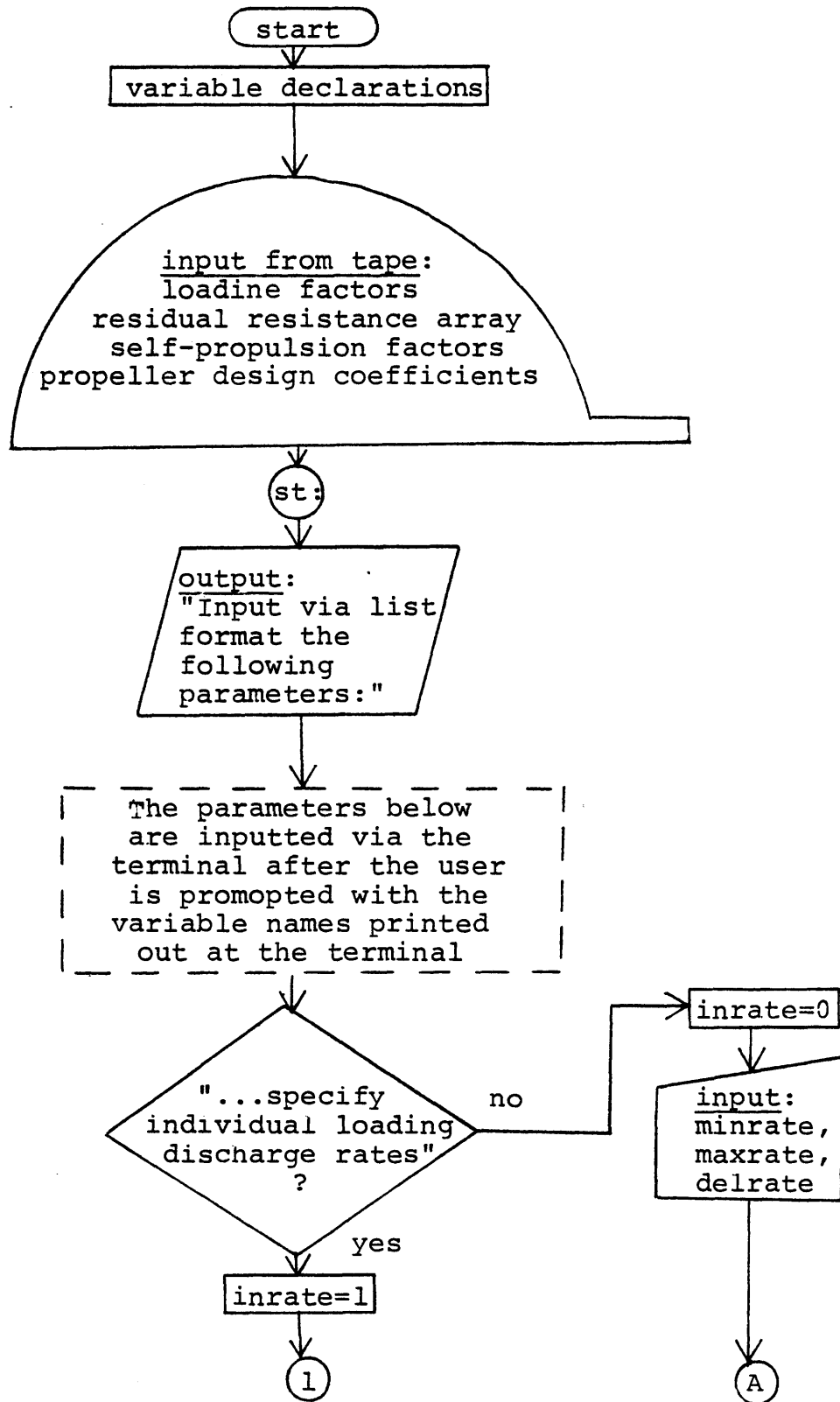
TABULAR SUMMARY FOR SENSITIVITY

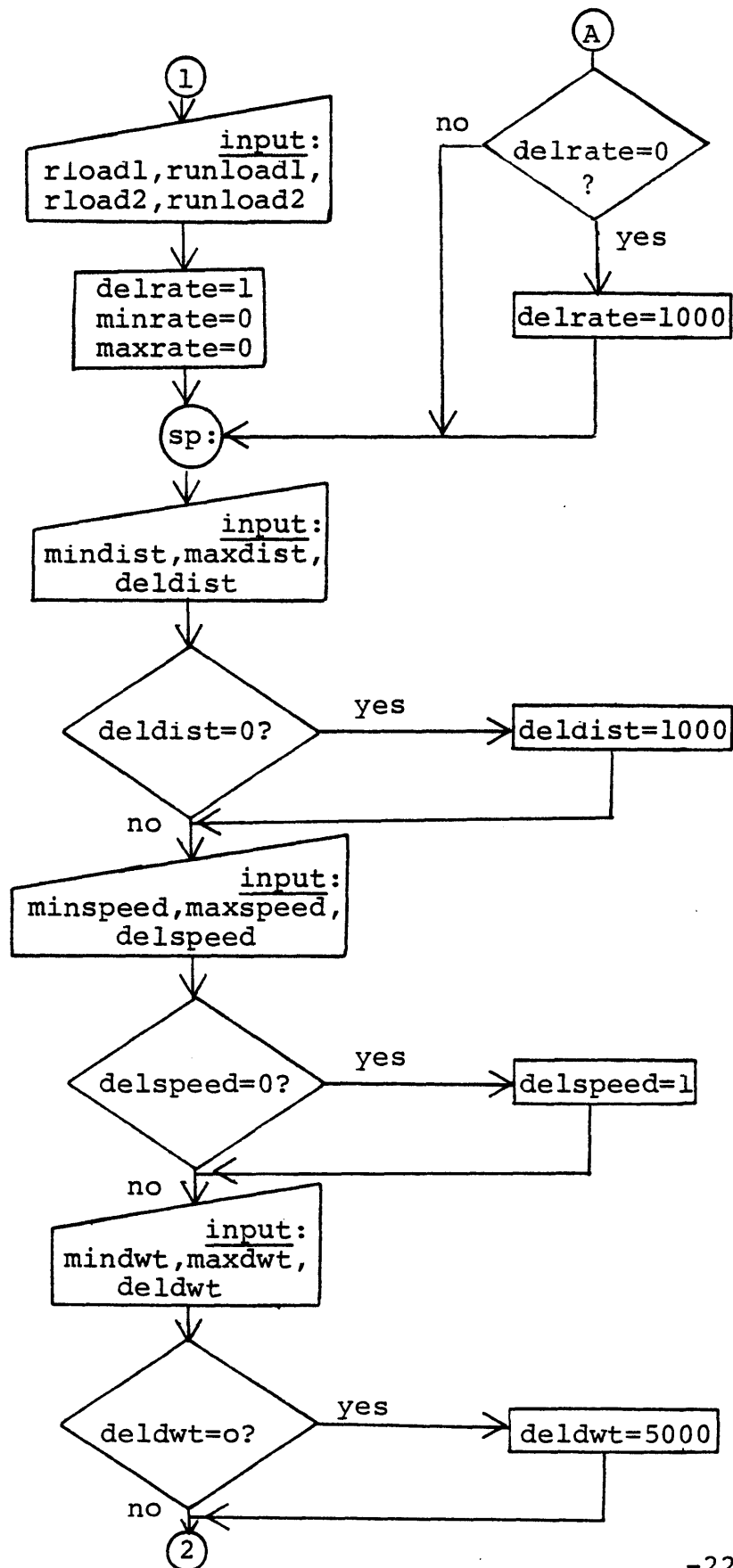
Port Separation Distance (Nautical Miles)	OUTPUT PARAMETERS	SENSITIVITY RUN PARAMETERS			
		aflowavel=1,000,000 cfixstor=48	aflowavel=1,000,000 cfixstor=48 cvastor = 1	aflowavel=1,000,000 cfixstor=48 maxtl=38	aflowavel=1,000,000 maxtl=38
500	RFR at Specified L-D rate, D&S mode	\$ 6.85 @2000	\$ 5.65 @3000	\$ 5.65 @3000	\$ 6.85 @2000
	RFR at Specified L-D rate, Integral mode	14.18 "	11.61 "	10.61 "	13.84 "
	RFR at Specified L-D rate, D&S mode	5.49 @12000	5.49 @12000	5.52 @12000	5.52 @12000
	RFR at Specified L-D rate, Integral mode	5.04 "	6.09 "	5.36 "	5.07 "
	RFR at L-D Rate Tradeoff Point	5.48	---	About \$5.50	About \$5.55
	D&S/Integral mode L-D rate Tradeoff Point (LT/day)	Near 10000	None: D&S Mode always favorable	Near 12000	Between 10000 & 12000
	% Port Time is of Total Voyage Time at L-D rate Tradeoff Point	57	---	48	48-29
2000	RFR at Specified L-D rate, D&S mode	\$14.85 @2000	\$12.50 @3000	\$14.43 @3000	\$16.05 @2000
	RFR at Specified L-D rate, Integral mode	22.31 "	18.85 "	19.47 "	22.84 "
	RFR at Specified L-D rate, D&S mode	10.46 @12000	10.46 @12000	12.84 @12000	12.84 @12000
	RFR at Specified L-D rate, Integral mode	9.87	11.05 "	12.39 "	12.05 "
	RFR at L-D Rate Tradeoff Point	About \$10.70	About \$10.00	About \$12.80	About \$13.45
	D&S/Integral mode L-D rate Tradeoff Point (LT/day)	Between 10000 & 12000	Between 18000 & 21000	Between 9000 & 12000	Between 6000 & 8000
	% Port Time is of Total Voyage Time at L-D rate Tradeoff Point	48-29	32-27	32-24	37-32
3500	RFR at Specified L-D rate, D&S mode	\$19.91 @2000	\$16.34 @3000	\$20.11 @3000	\$22.90 @2000
	RFR at Specified L-D rate, Integral mode	27.95 "	23.01 "	26.72 "	30.89 "
	RFR at Specified L-D rate, D&S mode	15.03 @12000	15.03 @12000	18.85 @12000	18.85 @12000
	RFR at Specified L-D rate, Integral mode	14.04 "	15.04 "	18.79 "	18.29 "
	RFR at L-D Rate Tradeoff Point	About \$5.30	About 15.04	About 19.00	About \$20,000
	D&S/Integral mode L-D rate Tradeoff Point (LT/day)	Near 8000	12000	Between 6000 & 12000	Between 6000 & 8000
	% Port Time is of Total Voyage Time at L-D rate Tradeoff Point	36	24	24-17	26-24
5000	RFR at Specified L-D rate, D&S mode	\$24.56 @2000	\$21.24 @3000	\$25.10 @3000	\$29.72 @2000
	RFR at Specified L-D rate, Integral mode	33.00 "	29.35 "	32.87 "	39.44 "
	RFR at Specified L-D rate, D&S mode	18.55 @12000	18.55 @12000	24.25 @12000	24.25 @12000
	RFR at Specified L-D rate, Integral mode	17.20 "	20.20 "	24.28 "	23.72 "
	RFR at L-D Rate Tradeoff Point	About \$19.00	About \$18.05	About \$24.25	About \$24.60
	D&S/Integral mode L-D rate Tradeoff Point (LT/day)	Near 8000	Near 15000	12000	Between 8000 & 10000
	% Port Time is of Total Voyage Time at L-D rate Tradeoff Point	36	19	17	25-19

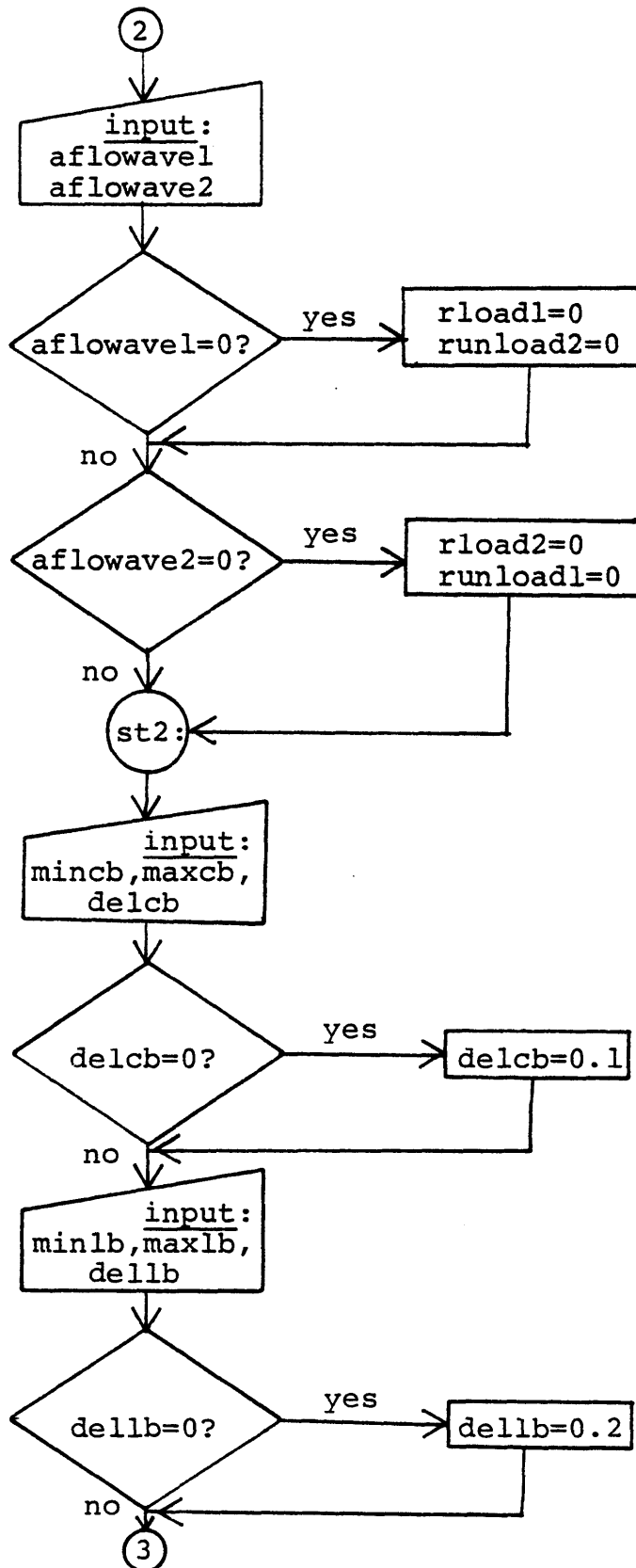
APPENDIX A

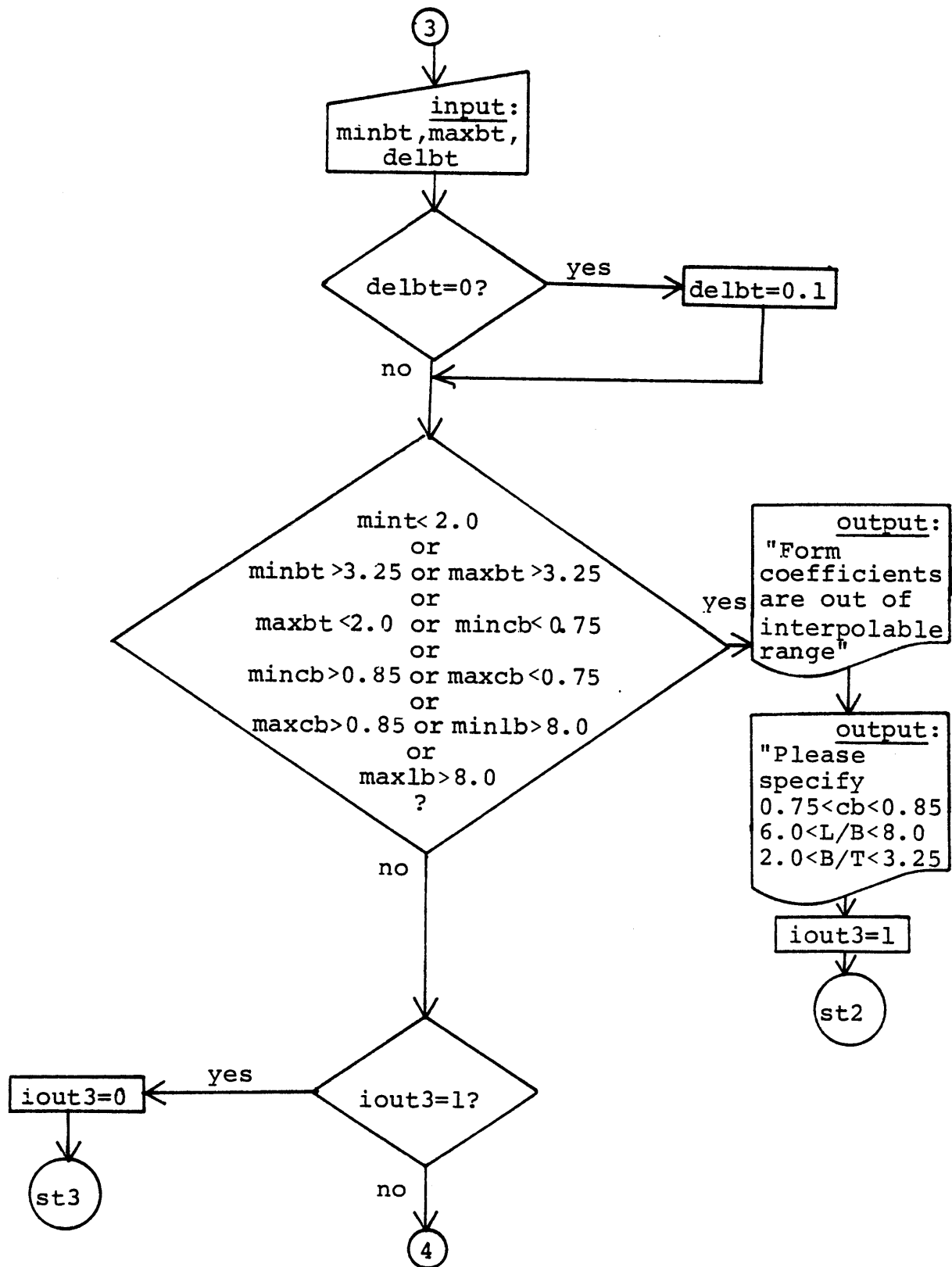
DOCUMENTATION FOR THE DROP-AND-SWAP COMPUTER MODEL

A.1 FLOWCHART OF THE DROP-AND-SWAP COMPUTER MODEL









4

The program now opens and reads the data in the "semifixedparams" file containing the present values of the following parameters: tugopdays, bargeopdays, tlink, tunlink, servmargin, fuelfmargin, sfc, cfuel, clube, nrcrew, cwages, csubs, csteelt, coutfitt, csteelb, coutfitb, ltug, wmisc, aother, admin, cfixport1, cfixport2, cvarport1, cvarport2, cfixterm, cvarterm, cfixstor, cvarstor, delay1, delay2, max1, maxb, maxt1, maxt2, disrate, econlife, inflafctr, vcargo

output:
"Input any changes to semifixed data via get data format"

Any changes to the semifixed variables are inputted by writing "variable = value," The last change is followed by a semicolon

"...want printed output?"

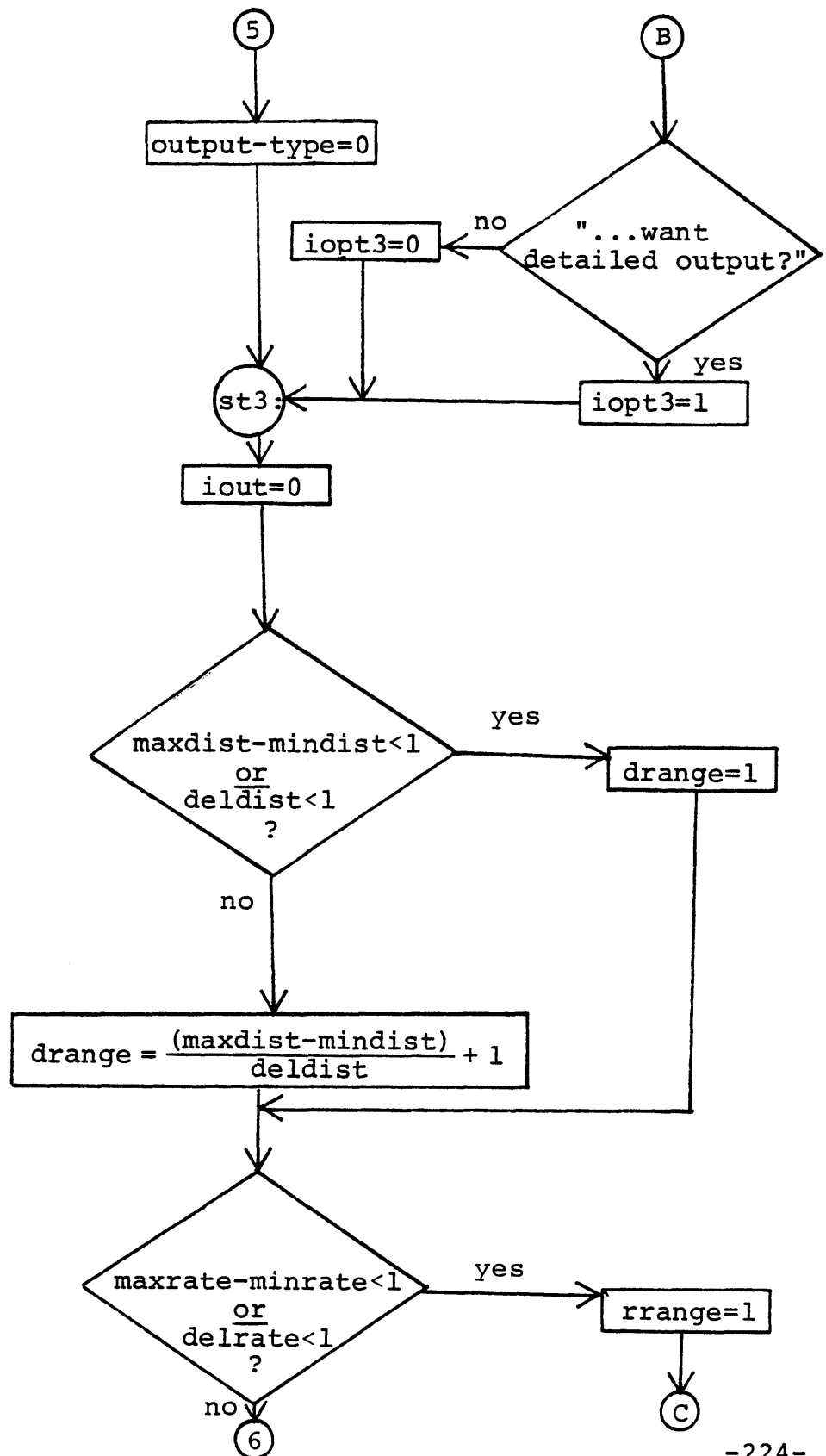
yes

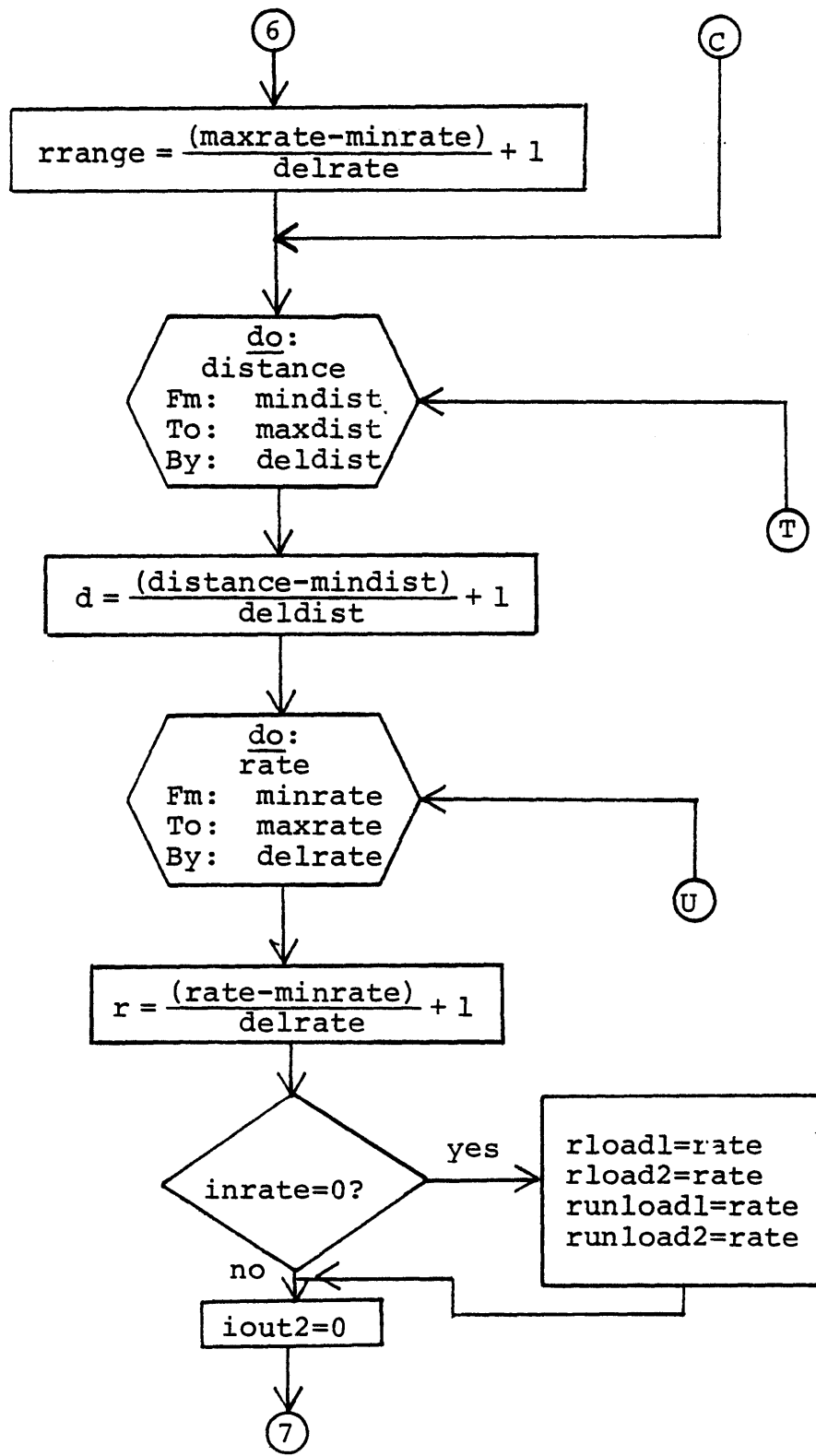
output-type=1

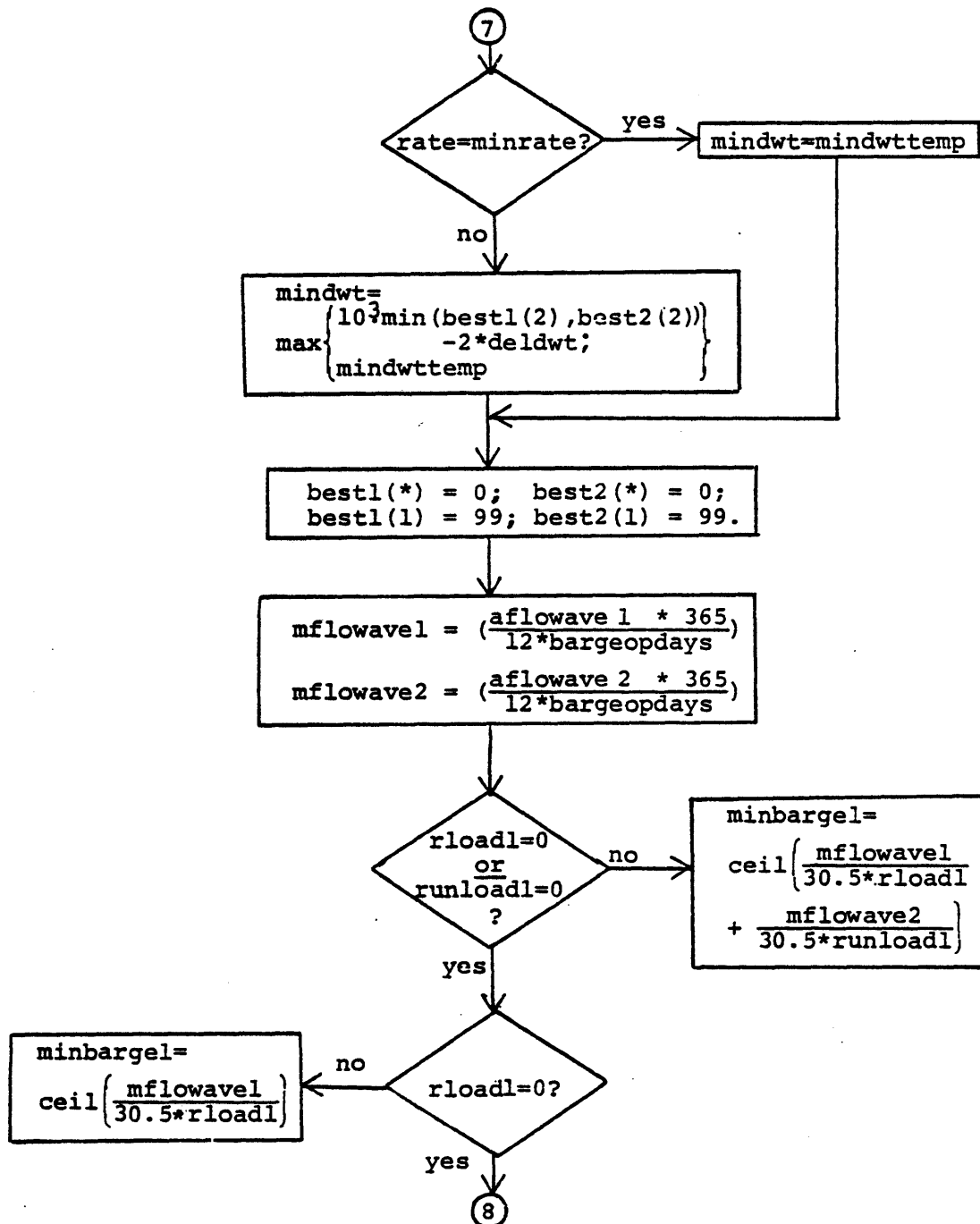
no

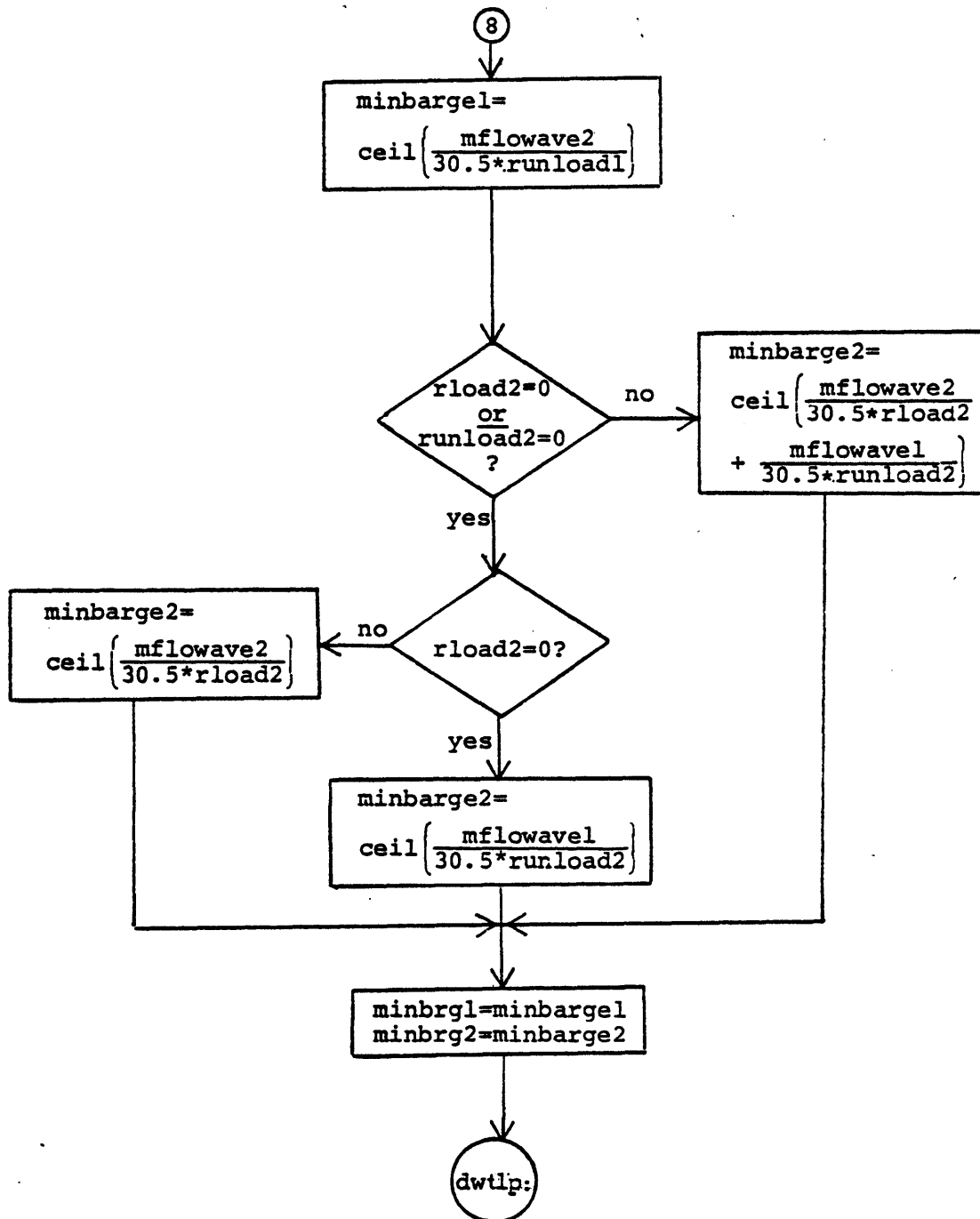
5

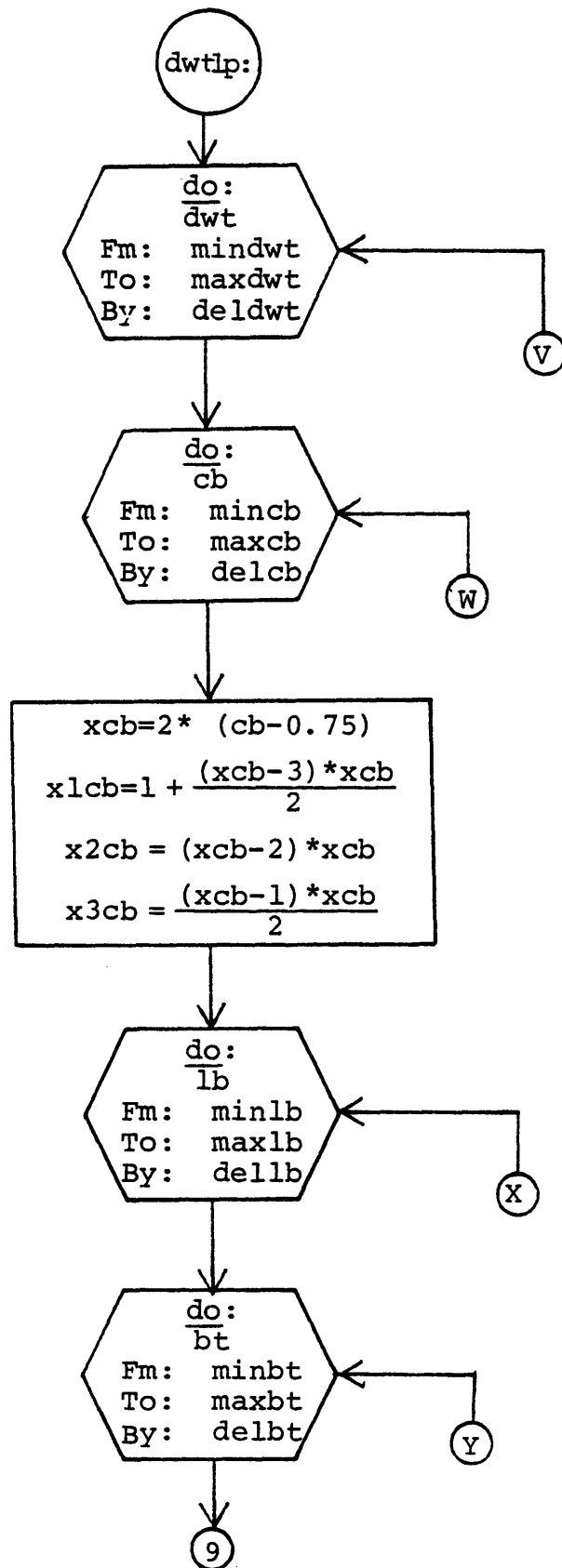
B

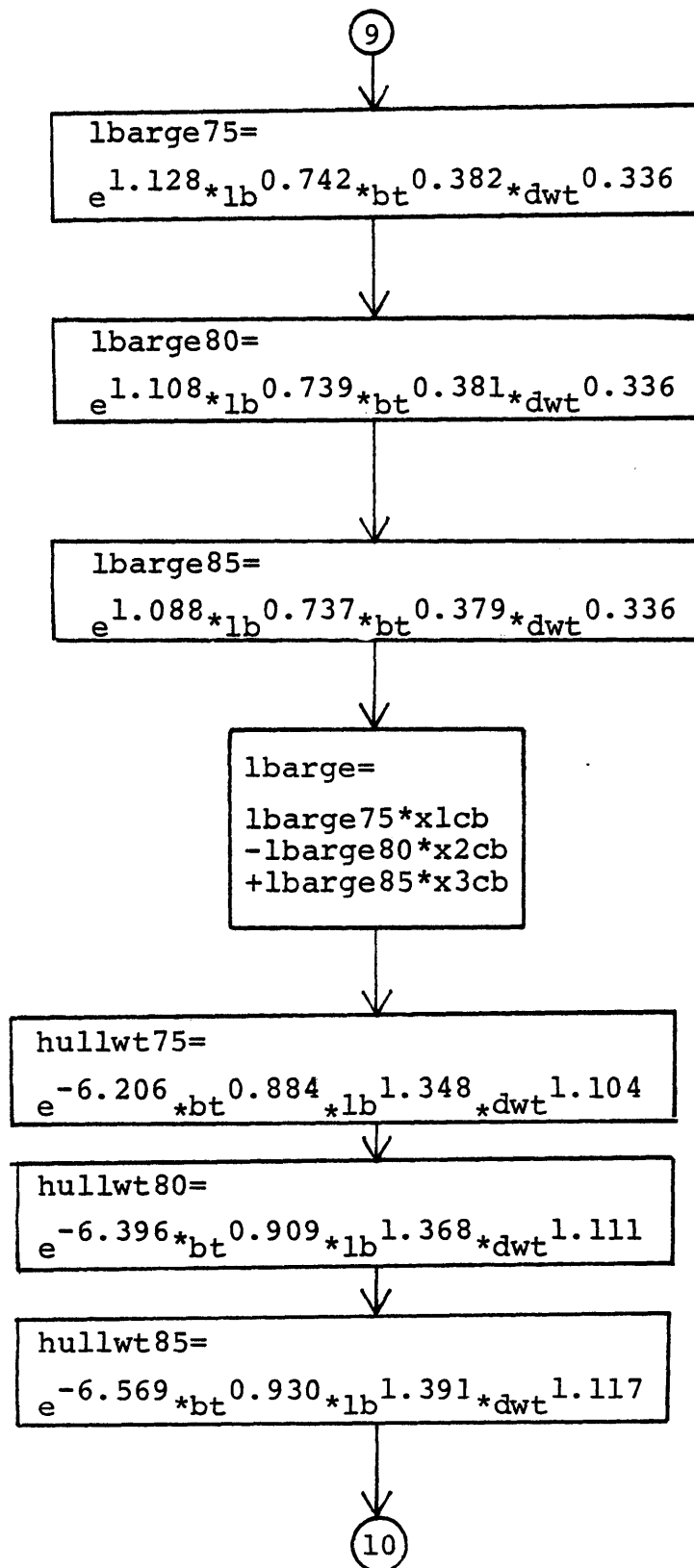


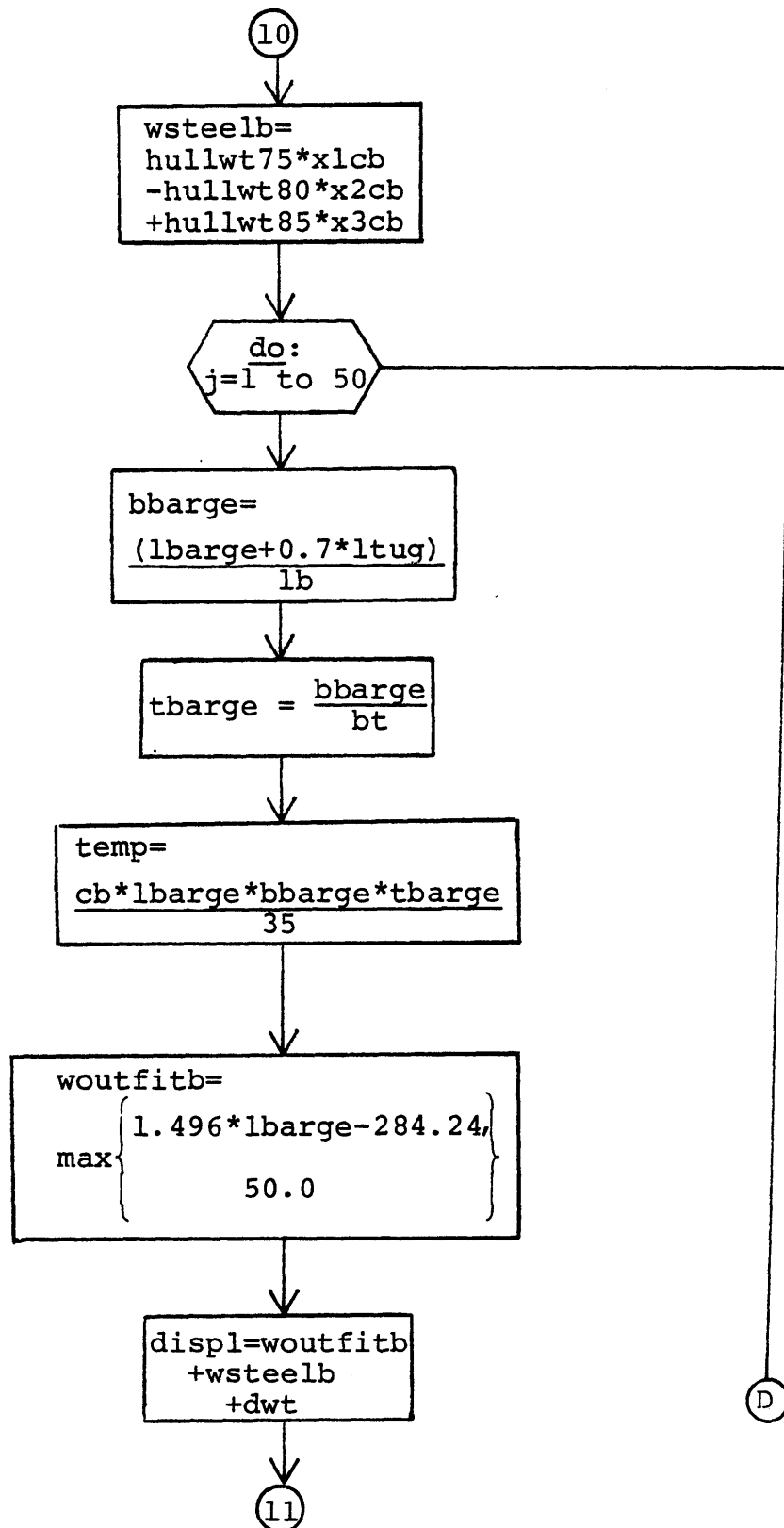


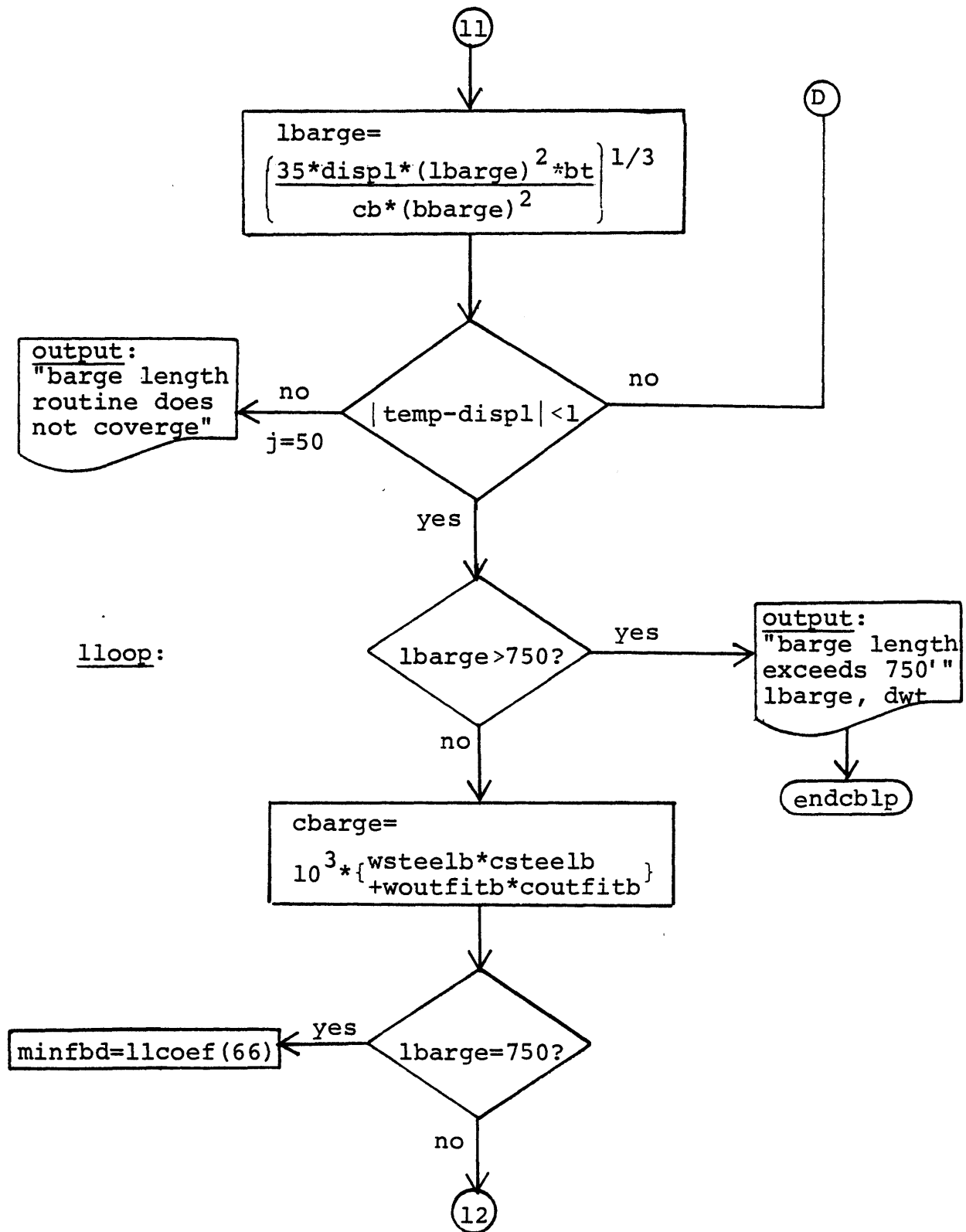


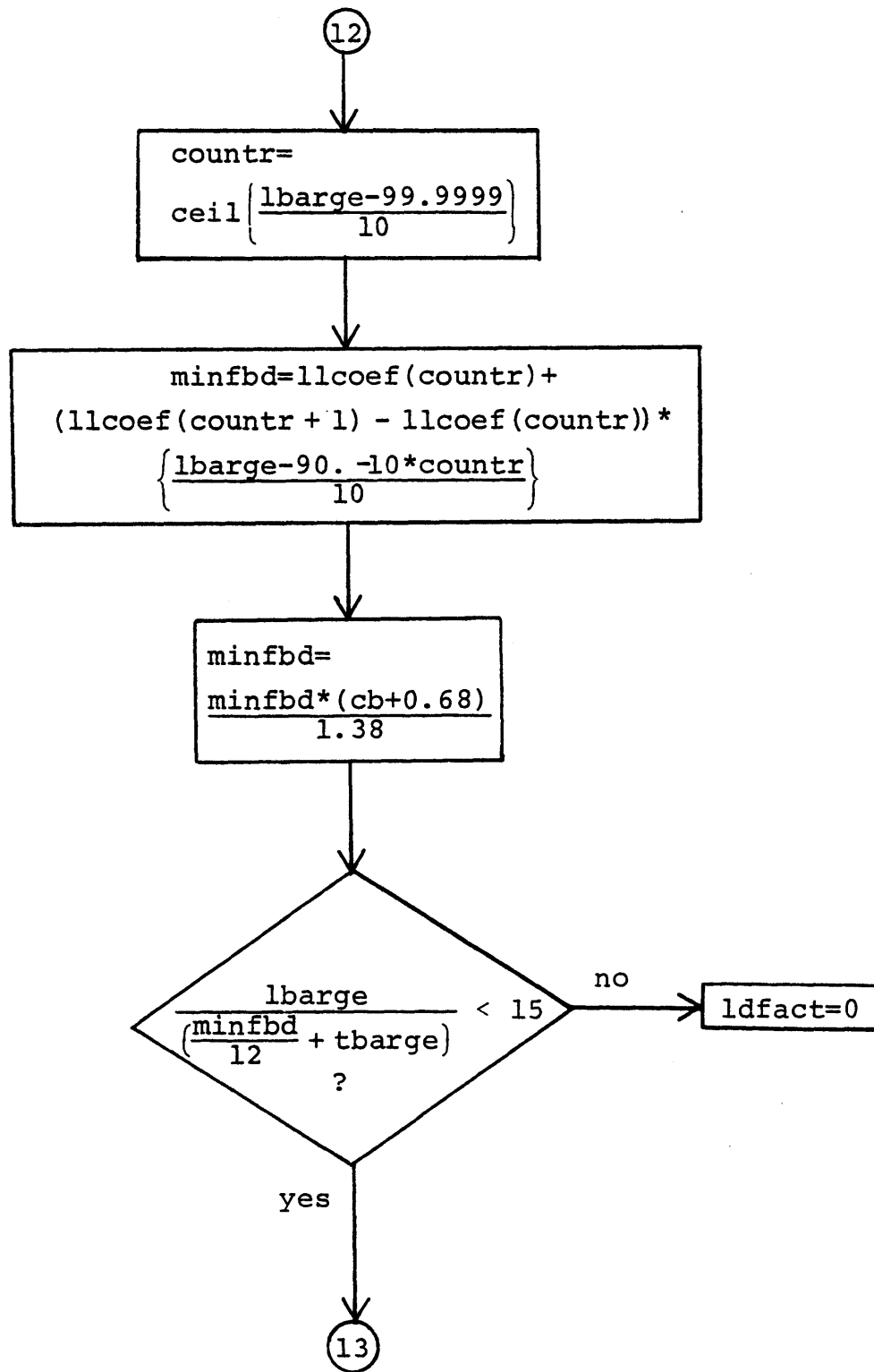


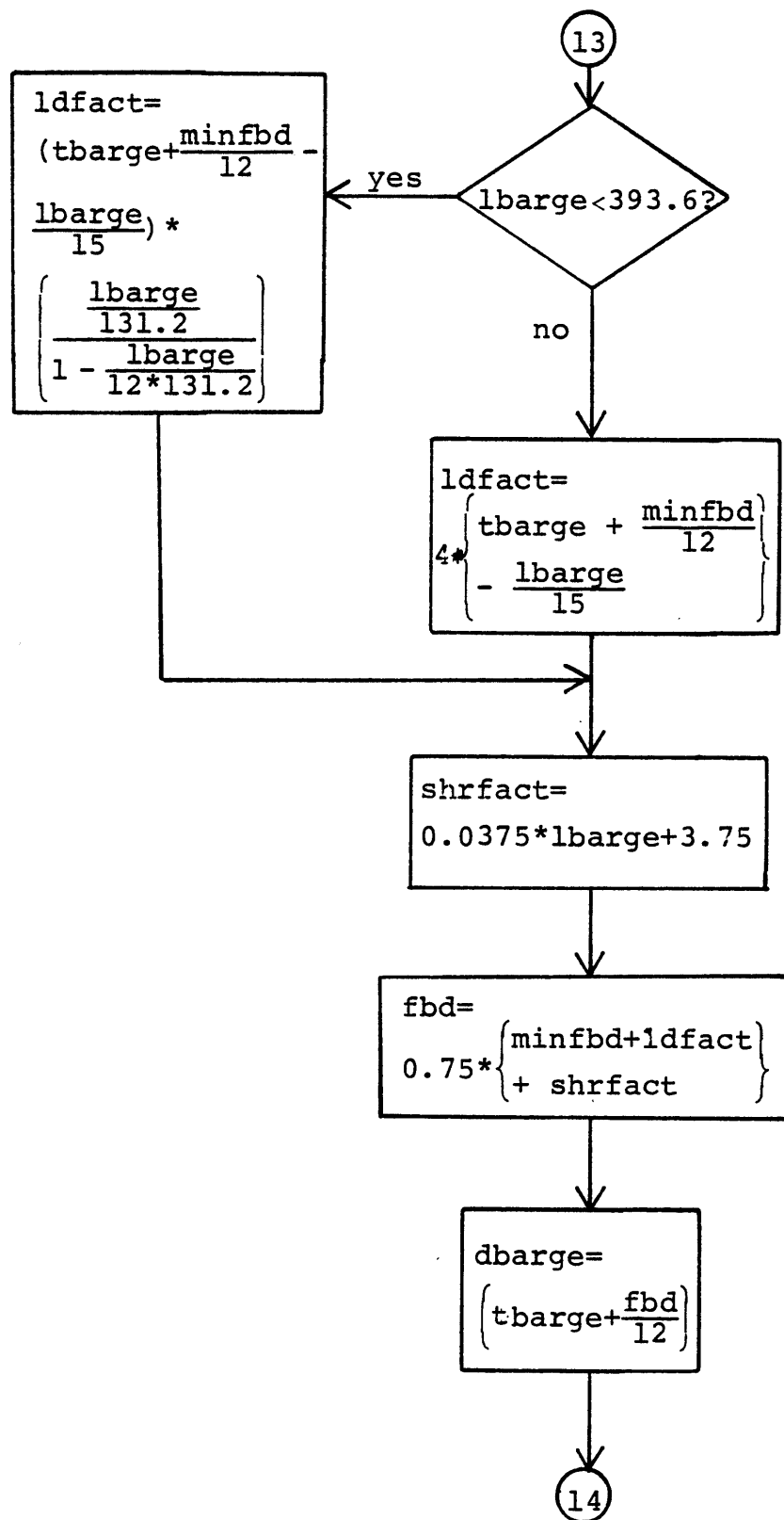


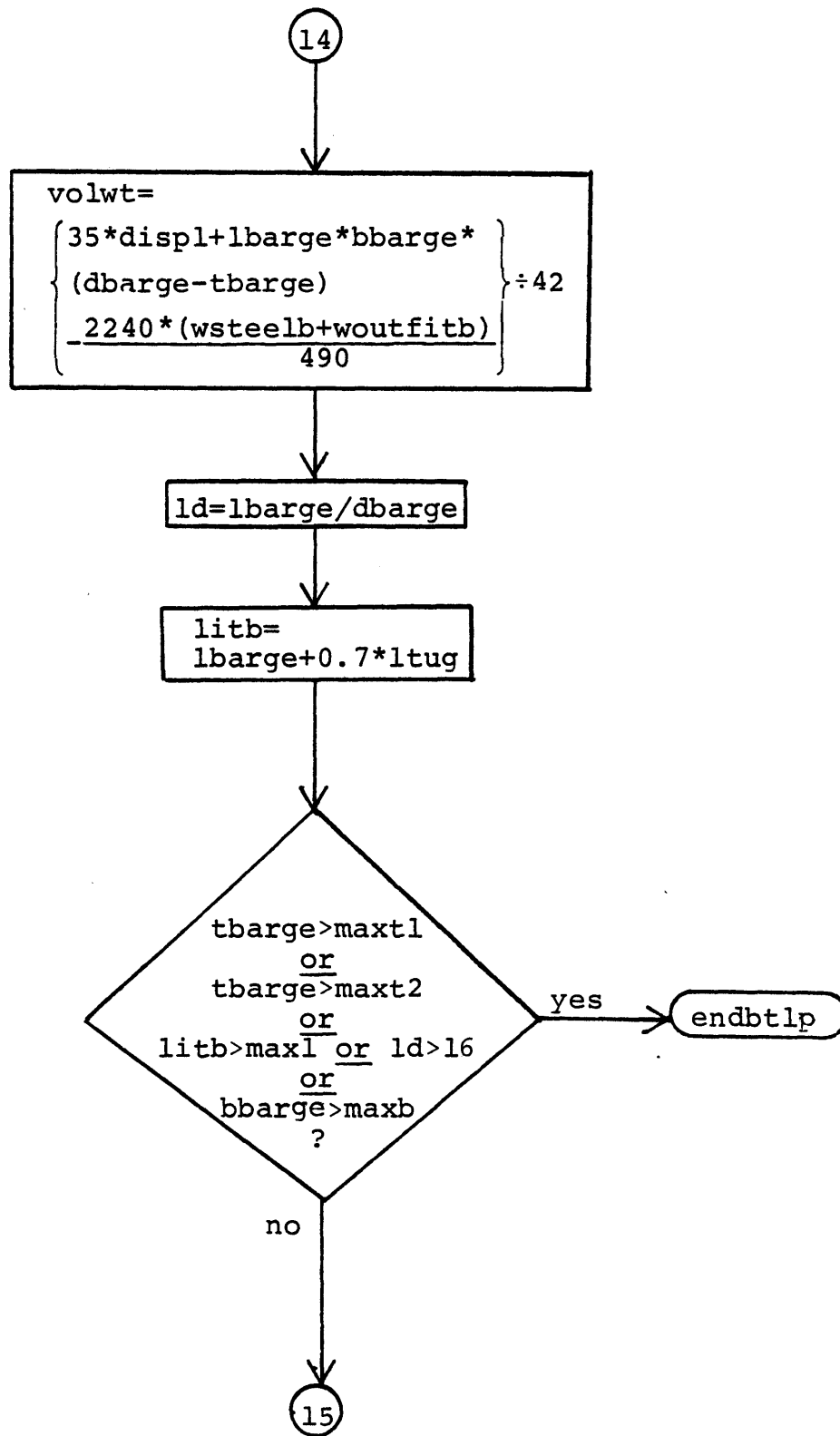


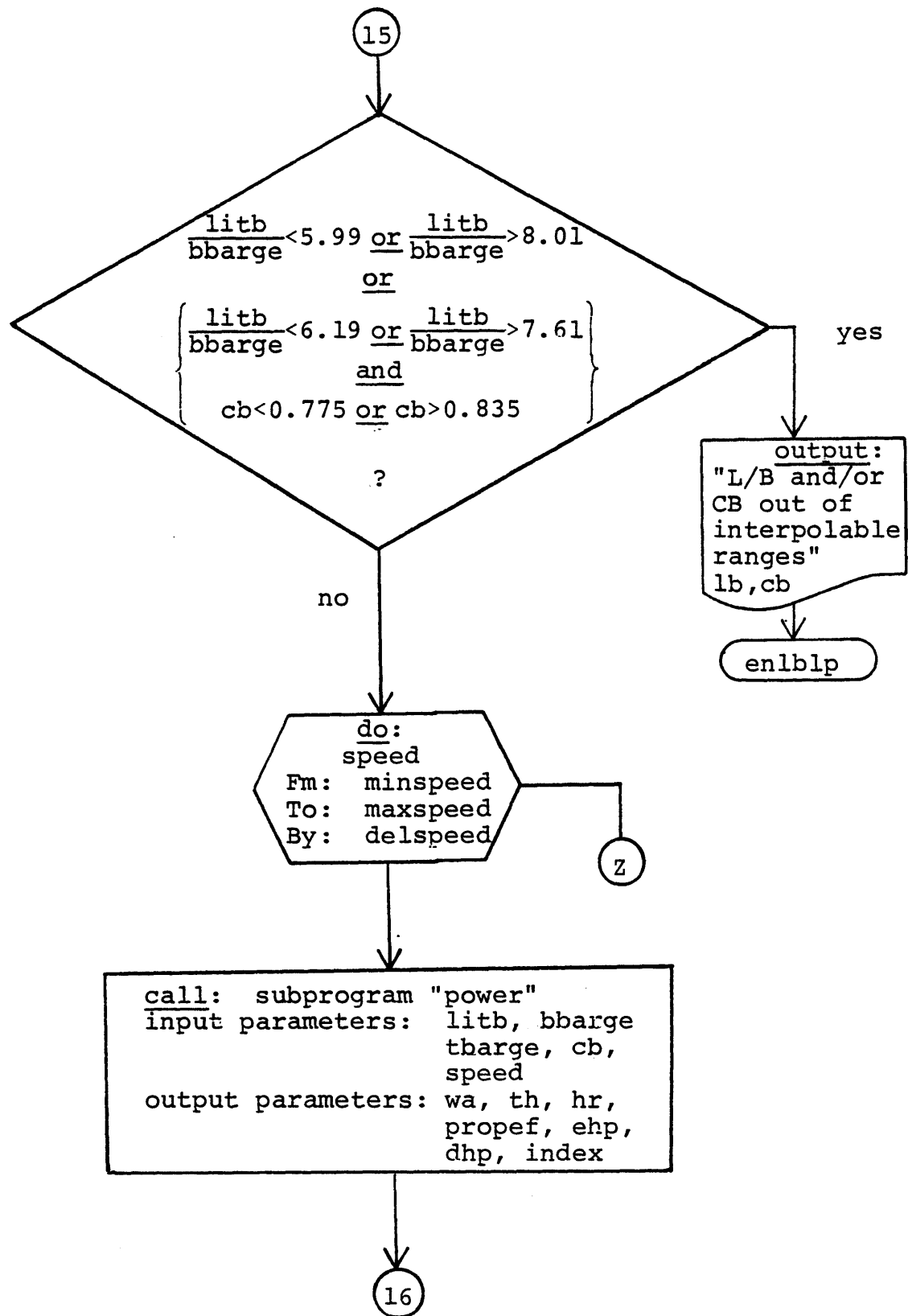


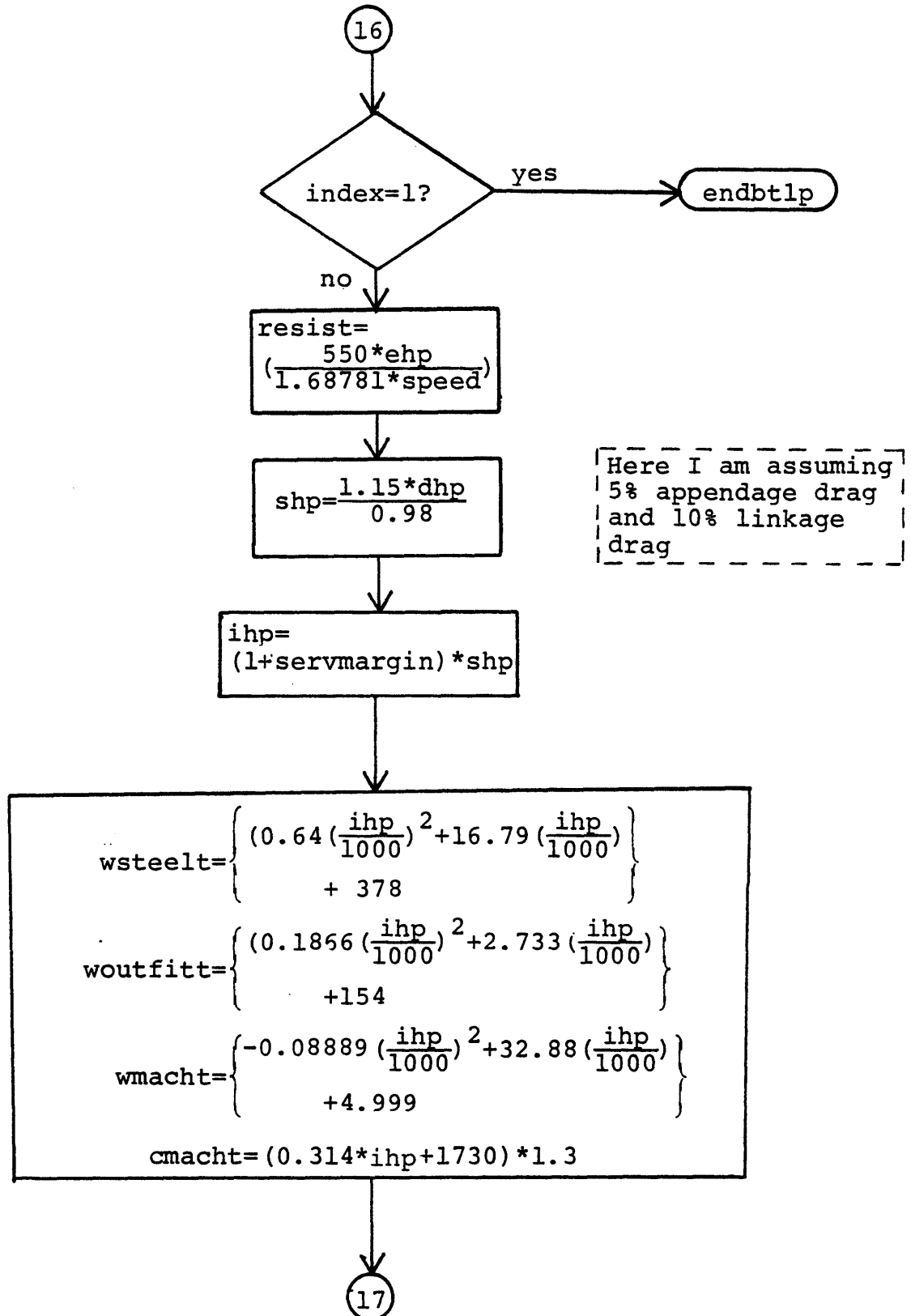


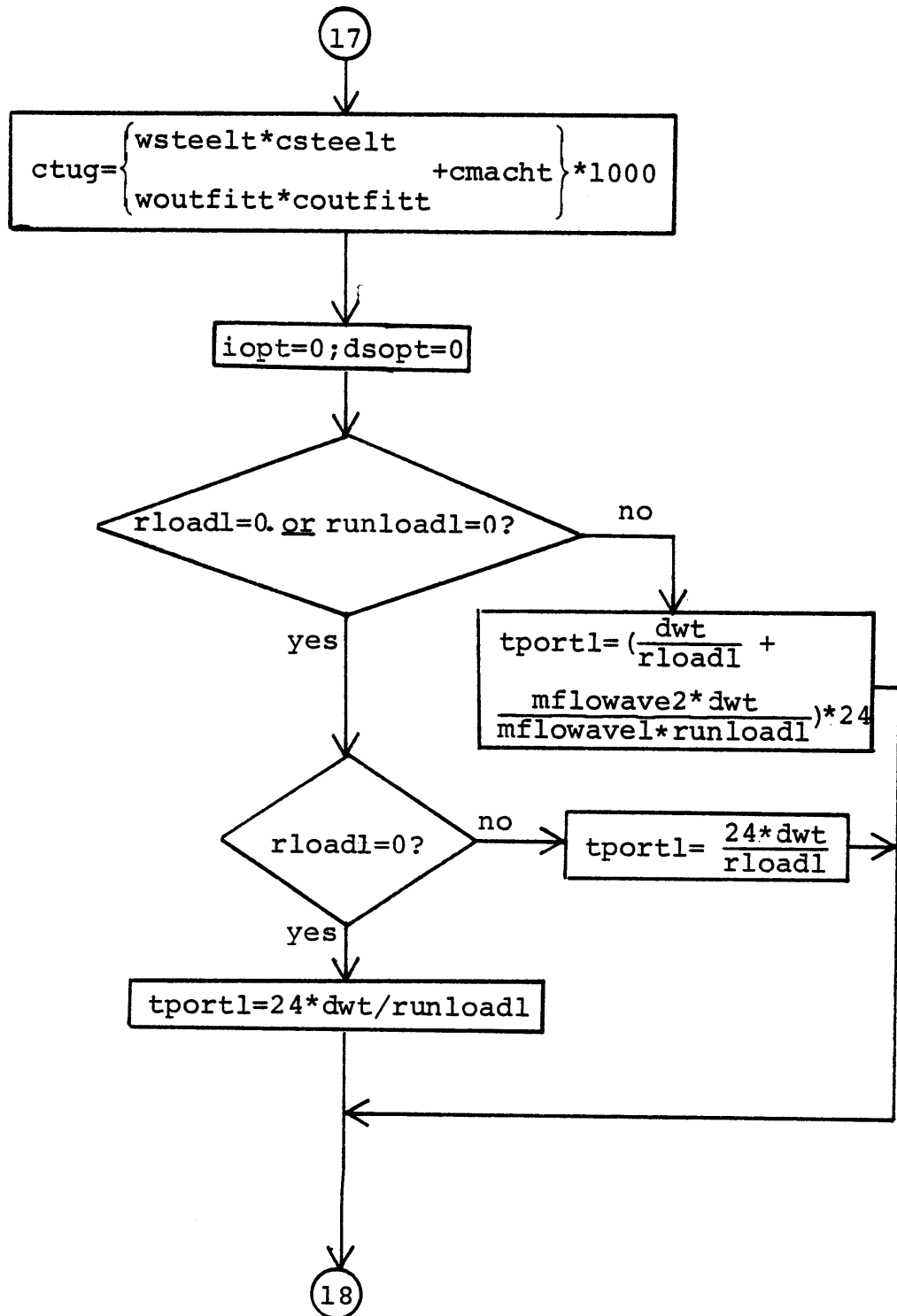


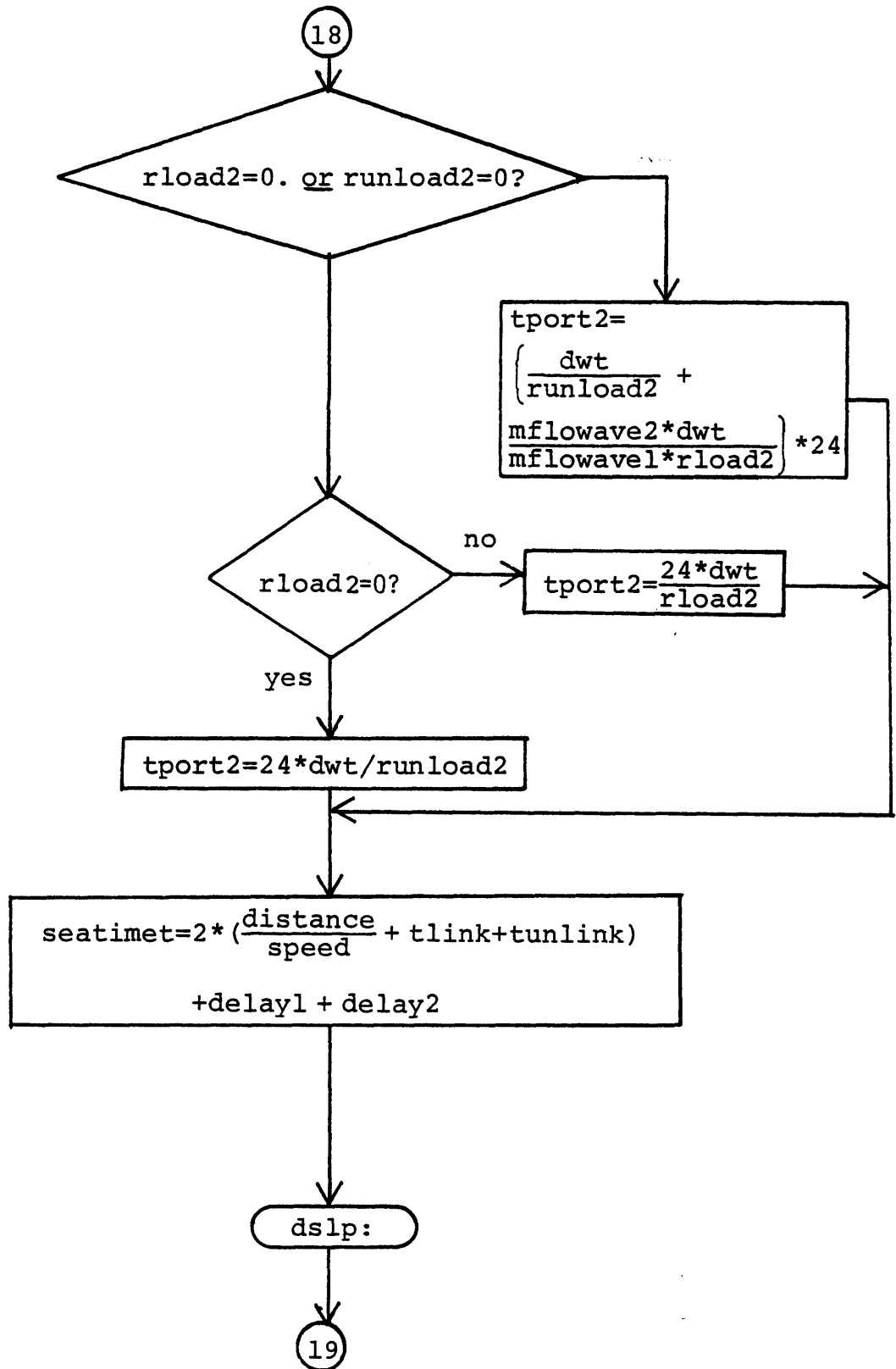


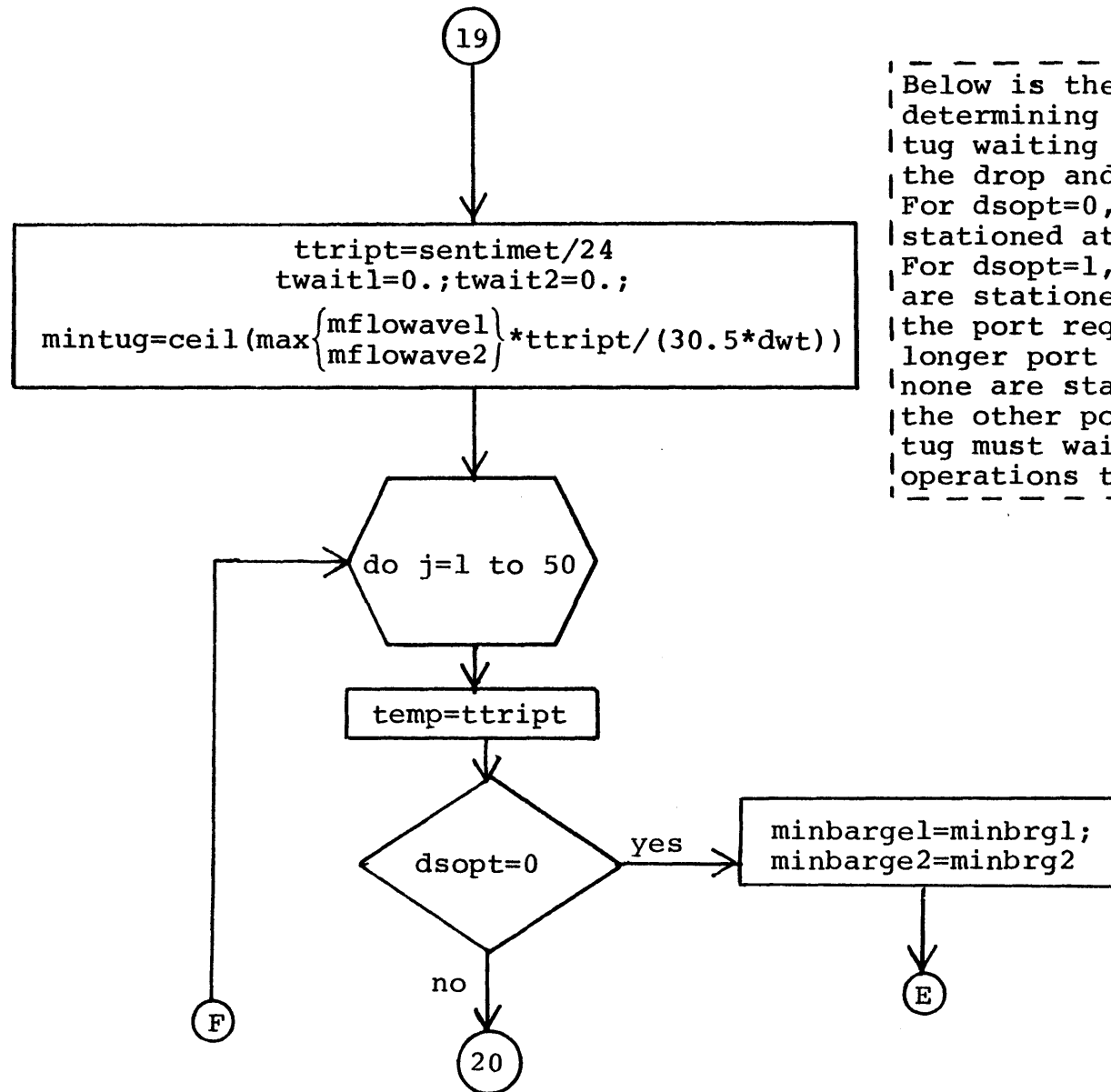




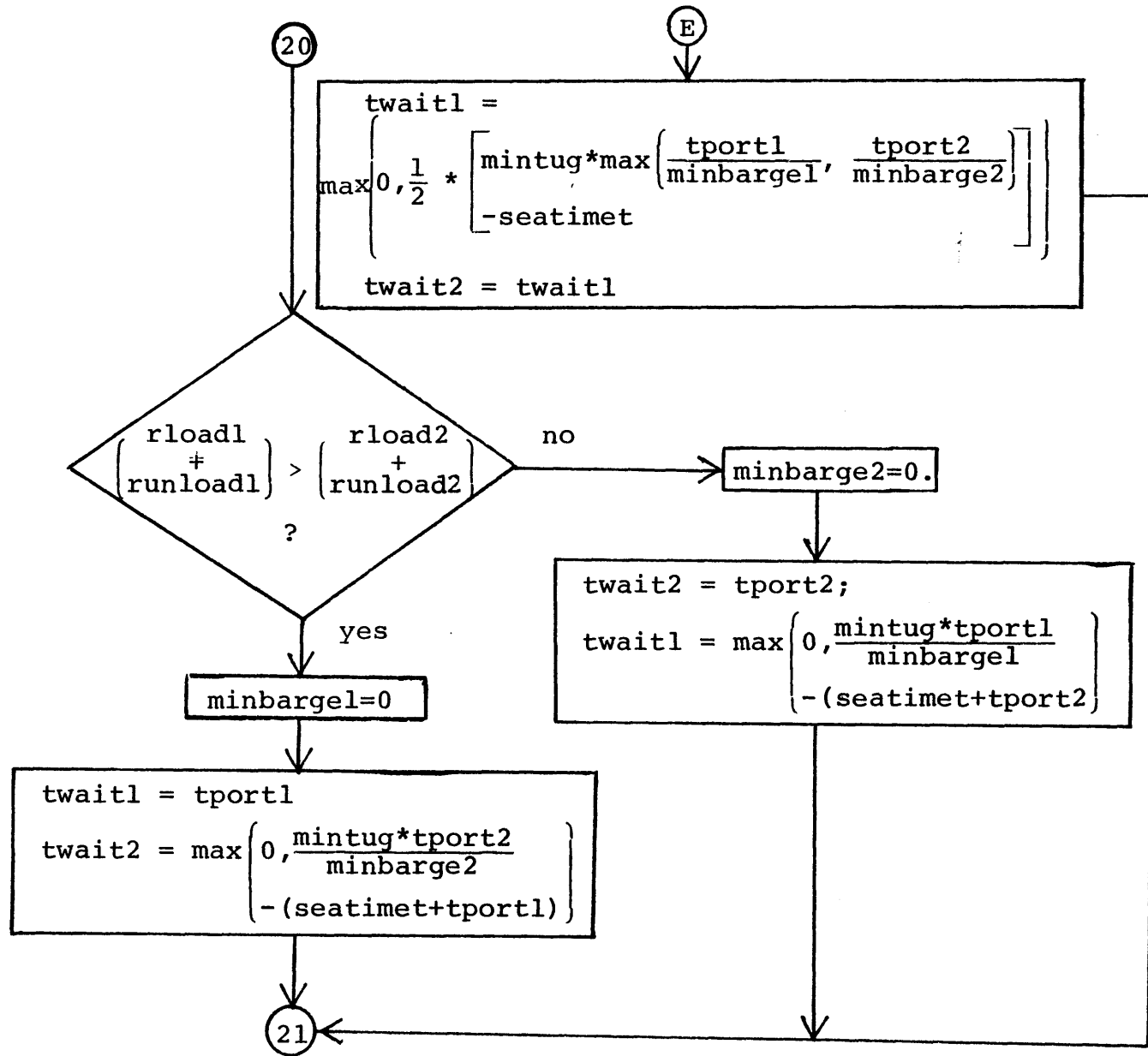


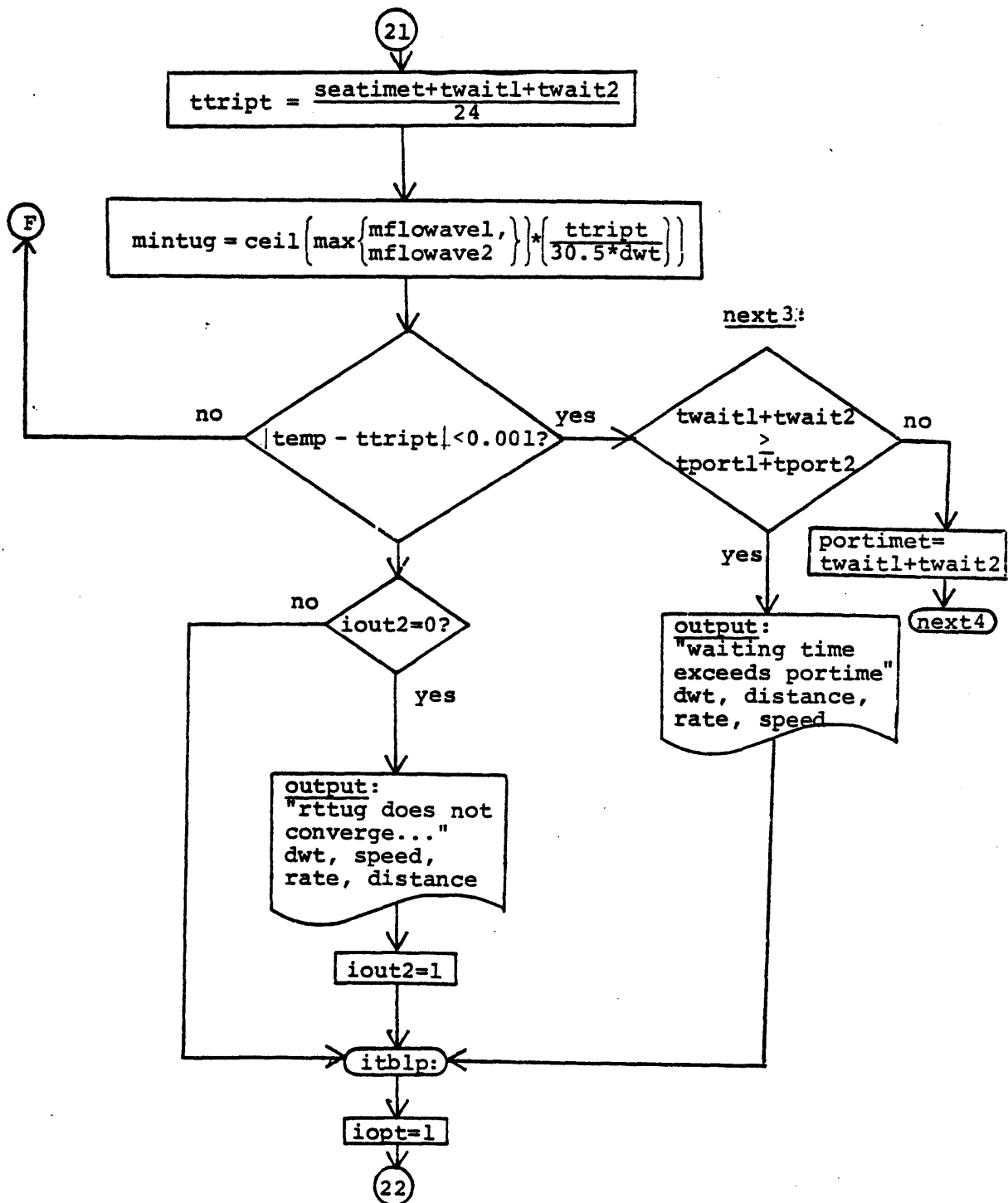


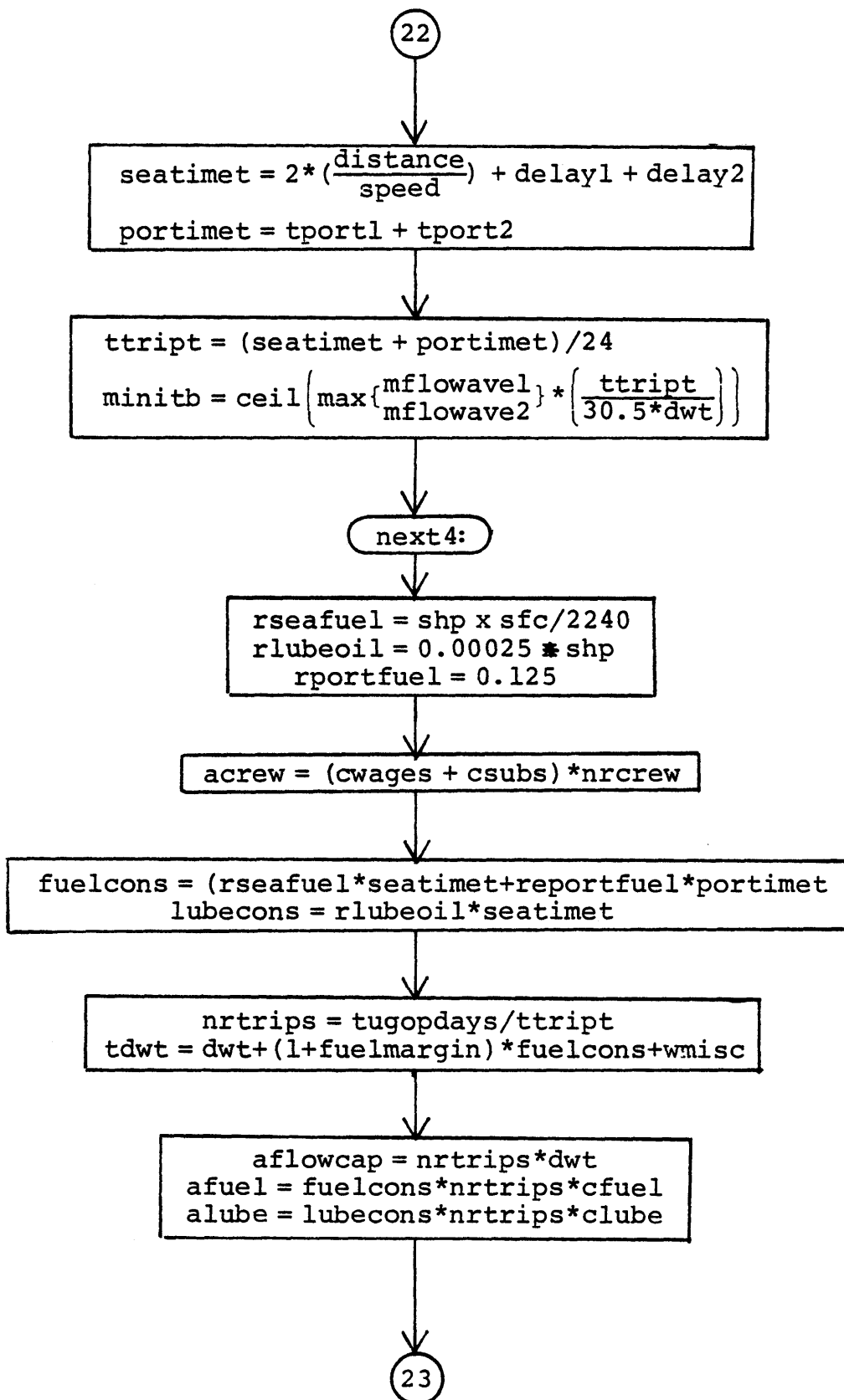


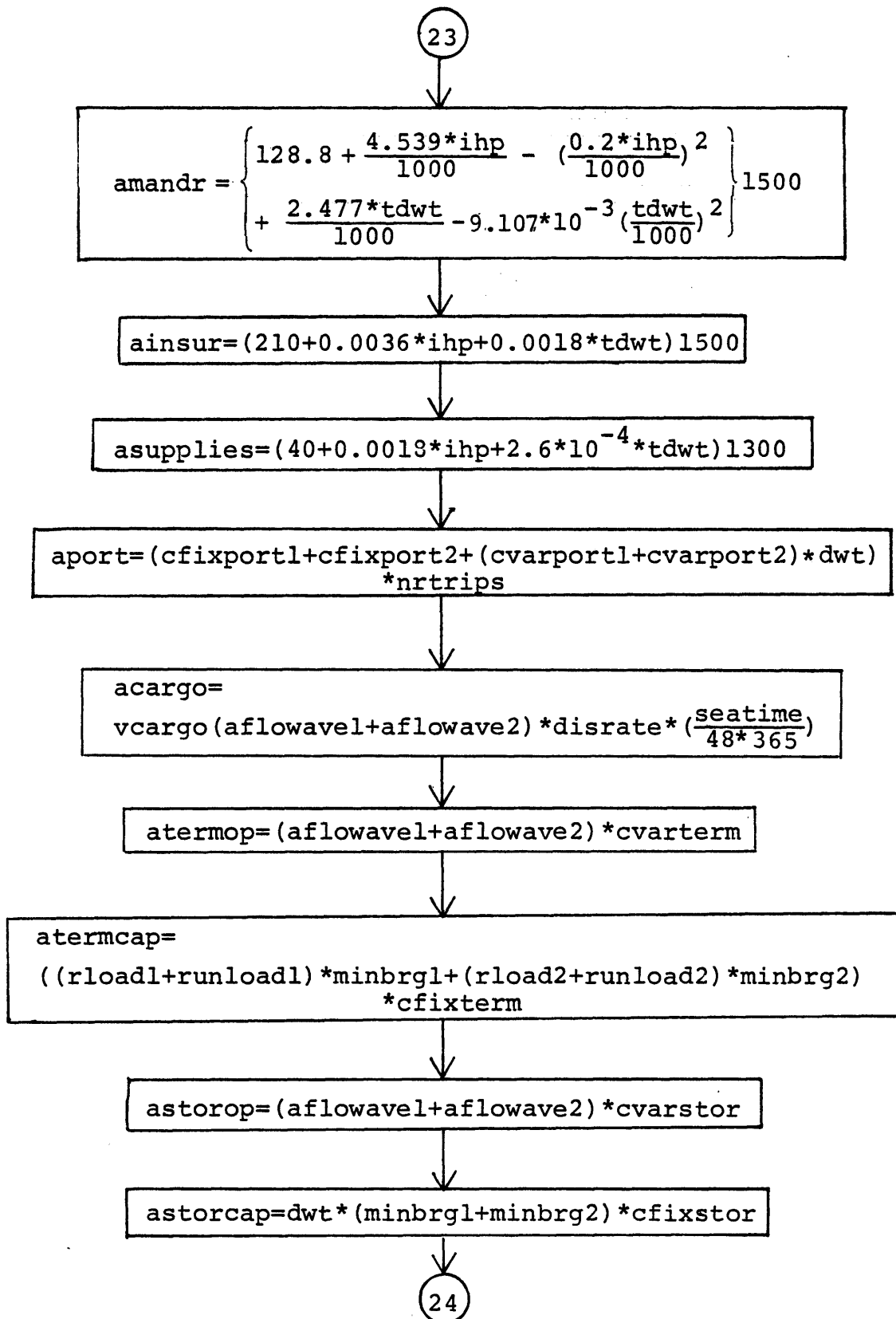


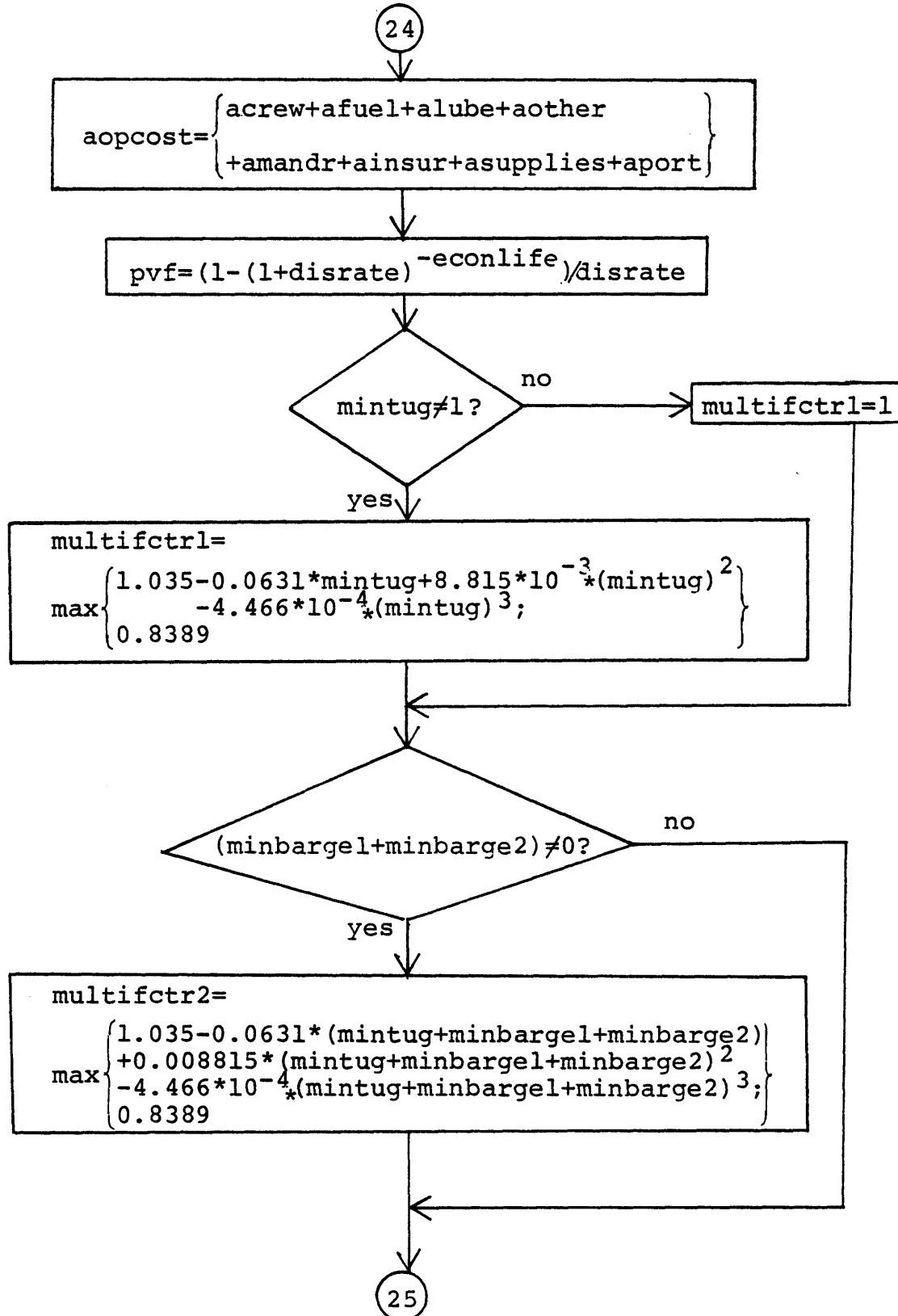
Below is the logic for determining the required tug waiting time for the drop and swap mode. For dsopt=0, barges are stationed at both ports. For dsopt=1, barges are stationed only at the port requiring longer port operations, none are stationed at the other port so the tug must wait for cargo operations there.

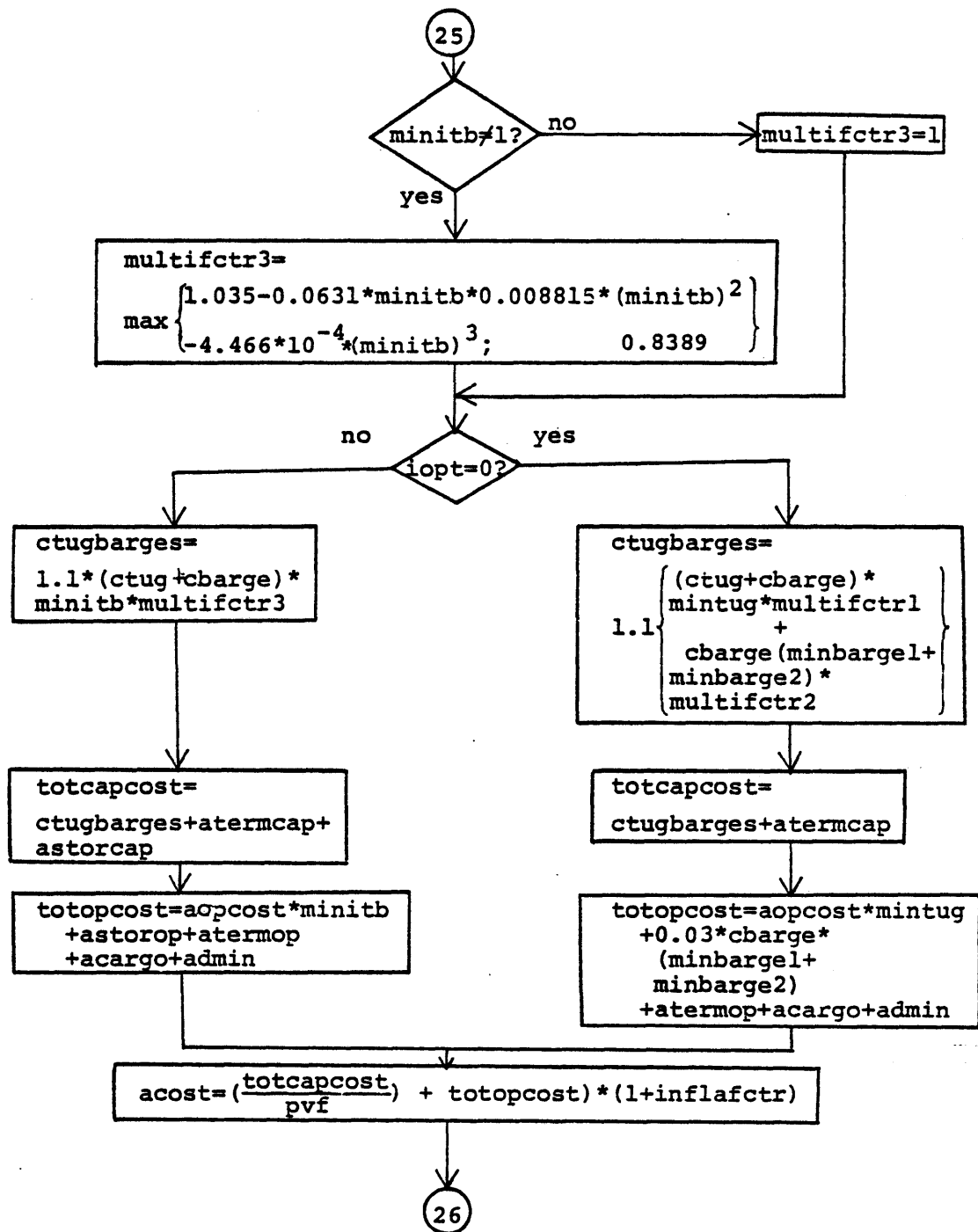


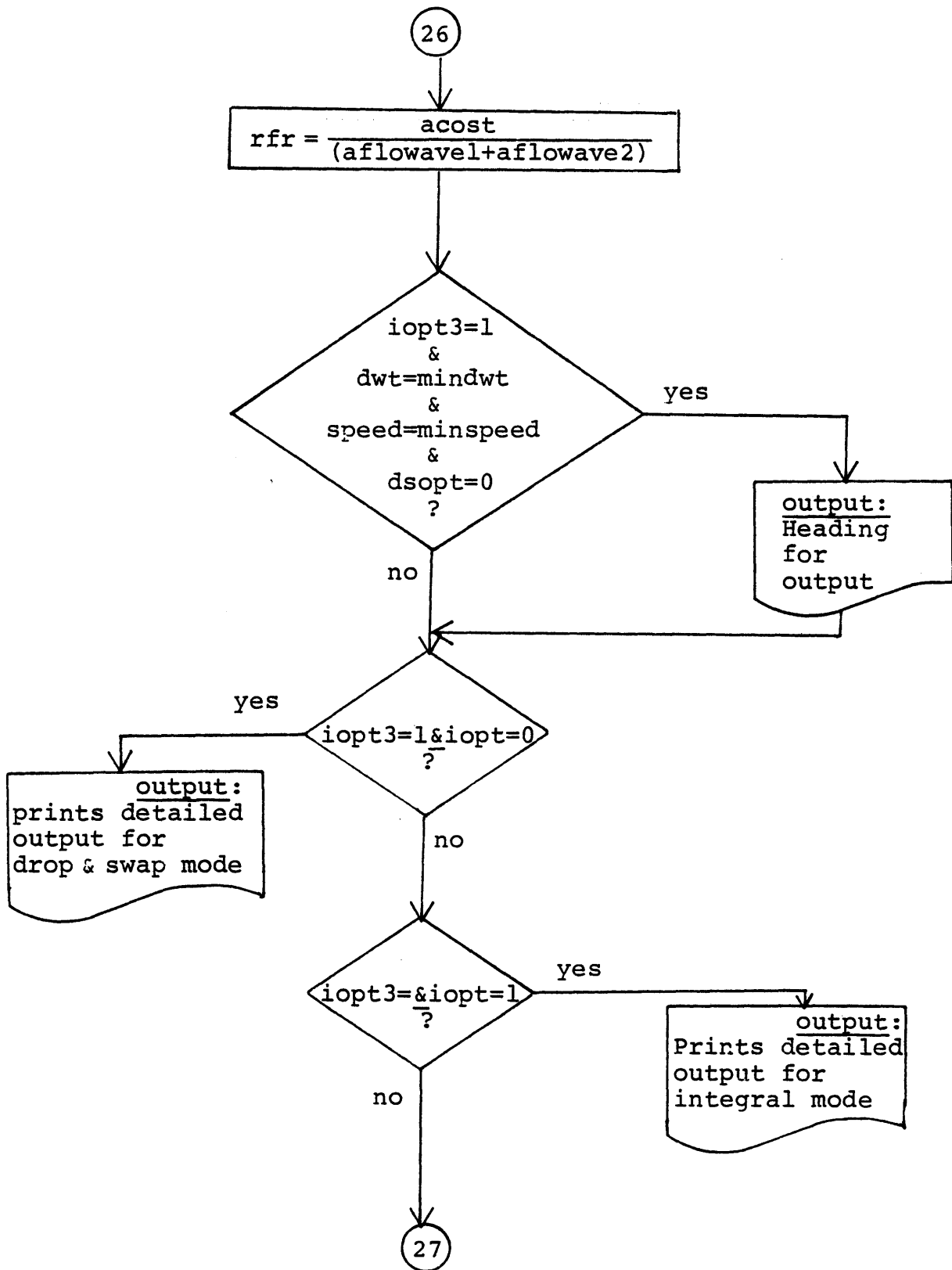


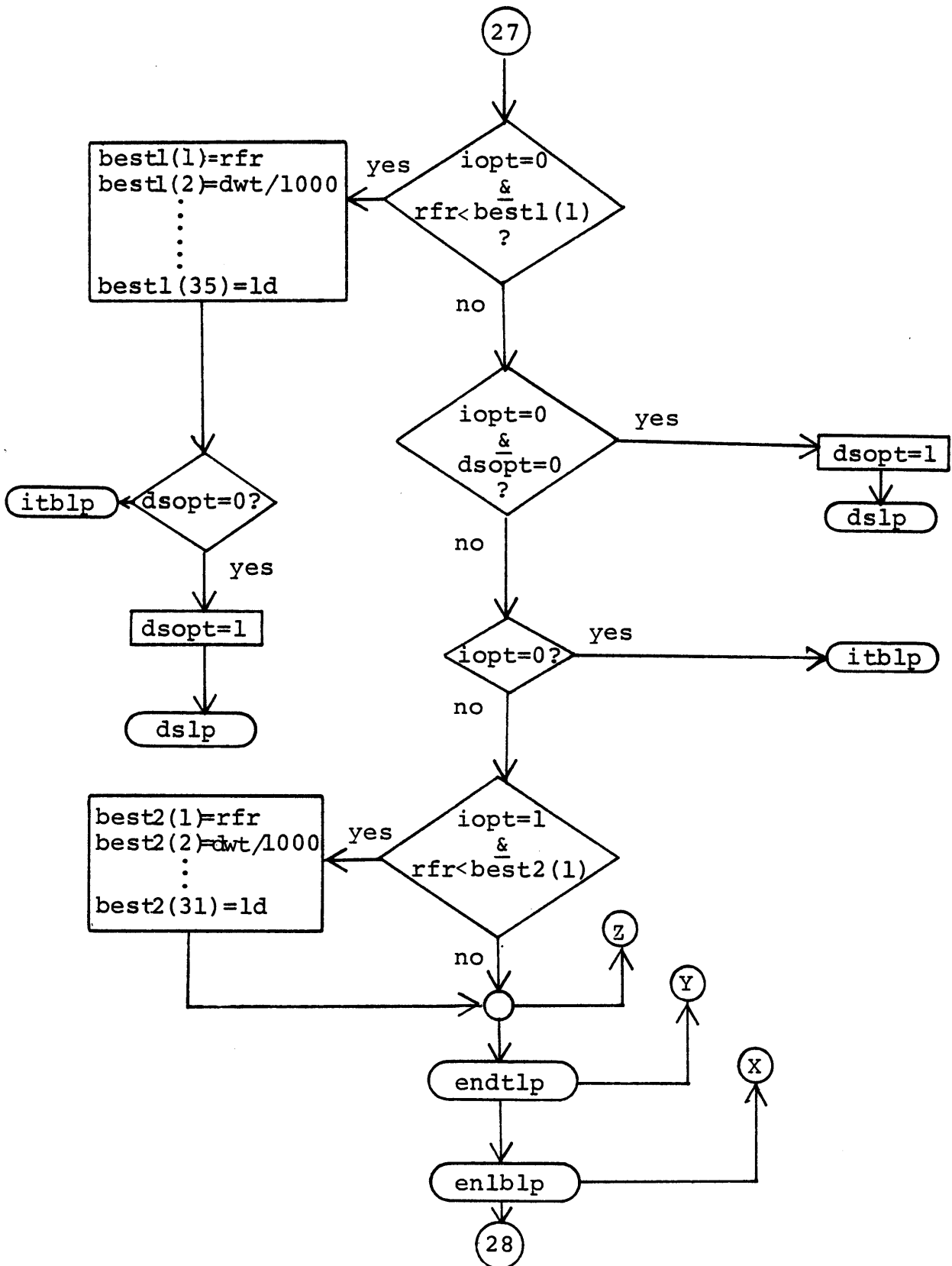


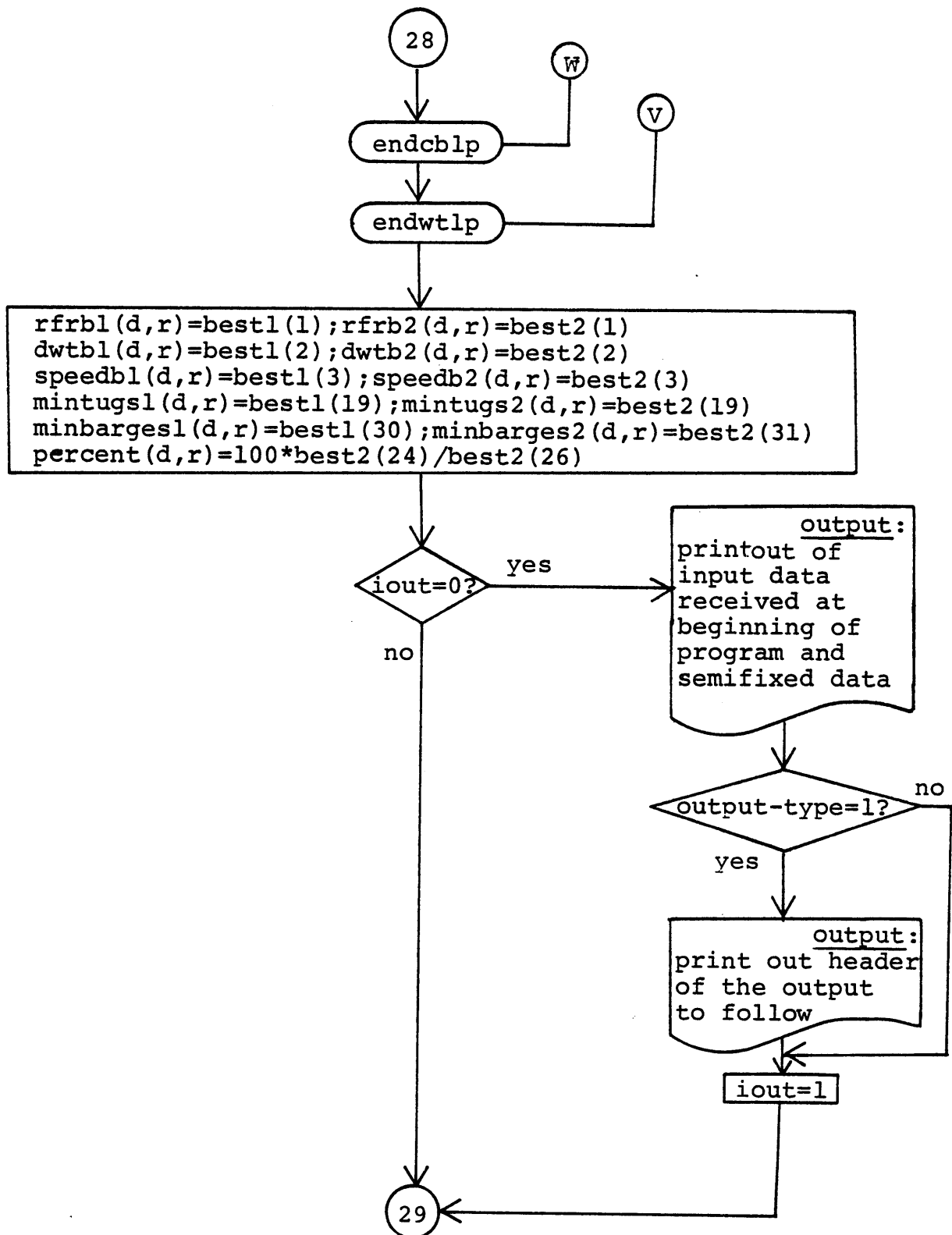


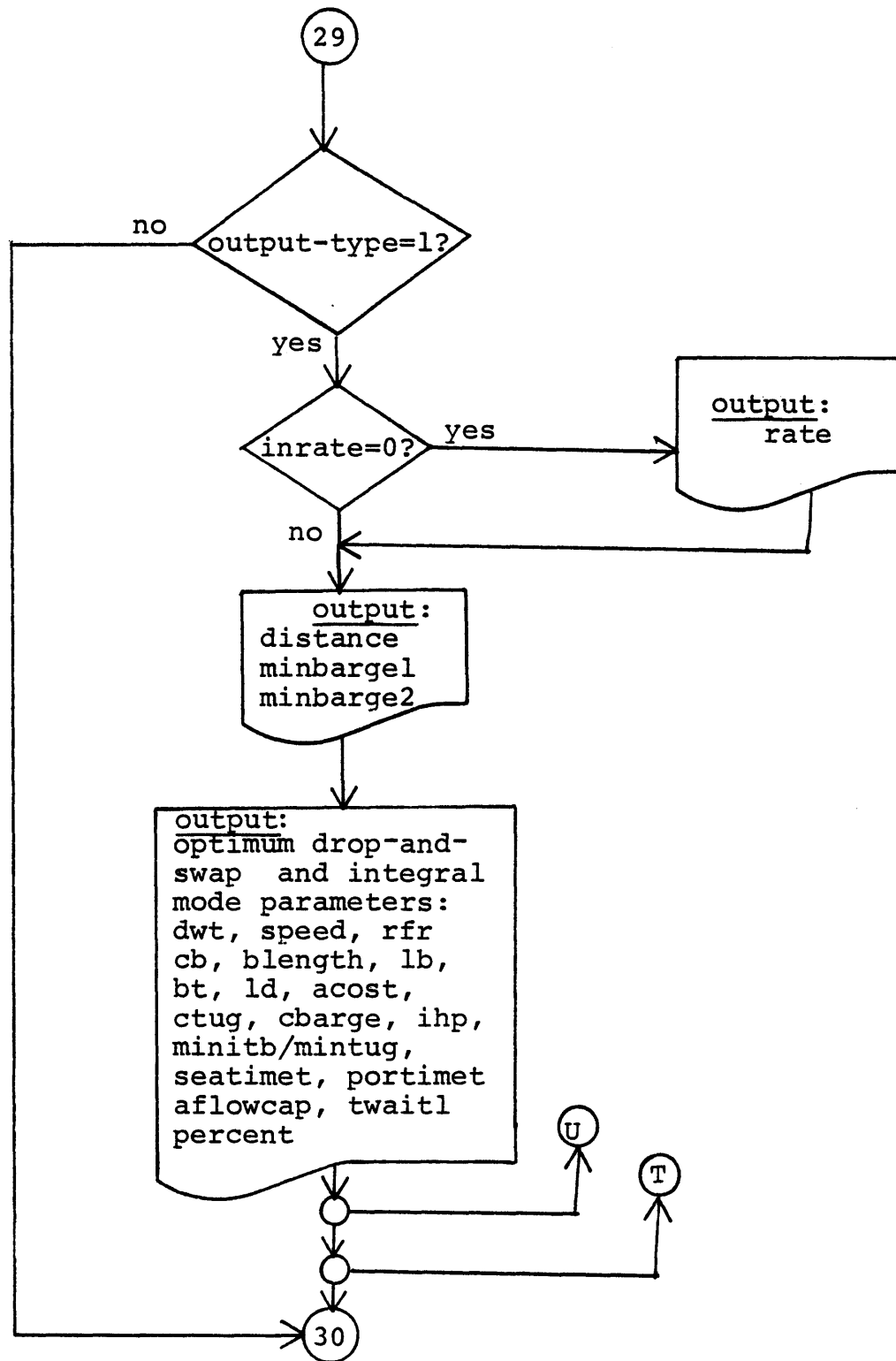


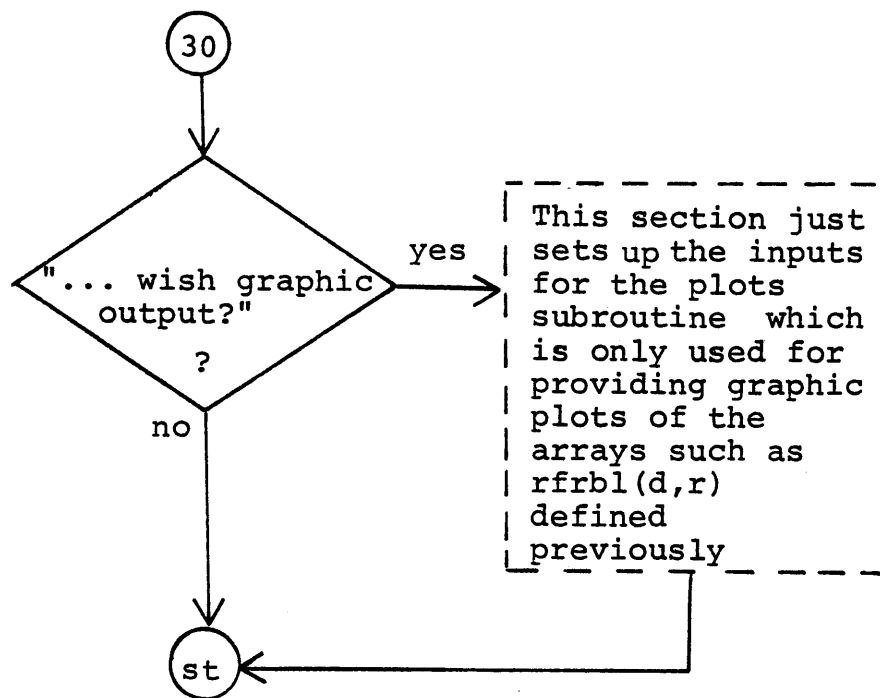













```

1 drop_and_swap: proc;
2 dcl(rfrb1, rfrb2, dwtb1, dwtb2, speedb1, speedb2, mintugs1, mintugs2,
3   minbarges1, minbarges2, percent)(25, 50) float bin;
4 dcl(distance, speed, dwt, tdwt, max1, maxb, maxt1, maxt2, rate,
5   ltug, lbarge, litb, bbarge, dbarge, tbarge, displ,
6   rload1, rload2, runload1, runload2, delay1, delay2,
7   aflowave1, aflowave2, mflowave1, mflowave2,
8   tugopdays, bargeopdays, tlink, tunlink, twait1, twait2,
9   servmargin, fuelmargin, sfc, temp, shp, ihp,
10  resist, fuelcons, lubecons, multifctr1, multifctr2, multifctr3,
11  wsteelt, woutfitt, wmacht, wmisc, esteelt, coutfitt, emacht,
12  nrcrew, cwages, csubs, ctug, cbarge, ctugbarges, cfuel, clube,
13  cfixport1, cfixport2, cvarport1, cvarport2,
14  cvarstor, cfixstor, cvarterm, cfixterm,
15  minbargel, minbargel2, minbrg1, minbrg2, mintug, minitb, tport1, tport2,
16  rseafuel, rlubeoil, rportfuel, seatimet, portimet, ttript,
17  disrate, pvf, inflafctr, vcargo, econlife, rfr, nrtrips, best1(50), best2(50),
18  acrew, afuel, alube, amandr, ainsur, asupplies, aport, atermpop, atermpcap, astorop, astorcap, acargo, aflowcap,
19  admin, aother, aopcost, totopcost, totcapcost, acost) float bin;
20 dcl(llcoef(66), c1(39), c2(47), dt(9), dq(13), rc(2, 8, 7, 8), t(15, 15), w(15, 15), eta(15, 15)) float bin;
21 dcl(lbargel75, lbargel80, lbargel85, xcb, x1cb, x2cb, x3cb, dhp, wa, th, hr, propef, ehpf) float bin;
22 dcl(hullwt75, hullwt80, hullwt85, wsteeltb, woutfittb, esteeltb, coutfittb) bin float;
23 dcl(mincb, maxcb, delcb, minlb, maxlb, dellb, minbt, maxbt, delbt, lb, bt, cb) dec float;
24 dcl(minfbd, ldfact, shrfact, fbd, ld, volwt) bin float;
25 dcl(minspeed, maxspeed, delspeed, mindwt, maxdwt, deldwt, mindwttemp, mindist, maxdist, deldist,
26   minrate, maxrate, delrate, drange, rrange, d, r, countr) fixed bin;
27 dcl(iopt, i, j, k, m, iout, iout2, iout3, iopt3, output_type, inrate, dsopt, index) fixed bin;
28 dcl nextchar char(1);
29 dcl power entry(bin float, bin float, bin float, bin float, bin float, bin float, bin float,
30   cin float, bin float, bin float, bin float, bin fixed, dim(39) bin float,
31   dim(47) bin float, dim(9) bin float, dim(13) bin float, dim(2, 8, 7, 8) bin float,
32   dim(15, 15) bin float, dim(15, 15) bin float, dim(15, 15) bin float) external;
33 dcl ibit fixed bin static;
34 dcl(parameters, llfact, powerdata, sysin) file stream input, sysprint file stream output;
35 open file(llfact) title("vfile_loadline.data");
36 get file(llfact) list(llcoef);
37 close file(llfact);
38 open file(powerdata) title("vfile_power.data");
39 get skip(4) file(powerdata) edit(((rc(i, j, k, m) do m = 1 to 8) do k = 1 to 7) do j = 1 to 8) do i = 1 to 2));
40   (skip(2), 7(col(1), x(5), 8 f(5)));
41 get skip(5) file(powerdata) edit(((w(i, j) do j = 1 to 15) do i = 1 to 15))(col(1), x(5), 15 f(5));
42 get skip(5) file(powerdata) edit(((eta(i, j) do j = 1 to 15) do i = 1 to 15))(col(1), x(5), 15 f(6));
43 get skip(5) file(powerdata) edit(((t(i, j) do j = 1 to 15) do i = 1 to 15))(col(1), x(5), 15 f(5));
44 get skip(3) file(powerdata) list(c1, c2, dt, dq);
45 close file(powerdata);
46 /* ----- BEGIN include file plot_entry_dels.incl.pl1 ----- */
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

1  16      Tick marks initial (0),                      /* grid_sw; fifth arg to plot_$setup */
1  17      Dotted grid initial (1),
1  18      Solid grid initial (2),
1  19      No_grid initial (3),
1  20
1  21      Normal_scaling initial (0),                    /* eq_scale_sw; sixth arg to plot_$setup */
1  22      Equal_scaling initial (1) fixed bin static options (constant);
1  23
1  24 /* ----- END include file plot_entry_dcls.incl.pl1 ----- */
1  46
1  47      /* input basic changeable parameters via get list statements */
1  48 st: put list("Input via list format the following parameters:");
1  49 put skip(2) edit("Do you wish to specify individual port L-D rates?: ")
1  50 (col(1), a);
1  51 call yesno;
1  52 irrate = ibit;
1  53 if irrate = 1 then do;
1  54     put skip edit("rload1, runload1, rload2, runload2: ")(a);
1  55     get list(rload1, runload1, rload2, runload2);
1  56     delrate = 1;
1  57     minrate = 0; maxrate = 0;
1  58     go to sp;
1  59 end;
1  60 put skip edit("minrate, maxrate, delrate: ")(a);
1  61 get list(minrate, maxrate, delrate);
1  62 if delrate = 0 then delrate = 1000;
1  63 sp: put skip edit("mindist, maxdist, deldist: ")(a);
1  64 get list(mindist, maxdist, deldist);
1  65 if deldist = 0 then deldist = 1000;
1  66 put skip edit("minspeed, maxspeed, delspeed: ")(a);
1  67 get list(minspeed, maxspeed, delspeed);
1  68 if delspeed = 0 then delspeed = 1;
1  69 put skip edit("mindwt, maxdwt, deldwt: ")(a);
1  70 get list(mindwt, maxdwt, deldwt);
1  71 if deldwt = 0 then deldwt = 5000;
1  72 mindwttemp = mindwt;
1  73 put skip edit("aflowave1, aflowave2: ")(a);
1  74 get list(aflowave1, aflowave2);
1  75 if aflowave1 = 0.e0 then do;
1  76     rload1 = 0.e0;
1  77     runload2 = 0.e0;
1  78 end;
1  79 if aflowave2 = 0.e0 then do;
1  80     rload2 = 0.e0;
1  81     runload1 = 0.e0;
1  82 end;
1  83      /* input form parameters of tug-barge system */
1  84 st2:
1  85 put skip edit("mincb, maxcb, delcb: ")(a);
1  86 get list(mincb, maxcb, delcb);
1  87 if delcb = 0.e0 then delcb = 0.1;
1  88 put skip edit("minlb, maxlb, dellb: ")(a);
1  89 get list(minlb, maxlb, dellb);
1  90 if dellb = 0.e0 then dellb = 0.2;
1  91 put skip edit("minbt, maxbt, delbt: ")(a);
1  92 get list(minbt, maxbt, delbt);
1  93 if delbt = 0.e0 then delbt = 0.1;
1  94 if minbt<2.0 ; minbt>3.25 ; maxbt>3.25 ; maxbt<2.0 ; mincb<0.75 ; mincb>0.85 ;
1  95 maxcb<0.75 ; maxcb>0.85 ; minlb>8.0 ; maxlb>8.0 then do;
1  96     put skip list("Form coefficients are out of interpolable range. Please specify:");

```

```

97 put skip list("0.75-cb-0.85    6.0-L/B-8.0    2.0-B/T-3.25");
98 iout3 = 1;
99 go to st2;
100 end;
101 if iout3 = 1 then do;
102 iout3 = 0;
103 go to st3;
104 end;
105 /* input basic semifixed parameter values via parameter file */
106 open file(parameters) title("vfile semifixedparams.data");
107 get file(parameters) edit(tugopdays, bargeopdays, tlink, tunlink,
108 servmargin, fuelfmargin, sfc, cfuel, clube, nrcrew,
109 cwages, csubs, csteelt, coutfitt, csteelb, coutfitb, ltug, wmisc, aother,
110 admin, cfixport1, cfixport2, cvarport1, cvarport2, cfixterm, cvarterm, cfixstor, cvarstor,
111 delay1, delay2, max1, maxb, maxt1, maxt2,
112 disrate, econlife, inflafctr, vcargo)(skip(3), 38(col(21), f(12, 2)));
113 close file(parameters);
114 put skip edit("input changes to semifixed data via get data format")(a);
115 get data(tugopdays, bargeopdays, tlink, tunlink,
116 servmargin, fuelfmargin, sfc, cfuel, clube, nrcrew,
117 cwages, csubs, csteelt, coutfitt, csteelb, coutfitb, ltug, wmisc, aother,
118 admin, cfixport1, cfixport2, cvarport1, cvarport2, cfixterm, cvarterm, cfixstor, cvarstor,
119 delay1, delay2, max1, maxb, maxt1, maxt2,
120 disrate, econlife, inflafctr, vcargo);
121 put skip edit("Do you want printed output?: ")(col(1), a);
122 call yesno;
123 output_type = ibit;
124 if output_type = 1 then do;
125 put skip edit("Do you want detailed output?: ")(col(1), a);
126 call yesno;
127 iout3 = ibit;
128 end;
129 /* this segment initializes various variables */
130 st3: iout = 0;
131 if(maxdist - mindist) < 1 ; deldist < 1 then drange = 1;
132 else drange = (maxdist - mindist)/deldist + 1;
133 if(maxrate - minrate) < 1 ; delrate < 1 then rrange = 1;
134 else rrange = (maxrate - minrate)/delrate + 1;
135 do distance = mindist to maxdist by deldist;
136 d = (distance - mindist)/deldist + 1;
137 do rate = minrate to maxrate by delrate;
138 r = (rate - minrate)/delrate + 1;
139 if inrate = 0 then do;
140 rload1 = rate;
141 rload2 = rate;
142 runload1 = rate;
143 runload2 = rate;
144 end;
145 iout2 = 0;
146 if rate ^= minrate then
147 mindwt = max(1.0e3*min(best1(2), best2(2))-2.*deldwt, mindwttemp);
148 else mindwt = mindwttemp;
149 best1 = 0.0e0;
150 best2 = 0.0e0;
151 best1(1) = 9.99e2;
152 best2(1) = 9.99e2;
153 mflowave1 = (aflowave1*3.65e2)/(1.2e1*bargeopdays);
154 mflowave2 = (aflowave2*3.65e2)/(1.2e1*bargeopdays);
155 if rload1 = 0.e0 ; runload1 = 0.e0 then
156 if rload1 = 0.e0 then minbarge1 = ceil(mflowave2/(3.05e1*runload1));

```

```

157     else minbargel = ceil(mflowave1/(3.05e1*rload1));
158     else minbargel = ceil(mflowave1/(3.05e1*rload1) + mflowave2/(3.05e1*runload1));
159     if rload2 = 0.e0 | runload2 = 0.e0 then
160         if rload2 = 0.e0 then minbargel2 = ceil(mflowave1/(3.05e1*runload2));
161         else minbargel2 = ceil(mflowave2/(3.05e1*rload2));
162     else minbargel2 = ceil(mflowave2/(3.05e1*rload2) + mflowave1/(3.05e1*runload2));
163     minbrg1 = minbargel; minbrg2 = minbargel2;
164     /* outermost loop-iterate barge size */
165 dwt1p:
166     do dwt = mindwt to maxdwt by deldwt;
167     /* in this section the cost of the barge and its principal dimensions must be */
168     /* determined as a function of cdwt. For the time being empirical relations */
169     /* will be used until a formal costing routine is developed. */
170     do cb = mincb to maxcb by delcb;
171         xcb = 2.*(cb - 0.75);
172         x1cb = 1.0 + (xcb-3.0)*xcb/2.0;
173         x2cb = (xcb-2.0)*xcb;
174         x3cb = (xcb-1.0)*xcb/2.0;
175         do lb = minlb to maxlb by dellb;
176             do bt = minbt to maxbt by delbt;
177                 lbarge75 = exp(1.128)*lb**0.742*bt**0.382*dwt**0.336;
178                 lbarge80 = exp(1.108)*lb**0.739*bt**0.381*dwt**0.336;
179                 lbarge85 = exp(1.088)*lb**0.737*bt**0.379*dwt**0.336;
180                 lbarge = lbarge75*x1cb - lbarge80*x2cb + lbarge85*x3cb;
181                 hullwt75 = exp(-6.206)*bt**0.884*lb**1.348*dwt**1.104;
182                 hullwt80 = exp(-6.396)*bt**0.909*lb**1.368*dwt**1.111;
183                 hullwt85 = exp(-6.569)*bt**0.930*lb**1.391*dwt**1.117;
184                 wsteelb = hullwt75*x1cb - hullwt80*x2cb + hullwt85*x3cb;
185                 do j = 1 to 50;
186                     bbarge = (lbarge + 0.7*ltug)/lb;
187                     tbarge = bbarge/bt;
188                     temp = cb*lbarge*bbarge*tbarge/35.;
189                     woutfitb = max(1.496*lbarge - 284.24, 50.);
190                     displ = woutfitb + wsteelb + dwt;
191                     lbarge = (35*displ*(lbarge/bbarge)**2*bt/cb)**0.333333333;
192                     if abs(temp - displ) < 1.e0 then go to lloop;
193                 end;
194                 put skip list("Barge length routine does not converge.");
195 lloop:
196                 if lbarge > 750. then do;
197                     put skip edit("Barge length exceeds 750'", "lbarge = ", lbarge, "dwt = ", dwt)
198                     (a, a, f(6), a, f(6));
199                     go to endcblp;
200                 end;
201                 cbarge = 1.0e3*(wsteelb*csteelb + woutfitb*coutfitb);
202                 if lbarge = 750. then minfbd = llcoef(66);
203                 else do;
204                     countr = ceil((lbarge-99.99999)/10.);
205                     minfbd = llcoef(countr) + (llcoef(countr+1)-llcoef(countr))*(lbarge-90.-10.*countr)/10.;
206                 end;
207                 minfbd = minfbd*(cb + 0.68)/1.38;
208                 if (lbarge/(minfbd/12. + tbarge)) < 15. then do;
209                     if lbarge < 393.6 then ldfact = (tbarge + minfbd/12. - lbarge/15.)*(lbarge/131.2)/(1-lbarge/(12.*131.2))
210                     else ldfact = (tbarge + minfbd/12. - lbarge/15.)*4.;
211                 end;
212                 else ldfact = 0.;
213                 shrfact = 0.0375*lbarge + 3.75;
214                 fbd = 0.75*(minfbd + ldfact + shrfact);
215                 dbarge = (tbarge + fbd/12.);
216                 volwt = (35.*displ + lbarge*bbarge*(dbarge-tbarge) - 2240.*(wsteelb+woutfitb)/490.)/42.;

```

```

217         ld = lbarge/dbarge;
218         litb = lbarge + 7.e-1*ltug;
219         if(tbarge>maxt1) |(tbarge>maxt2) |(litb>max1) |(bbarge>maxb) |(ld > 16.)
220         then go to endbtlp;
221     /* intermediate loop-iterate over tug/barge speed */
222     /* calculation of tug IHP */
223     if litb/bbarge < 5.99 | litb/bbarge > 8.01 |(((litb/bbarge < 6.19 | litb/bbarge > 7.61) &(cb < 0.775 | cb
\c > 0.835))
224     then do;
225     /*      put skip edit("L/B and/or CB are out of interpolable ranges: ", "L/B = ", lb, ", CB = ", cb,
\c/
226     /*      ", litb/bbarge = ", litb/bbarge)(a, 3(a,f(5,2)));      */
227     go to endlbtp;
228     end;
229     do speed = minspeed to maxspeed by delspeed;
230     /* this is where the external subroutine power is called to determine dhp */
231     call power(litb, bbarge, tbarge, cb, speed, wa, th, hr, propof, ehp, dhp, index, c1, c2, dt, dq, rc,
\c, w, eta);
232     if index = 1 then do;
233     go to endbtlp;
234     end;
235     resist = 550.*ehp/((1.68781*speed);
236     /* here I am assuming 5% appendage drage and 10% linkage drag */
237     shp = 1.15*dhp/0.98;
238     ihp = (1.0e0 + servmargin)*shp;
239     /* calculate tug cost using univ of michigan study-July 77 --single screw tug */
240     wsteelt = (6.4e-1*(ihp/1.0e3)**2 + 1.679e1*ihp/1.0e3 + 3.78e2);
241     woutfitt = (1.866e-1*(ihp/1.0e3)**2 + 2.733e0*ihp/1.0e3 + 1.54e2);
242     wmacht = (-8.889e-2*(ihp/1.0e3)**2 + 3.288e1*ihp/1.0e3 + 4.999e0);
243     cmacht = 1.3*(3.14e-1*ihp + 1.730e3);
244     ctug = (wsteelt*csteelt + woutfitt*coutfitt + cmacht)*1.0e3;
245     /* calculation of annual operating costs */
246     /* drop and swap opcosts */
247     iopt = 0;
248     dsopt = 0;
249     if rload1 = 0.e0 | runload1 = 0.e0 then
250     if rload1 = 0.e0 then tport1 = 2.4e1*dwt/runload1;
251     else tport1 = 2.4e1*dwt/rload1;
252     else tport1 = (dwt/rload1 + mflowave2*dwt/(mflowave1*runload1))*2.4e1;
253     if rload2 = 0.e0 | runload2 = 0.e0 then
254     if rload2 = 0.e0 then tport2 = 2.4e1*dwt/runload2;
255     else tport2 = 2.4e1*dwt/rload2;
256     else tport2 = (dwt/runload2 + mflowave2*dwt/(mflowave1*rload2))*2.4e1;
257     seatimet = 2*(distance/speed + tlink + tunlink) + delay1 + delay2;
258     dslp: ttript = seatimet/2.4e1;
259     mintug = ceil(max(mflowave1, mflowave2)*ttript/(3.05e1*dwt));
260     /* iterative calculation of ttript, twait1, twait2, and mintug */
261     twait1 = 0.0; twait2 = 0.0;
262     do j = 1 to 50;
263     temp = ttript;
264     if dsopt = 0 then do;
265     minbarg1 = minbrg1; minbarg2 = minbrg2;
266     twait1 = max(0.e0, 5.e-1*(mintug*max(tport1/minbarg1, tport2/minbarg2) - seatimet));
267     twait2 = twait1;
268     end;
269     else do;
270     /* Calculations for D & S mode with unbalanced trade -- no barges at one port */
271     if rload1 + runload1 > rload2 + runload2
272     then do;
273     minbarg1 = 0.e0;
274     twait1 = tport1;
275     twait2 = max(0.e0, mintug*(tport2/minbarg2) - (seatimet + tport1));

```

```

276         end;
277         else do;
278             minbarg2 = 0.e0;
279             twait2 = tport2;
280             twait1 = max(0.e0, mintug*(tport1/minbarg1) -(seatimet + tport2));
281         end;
282     end;
283     ttript = (seatimet + twait1 + twait2)/2.4e1;
284     mintug = ceil(max(mflowave1, mflowave2)*ttript/(3.05e1*dwt));
285     if abs(temp - ttript) < 1.0e-3 then go to next3;
286 end;
287 if iout2 = 0 then do;
288     put list("Rttug does not coverage. Divergence is:",(temp-ttript));
289     put data(dwt, speed, rate, distance);
290     iout2 = 1;
291 end;
292 go to itblp;
293 next3:
294     if(twait1 + twait2) >=(tport1 + tport2) then do;
295         put skip list("Waiting time exceeds porttime. No advantage in drop and swap mode.");
296         put skip data(dwt, speed, rate, distance);
297         go to itblp;
298     end;
299     portimet = twait1 + twait2;
300     go to next4;
301 /* integral tug/barge opcosts */
302 itblp:
303     iopt = 1;
304     seatimet = 2*distance/speed + delay1 + delay2;
305     portimet = tport1 + tport2;
306     ttript = (seatimet + portimet)/2.4e1;
307     minitb = ceil(max(mflowave1, mflowave2)*ttript/(3.05e1*dwt));
308 next4:
309     rseafuel = shp*sfc/2.240e3;
310     rlubeoil = shp*2.5e-4;
311     rportfuel = 1.25e-1;
312     acrew = (cwages + csubs)*nrcrew;
313     fuelcons = rseafuel*seatimet + rportfuel*portimet;
314     lubecons = rlubeoil*seatimet;
315     nrtrips = tugopdays/ttript;
316     tdwt = dwt +(1.0e0 + fuelmargin)*fuelcons + wmisc;
317     aflowcap = nrtrips*dwt;
318     afuel = fuelcons*nrtrips*cfuel;
319     alube = lubecons*nrtrips*clube;
320     amandr = 1.5*(1.288e2 + 4.539e0*ihp/1.0e3 - 4.0e-2*(ihp/1.0e3)**2 + 2.477e0*tdwt/1.0e3
321             - 9.107e-3*(tdwt/1.0e3)**2)*1.0e3;
322     ainsur = 1.5*(2.10e2 + 3.6e-3*ihp + 1.8e-3*tdwt)*1.0e3;
323     asupplies = 1.3*(4.0e1 + 1.8e-3*ihp + 2.6e-4*tdwt)*1.0e3;
324     aport = (cfixport1 + cfixport2 +(cvarport1 + cvarport2)*dwt)*nrtrips;
325     acargo = vcargo*(aflowave1 + aflowave2)*disrate*seatimet/(48.*365.);
326     atermp = (aflowave1 + aflowave2)*cvarterm;
327     atermpcap = ((rload1 + runload1)*minbrg1 + (rload2 + runload2)*minbrg2)*cfixterm;
328     astorop = (aflowave1 + aflowave2)*cvarstor;
329     astorcap = dwt*(minbrg1 + minbrg2)*cfixstor;
330     aopcost = acrew + afuel + alube + aother + amandr + ainsur
331             + asupplies + aport;
332     pvf = (1.0e0 -(1.0e0 + disrate)**-econlife)/disrate;
333 /* compute total annual costs */
334     if mintug ^= 1 then multifctr1 = max(0.8389, 1.035 - 0.0631*mintug + 0.0086.5*mintug**2
335             - 0.0004466*mintug**3);
336     else multifctr1 = 1.;

```



```

389      best1(21) = bbarge;
390      best1(22) = tbarge;
391      best1(23) = displ;
392      best1(24) = portimet/24.;
393      best1(25) = seatimet/24.;
394      best1(26) = ttript;
395      best1(27) = aflowcap;
396      best1(28) = twait1/24.;
397      best1(29) = twait2/24.;
398      best1(30) = minbarge1;
399      best1(31) = minbarge2;
400      best1(32) = cb;
401      best1(33) = lb;
402      best1(34) = bt;
403      best1(35) = ld;
404      if dsopt = 0 then do;
405          dsopt = 1;
406          go to dslp;
407      end;
408      else go to itblp;
409  end;
410  else if(iopt = 0) & dsopt = 0 then do;
411      dsopt = 1;
412      go to dslp;
413  end;
414  else if iopt = 0 then go to itblp;
415  if(iopt = 1) & (rfr < best2(1)) then do;
416      best2(1) = rfr;
417      best2(2) = dwt/1.e3;
418      best2(3) = speed;
419      best2(4) = acost/1.e6;
420      best2(5) = ctug/1.e6;
421      best2(6) = cbarge/1.e6;
422      best2(7) = aopcost;
423      best2(8) = totopcost;
424      best2(9) = ctugbarges;
425      best2(10) = alube;
426      best2(11) = afuel;
427      best2(12) = amandr;
428      best2(13) = ainsur;
429      best2(14) = asupplies;
430      best2(15) = aport;
431      best2(16) = resist;
432      best2(17) = ihp;
433      best2(18) = nrtrips;
434      best2(19) = minitb;
435      best2(20) = lbarge;
436      best2(21) = bbarge;
437      best2(22) = tbarge;
438      best2(23) = displ;
439      best2(24) = portimet/24.;
440      best2(25) = seatimet/24.;
441      best2(26) = ttript;
442      best2(27) = aflowcap;
443      best2(28) = cb;
444      best2(29) = lb;
445      best2(30) = bt;
446      best2(31) = ld;
447  end;
448  end;
449 endbt1p: end;

```



```

450 endlb1p: end;
451 endcol1p: end;
452 endwt1p: end;
453 next5:
454   rfrb1(d, r) = best1(1); dwtb1(d, r) = best1(2); speedb1(d, r) = best1(3); mintugs1(d, r) = best1(19);
455   rfrb2(d, r) = best2(1); dwtb2(d, r) = best2(2); speedb2(d, r) = best2(3); mintugs2(d, r) = best2(19);
456   minbarges1(d, r) = best1(30); percent(d, r) = 100.*best2(24)/best2(26);
457   minbarges2(d, r) = best1(31);
458   if iout = 0 then do;
459     /* output of input parameters in put data format */
460     put skip(2) edit("aflowave1 = ", aflowave1, "minrate = ", minrate, "mindist = ",
461       mindist, "minspeed = ", minspeed, "mindwt = ", mindwt)(r(format1));
462     put skip edit("aflowave2 = ", aflowave2, "maxrate = ", maxrate, "maxdist = ",
463       maxdist, "maxspeed = ", maxspeed, "maxdwt = ", maxdwt)(r(format1));
464     put skip edit("delrate = ", delrate, "deldist = ",
465       deldist, "delspeed = ", delspeed, "deldwt = ", deldwt)
466       (col(25), a, f(6), col(50), a, f(6), col(75), a, f(6, 2), col(100), a, f(8));
467   format1:
468     format(col(1), a, f(8), col(25), a, f(6), col(50), a, f(6), col(75), a, f(6, 2), col(100), a, f(8));
469     put skip(2) edit("max1 = ", max1, "tugopdays = ", tugopdays, "servmargin = ",
470       servmargin, "csteelt = ", csteelt, "csteelb = ", csteelb)(r(format2));
471     put skip edit("maxb = ", maxb, "bargeopdays = ", bargeopdays, "fuelmargin = ",
472       fuelmargin, "couthitt = ", couthitt, "couthitb = ", couthitb)(r(format2));
473     put skip edit("maxt1 = ", maxt1, "tlink = ", tlink, "sfc = ", sfc,
474       "cwages = ", cwages, "cfixport1 = ", cfixport1)(r(format2));
475     put skip edit("maxt2 = ", maxt2, "tunlink = ", tunlink, "cfuel = ", cfuel,
476       "csubs = ", csubs, "cfixport2 = ", cfixport2)(r(format2));
477     put skip edit("ltug = ", ltug, "delay1 = ", delay1, "clube = ", clube,
478       "nrcrew = ", nrcrew, "cvarport1 = ", cvarport1)(r(format2));
479   format2:
480     format(col(1), a, f(8, 2), col(25), a, f(6), col(50), a, f(8, 2), col(75), a, f(8, 2), col(100), a, f(8, 2));
481     put skip edit("delay2 = ", delay2, "vcargo = ", vcargo, "cvarport2 = ", cvarport2)(col(25), a, f(6), col
482       f(8, 2), col(100), a, f(8, 2));
483     put skip(2) edit("aother = ", aother, "disrate = ", disrate, "mincb = ", mincb, "minlb = ", minlb, "minbt = ",
484       "cvarstor = ", cvarstor)(r(format3));
485     put skip edit("admin = ", admin, "econlife = ", econlife, "maxcb = ", maxcb, "maxlb = ", maxlb, "maxbt = ",
486       "cfixstor = ", cfixstor)(r(format3));
487   format3:
488     format(col(1), a, f(10, 2), col(25), a, f(6, 2), col(50), a, f(6, 2), col(67), a, f(6, 1), col(84), a, f(6, 1),
489       col(100), a, f(8, 2));
490     put skip edit("pvf = ", pvf, "delcb = ", delcb, "dellb = ", dellb, "delbt = ", delbt, "cvarterm = ", cva
491       (col(25), a, f(6, 2), col(50), a, f(6, 2), col(67), a, f(6, 1), col(84), a, f(6, 1), col(100), a, f(8, 2));
492     put skip edit("inflafctr = ", inflafctr, "cfixterm = ", cfixterm)(col(25), a, f(6, 2), col(100), a, f(8, 2));
493     if output_type = 1 then
494       put skip(3) edit(" DWT SPD RFR CB BLENGTH L/B B/T L/D ACOST CTUG CBARGE IHP ",
495         "NO TUG/ITB NRTRIPS SEATIMET PORTIMFT AFLOWCAP TWAIT1/%",
496         "(1000)(KTS) ($)", "($M) ($M) ($M) (DAYS) (DAYS) (DAYS)"
497         (a, a, col(1), a, col(49), a, col(97), a);
498     iout = 1;
499   end;
500   if output_type = 1 then do;
501     if inrate = 0 then put skip(2) edit("rate = ", rate)(a, f(8));
502     put edit("distance = ", distance, "minbarga1 = ", best1(30),
503       "minbarga2 = ", best1(31))(col(1), 3(x(2), a, f(8)));
504     put skip edit(best1(2), best1(3), best1(1), best1(32), best1(20), best1(33), best1(34), best1(35), best1(4), best
505       best1(6), best1(17), best1(19), best1(18), best1(25), best1(24), best1(27), best1(29))

```

```

506      (x(1), f(3), x(2), f(4, 1), x(1), f(6, 2), x(1), f(4, 2), x(1), f(5, 1), x(3), f(4, 2), x(1), f(4, 2), x(1), f(
\c4, 1),
507      3(x(1), f(7, 3)), x(1), f(5), x(4), f(2), x(6), f(6, 2), x(2), f(6, 2), x(2), f(7, 2), x(2), f(7), x(2), f(6,
\c2));
508      put skip edit(best2(2), best2(3), best2(1), best2(28), best2(20), best2(29), best2(30), best2(31), best2(4), best
\c2(5),
509      best2(6), best2(17), best2(19), best2(18), best2(25), best2(24), best2(27), 100*best2(24)/best2(26))
510      (x(1), f(3), x(2), f(4, 1), x(1), f(6, 2), x(1), f(4, 2), x(1), f(5, 1), x(3), f(4, 2), x(1), f(4, 2), x(1), f(
\c4, 1),
511      3(x(1), f(7, 3)), x(1), f(5), x(4), f(2), x(6), f(6, 2), x(2), f(6, 2), x(2), f(7, 2), x(2), f(7), x(2), f(3))
\c;
512      end;
513      end; end;
514      put skip edit("Do you wish graphic output?: ")(col(1), a);
515      call yesno;
516      if ibit = 0 then go to st;
517      put skip edit("Do you wish to plot rfr vs. rate while varying distance?: ")
518      (col(1), a);
519      call yesno;
520      if ibit = 1 then call plots("REQUIRED FREIGHT RATE VS LOAD/DISCH RATE WHILE VARYING DISTANCE",
521      "TONS PER DAY", "RFR-$", rfrb1, rfrb2, 0);
522      put skip edit("Do you wish to plot rfr vs. distance while varying rate?: ")
523      (col(1), a);
524      call yesno;
525      if ibit = 1 then call plots("REQUIRED FREIGHT RATE VS DISTANCE WHILE VARYING L-D RATE",
526      "MILES", "RFR-$", rfrb1, rfrb2, 1);
527      put skip edit("Do you wish to plot barge DWT vs. L/D rate while varying distance?: ")
528      (col(1), a);
529      call yesno;
530      if ibit = 1 then call plots("BARGE DWT VS LOAD/DISCHARGE RATE WHILE VARYING DISTANCE",
531      "TONS PER DAY", "DWT", dwtb1, dwtb2, 0);
532      put skip edit("Do you wish to plot barge DWT vs. distance while varying L/D rate?: ")
533      (col(1), a);
534      call yesno;
535      if ibit = 1 then call plots("BARGE DWT VS DISTANCE WHILE VARYING L/D RATE",
536      "MILES", "DWT", dwtb1, dwtb2, 1);
537      put skip edit("Do you wish to plot tugspeed vs. L/D rate while varying distance?: ")
538      (col(1), a);
539      call yesno;
540      if ibit = 1 then call plots("TUG SPEED VS LOAD/DISCHARGE RATE WHILE VARYING DISTANCE",
541      "TONS PER DAY", "KTS", speedb1, speedb2, 0);
542      put skip edit("Do you wish to plot tugspeed vs. distance while varying L/D rate?: ")
543      (col(1), a);
544      call yesno;
545      if ibit = 1 then call plots("TUG SPEED VS DISTANCE WHILE VARYING L/D RATE",
546      "MILES", "KTS", speedb1, speedb2, 1);
547      put skip edit("Do you wish to plot the minimum no. of required tugs vs L/D rate while varying Distance?: ")
548      (col(1), a);
549      call yesno;
550      if ibit = 1 then call plots("MIN. NO. OF TUGS REQUIRED VS L/D RATE WHILE VARYING DISTANCE",
551      "TONS PER DAY", "NO. ; TUGS", mintugs1, mintugs2, 0);
552      put skip edit("Do you wish to plot the minimum no. of required tugs vs distance while varying L/D rate?: ")
553      (col(1), a);
554      call yesno;
555      if ibit = 1 then call plots("MIN. NO. OF TUGS REQUIRED VS DISTANCE WHILE VARYING L/D RATE",
556      "MILES", "NO. ; TUGS", mintugs1, mintugs2, 1);
557      put skip edit("Do you wish to plot no. of barges required at port 1 and port 2 under D&S ",
558      "vs L/D rate?: ")(col(1), a, a);
559      call yesno;
560      if ibit = 1 then call plots("MIN NO. BARGES REQUIRED AT PORT 1 AND PORT 2 VS L/D RATE ",
561      "TONS PER DAY", "NO. ; BARGES", minbarges1, minbarges2, 0);

```

```

562 put skip edit("Do you wish to plot portime % vs. L/D rate while varying distance?: ") (col(1), a);
563 call yesno;
564 if ibit = 1 then call plots("PORTIME % VS LOAD/DISCHARGE RATE WHILE VARYING DISTANCE",
565   "TONS PER DAY", "% PORTIME/TRIPTIME", percent, percent, 0);
566 go to st;
567 yesno: proc;
568   do j = 1 to 8;
569     get edit(nextchar)(a(1));
570     if nextchar = "Y" | nextchar = "y" | nextchar = "1" then do;
571       ibit = 1;
572       get skip;
573       return;
574     end;
575     if nextchar = "N" | nextchar = "n" | nextchar = "0" then do;
576       ibit = 0;
577       get skip;
578       return;
579     end;
580     ibit = 0;
581     get skip;
582     return;
583   end;
584 end yesno;
585 plots: proc(blurb1, blurb2, blurb3, o1, o2, graph_type);
586   del(blurb1, blurb2, blurb3) char(*);
587   del(o1(*, *), o2(*, *)) float bin;
588   del(x, y1, y2)(250) float bin;
589   del(xmin, xmax, ymin, ymax) float bin;
590   del graph_type fixed bin;
591   stgr:
592     call plot_$setup(blurb1, blurb2, blurb3, 1, 0e0, 1, 0);
593     put skip edit("Do you wish to specify graph scales?: ") (col(1), a);
594     call yesno;
595     if ibit = 1 then do;
596       xmin = 0.e0; xmax = 0.e0; ymin = 0.e0; ymax = 0.e0;
597       put skip edit("Specify xmin, xmax, ymin, ymax via get data format: ")
598         (col(1), a);
599       get skip data(xmin, xmax, ymin, ymax);
600       call plot_$scale(xmin, xmax, ymin, ymax);
601     end;
602     if graph_type = 0 then do;
603       do d = 1 to drange;
604         do r = 1 to rrange;
605           x(r) = minrate +(r - 1)*delrate;
606           y1(r) = o1(d, r);
607           y2(r) = o2(d, r);
608         end;
609         call plot_(x, y1, rrange, 2, "");
610         call plot_(x, y2, rrange, 2, "+");
611       end; end;
612     if graph_type = 1 then do;
613       do r = 1 to rrange;
614         do d = 1 to drange;
615           x(d) = mindist +(d - 1)*deldist;
616           y1(d) = o1(d, r);
617           y2(d) = o2(d, r);
618         end;
619         call plot_(x, y1, drange, 2, "");
620         call plot_(x, y2, drange, 2, "+");
621       end; end;

```

```
622 put skip edit("Do you wish to revise plot?: ")(col(1), a);
623 call yesno;
624 if ibit = 1 then go to stgr;
625 return;
626 end plots;
627 end drop_and_swap;
```


clube	033060	automatic	float bin(27)	dcl 4 set ref 107 115 319 477
cmacht	033050	automatic	float bin(27)	dcl 4 set ref 243 244
count	036745	automatic	fixed bin(17,0)	dcl 25 set ref 204 205 205 205 205
coutfitb	036651	automatic	float bin(27)	dcl 22 set ref 107 115 201 471
coutfitt	033047	automatic	float bin(27)	dcl 4 set ref 107 115 244 471
csteelb	036650	automatic	float bin(27)	dcl 22 set ref 107 115 201 469
csteelt	033046	automatic	float bin(27)	dcl 4 set ref 107 115 244 469
csubs	033053	automatic	float bin(27)	dcl 4 set ref 107 115 312 475
ctug	033054	automatic	float bin(27)	dcl 4 set ref 244 343 348 359 363 373 420
ctugbarges	033056	automatic	float bin(27)	dcl 4 set ref 343 344 348 349 377 424
cvarport1	033063	automatic	float bin(27)	dcl 4 set ref 107 115 324 477
cvarport2	033064	automatic	float bin(27)	dcl 4 set ref 107 115 324 481
cvarstor	033065	automatic	float bin(27)	dcl 4 set ref 107 115 328 483
cvarterm	033067	automatic	float bin(27)	dcl 4 set ref 107 115 326 490
cwages	033052	automatic	float bin(27)	dcl 4 set ref 107 115 312 473
c	036743	automatic	fixed bin(17,0)	dcl 25 set ref 136 453 454 454 454 455 455 455 455 456 456 457 603 606 607 614 615 615 616 616 617 617
dtarge	033003	automatic	float bin(27)	dcl 4 set ref 215 216 217
delay1	033012	automatic	float bin(27)	dcl 4 set ref 107 115 257 304 477
delay2	033013	automatic	float bin(27)	dcl 4 set ref 107 115 257 304 481
delbt	036702	automatic	float dec(10)	dcl 23 set ref 92 93 93 176 490
delcb	036660	automatic	float dec(10)	dcl 23 set ref 86 87 87 170 490
delcist	036735	automatic	fixed bin(17,0)	dcl 25 set ref 64 65 65 131 132 135 136 464 615
delcwt	036731	automatic	fixed bin(17,0)	dcl 25 set ref 70 71 71 146 165 464
delib	036671	automatic	float dec(10)	dcl 23 set ref 89 90 90 175 490
delrate	036740	automatic	fixed bin(17,0)	dcl 25 set ref 56 61 62 62 133 134 137 138 464 605
delsspeed	036726	automatic	fixed bin(17,0)	dcl 25 set ref 67 68 68 229 464
disp	036635	automatic	float bin(27)	dcl 21 set ref 231 237
displ	033005	automatic	float bin(27)	dcl 4 set ref 190 191 192 216 391 438
disrate	033107	automatic	float bin(27)	dcl 4 set ref 107 115 325 332 332 483
distance	032766	automatic	float bin(27)	dcl 4 set ref 135 136 257 289 296 304 502
dis	033546	automatic	float bin(27)	array dcl 20 set ref 44 231
disrange	036741	automatic	fixed bin(17,0)	dcl 25 set ref 131 132 603 614 619 620
disrpt	036761	automatic	fixed bin(17,0)	dcl 27 set ref 248 264 354 404 405 410 411
dt	033535	automatic	float bin(27)	array dcl 20 set ref 44 231
dwt	032770	automatic	float bin(27)	dcl 4 set ref 165 177 178 179 181 182 183 190 197 249 251 252 252 253 255 256 256 259 284 289 296 307 316 317 324 329 354 359 363 370 417
dwtb1	005004	automatic	float bin(27)	array dcl 2 set ref 454 530 535
dwtb2	007346	automatic	float bin(27)	array dcl 2 set ref 455 530 535
econlife	033113	automatic	float bin(27)	dcl 4 set ref 107 115 332 485
erp	036642	automatic	float bin(27)	dcl 21 set ref 231 235
eta	036265	automatic	float bin(27)	array dcl 20 set ref 42 231
fed	036721	automatic	float bin(27)	dcl 24 set ref 214 215
fuelcons	033035	automatic	float bin(27)	dcl 4 set ref 313 316 318
fuelmargin	033027	automatic	float bin(27)	dcl 4 set ref 107 115 316 471
graph_type		parameter	fixed bin(17,0)	dcl 590 ref 585 602 612
hr	036640	automatic	float bin(27)	dcl 21 set ref 231
nullwt75	036643	automatic	float bin(27)	dcl 22 set ref 181 184
nullwt80	036644	automatic	float bin(27)	dcl 22 set ref 182 184
nullwt85	036645	automatic	float bin(27)	dcl 22 set ref 183 184
i	036747	automatic	fixed bin(17,0)	dcl 27 set ref 39 39 41 41 42 42 43 43
ibit	006010	internal static	fixed bin(17,0)	dcl 33 set ref 52 123 127 516 520 525 530 535 540 545 550 555 560 564 571 576 580 595 624
inp	033033	automatic	float bin(27)	dcl 4 set ref 238 240 240 241 241 242 242 243 320 320 322 323 359 363 363 432
index	036762	automatic	fixed bin(17,0)	dcl 27 set ref 231 232
inflafctr	033111	automatic	float bin(27)	dcl 4 set ref 107 115 352 492

inrate	036760	automatic	fixed bin(17,0)	dcl 27 set ref 52 53 139 501
iopt	036746	automatic	fixed bin(17,0)	dcl 27 set ref 247 302 342 359 363 368 410 414 415
iopt3	036756	automatic	fixed bin(17,0)	dcl 27 set ref 127 354 359 363
iout	036753	automatic	fixed bin(17,0)	dcl 27 set ref 130 458 498
iout2	036754	automatic	fixed bin(17,0)	dcl 27 set ref 145 287 290
iout3	036755	automatic	fixed bin(17,0)	dcl 27 set ref 98 101 102
j	036750	automatic	fixed bin(17,0)	dcl 27 set ref 39 39 41 41 42 42 43 43 185 262 568
k	036751	automatic	fixed bin(17,0)	dcl 27 set ref 39 39
lb	036705	automatic	float dec(10)	dcl 23 set ref 175 177 178 179 181 182 183 186 359 363 401 444
lbarge	033000	automatic	float bin(27)	dcl 4 set ref 180 186 188 189 191 191 195 197 202 204 205 208 209 209 209 210 213 216 217 218 359 363 388 435
lbarge75	036626	automatic	float bin(27)	dcl 21 set ref 177 180
lbarge80	036627	automatic	float bin(27)	dcl 21 set ref 178 180
lbarge85	036630	automatic	float bin(27)	dcl 21 set ref 179 180
ld	036722	automatic	float bin(27)	dcl 24 set ref 217 219 359 363 403 446
ldfact	036717	automatic	float bin(27)	dcl 24 set ref 209 210 212 214
litb	033001	automatic	float bin(27)	dcl 4 set ref 218 219 223 223 223 223 231
lloef	033305	automatic	float bin(27)	array dcl 20 set ref 36 202 205 205 205
lifact	000024	constant	file	input stream dcl 34 set ref 34 35 36 37
ltug	032777	automatic	float bin(27)	dcl 4 set ref 107 115 186 218 477
luccons	033036	automatic	float bin(27)	dcl 4 set ref 314 319
m	036752	automatic	fixed bin(17,0)	dcl 27 set ref 39 39
maxb	032773	automatic	float bin(27)	dcl 4 set ref 107 115 219 471
maxbt	036677	automatic	float dec(10)	dcl 23 set ref 92 94 94 176 485
maxcb	036655	automatic	float dec(10)	dcl 23 set ref 86 94 94 170 485
maxdist	036734	automatic	fixed bin(17,0)	dcl 25 set ref 64 131 132 135 462
maxdwt	036730	automatic	fixed bin(17,0)	dcl 25 set ref 70 165 462
maxl	032772	automatic	float bin(27)	dcl 4 set ref 107 115 219 469
maxlb	036666	automatic	float dec(10)	dcl 23 set ref 89 94 175 485
maxrate	036737	automatic	fixed bin(17,0)	dcl 25 set ref 57 61 133 134 137 462
maxspeed	036725	automatic	fixed bin(17,0)	dcl 25 set ref 67 229 462
maxt1	032774	automatic	float bin(27)	dcl 4 set ref 107 115 219 473
maxt2	032775	automatic	float bin(27)	dcl 4 set ref 107 115 219 475
mflowave1	033016	automatic	float bin(27)	dcl 4 set ref 153 157 158 159 162 252 256 259 284 307
mflowave2	033017	automatic	float bin(27)	dcl 4 set ref 154 155 158 161 162 252 256 259 284 307
minbarge1	033071	automatic	float bin(27)	dcl 4 set ref 155 157 158 163 265 266 273 280 337 337 337 337 343 345 398
minbarge2	033072	automatic	float bin(27)	dcl 4 set ref 159 161 162 163 265 266 275 278 337 337 337 337 343 345 399
minbarges1	023520	automatic	float bin(27)	array dcl 2 set ref 456 560
minbarges2	026062	automatic	float bin(27)	array dcl 2 set ref 457 560
minbrg1	033073	automatic	float bin(27)	dcl 4 set ref 163 265 327 329
minbrg2	033074	automatic	float bin(27)	dcl 4 set ref 163 265 327 329
minbt	036674	automatic	float dec(10)	dcl 23 set ref 92 94 94 176 483
mincb	036652	automatic	float dec(10)	dcl 23 set ref 86 94 94 170 483
mindist	036733	automatic	fixed bin(17,0)	dcl 25 set ref 64 131 132 135 136 460 615
mindwt	036727	automatic	fixed bin(17,0)	dcl 25 set ref 70 72 146 148 165 354 460
mindwttemp	036732	automatic	fixed bin(17,0)	dcl 25 set ref 72 146 148
minfbd	036716	automatic	float bin(27)	dcl 24 set ref 202 205 207 207 208 209 210 214
minitb	033076	automatic	float bin(27)	dcl 4 set ref 307 339 339 339 339 348 350 363 434
minlb	036663	automatic	float dec(10)	dcl 23 set ref 89 94 175 483
minrate	036736	automatic	fixed bin(17,0)	dcl 25 set ref 57 61 133 34 137 138 146 460 605
minspeed	036724	automatic	fixed bin(17,0)	dcl 25 set ref 67 229 354 460

mintug	033075	automatic	float bin(27)	dcl 4 set ref 259 266 275 280 284 334 334 334 334 337 337 337 343 345 359 387 array dcl 2 set ref 454 550 555 array dcl 2 set ref 455 550 555 dcl 4 set ref 334 336 343 dcl 4 set ref 337 343 dcl 4 set ref 339 341 348 unaligned dcl 28 set ref 569 570 570 570 575 575 575
mintugs1	016614	automatic	float bin(27)	dcl 4 set ref 107 115 312 477
mintugs2	021156	automatic	float bin(27)	dcl 4 set ref 315 317 318 319 324 359 363 386 433
multifctr1	033037	automatic	float bin(27)	array dcl 587 ref 585 606 616 array dcl 587 ref 585 607 617 dcl 27 set ref 123 124 493 500 input stream dcl 34 set ref 34 106 107 113 array dcl 2 set ref 456 564 564 external dcl 1-3 ref 609 610 619 620 external dcl 1-3 ref 600 external dcl 1-3 ref 591 dcl 4 set ref 299 305 306 313 359 363 363 392 439
multifctr2	033040	automatic	float bin(27)	external dcl 29 ref 231 input stream dcl 34 set ref 34 38 39 41 42 43 44 45
multifctr3	033041	automatic	float bin(27)	dcl 21 set ref 231 dcl 4 set ref 332 352 490
nextchar	036763	automatic	char(1)	dcl 25 set ref 138 453 454 454 454 455 455 455 455 456 456 457 604 605 605 606 606 607 607 613 616 617 dcl 4 set ref 137 138 140 141 142 143 146 289 296 501
nrnew	033051	automatic	float bin(27)	array dcl 20 set ref 39 231 dcl 4 set ref 235 384 431
nrtrips	033115	automatic	float bin(27)	dcl 4 set ref 353 359 363 368 369 415 416 array dcl 2 set ref 453 520 525 array dcl 2 set ref 455 520 525 dcl 4 set ref 55 76 140 155 155 157 158 249 249 251 252 271 327 dcl 4 set ref 55 80 141 159 159 161 162 253 253 255 256 271 327 dcl 4 set ref 310 314 dcl 4 set ref 311 313 dcl 25 set ref 133 134 604 609 610 613 dcl 4 set ref 308 313 dcl 4 set ref 55 81 142 155 155 158 249 249 252 271 327 dcl 4 set ref 55 77 143 159 159 162 253 253 256 271 327 dcl 4 set ref 257 258 266 275 280 283 304 306 313 314 325 359 363 393 440 dcl 4 set ref 107 115 238 469 dcl 4 set ref 107 115 308 473 dcl 4 set ref 237 238 308 310 dcl 24 set ref 213 214 dcl 4 set ref 229 231 235 257 289 296 304 354 359 363 371 418 array dcl 2 set ref 454 540 545 array dcl 2 set ref 455 545 545 input stream dcl 34 set ref 34 55 61 64 67 70
o1		parameter	float bin(27)	
o2		parameter	float bin(27)	
output_type	036757	automatic	fixed bin(17,0)	
parameters	000022	constant	file	
percent	030424	automatic	float bin(27)	
plot	000034	constant	entry	
plot_scale	000040	constant	entry	
plot_setup	000036	constant	entry	
portimet	033105	automatic	float bin(27)	
power	000020	constant	entry	
powerdata	000026	constant	file	
propel	036641	automatic	float bin(27)	
pvf	033110	automatic	float bin(27)	
r	036744	automatic	fixed bin(17,0)	
rate	032776	automatic	float bin(27)	
re	033563	automatic	float bin(27)	
resist	033034	automatic	float bin(27)	
rfr	033114	automatic	float bin(27)	
rfr1	000100	automatic	float bin(27)	
rfr2	002442	automatic	float bin(27)	
rioad1	033006	automatic	float bin(27)	
rioad2	033007	automatic	float bin(27)	
riosecil	033102	automatic	float bin(27)	
rportfuel	033103	automatic	float bin(27)	
rrange	036742	automatic	fixed bin(17,0)	
rseafuel	033101	automatic	float bin(27)	
runload1	033010	automatic	float bin(27)	
runload2	033011	automatic	float bin(27)	
seatimet	033104	automatic	float bin(27)	
servmargin	033026	automatic	float bin(27)	
sfc	033030	automatic	float bin(27)	
snp	033032	automatic	float bin(27)	
snrfact	036720	automatic	float bin(27)	
speed	032767	automatic	float bin(27)	
speedb1	011710	automatic	float bin(27)	
speedb2	014252	automatic	float bin(27)	
sysin	000030	constant	file	

sysprint	000032 constant	file
t	035363 automatic	float bin(27)
tboarge	033004 automatic	float bin(27)
tbot	032771 automatic	float bin(27)
temp	033031 automatic	float bin(27)
tn	036637 automatic	float bin(27)
tlink	033022 automatic	float bin(27)
toteapcost	033303 automatic	float bin(27)
totepecost	033302 automatic	float bin(27)
tport1	033077 automatic	float bin(27)
tport2	033100 automatic	float bin(27)
ttript	033106 automatic	float bin(27)
tugopdays	033020 automatic	float bin(27)
tunlink	033023 automatic	float bin(27)
twait1	033024 automatic	float bin(27)
twait2	033025 automatic	float bin(27)
vcargo	033112 automatic	float bin(27)
vclwt	036723 automatic	float bin(27)
x	035724 automatic	float bin(27)
xa	036636 automatic	float bin(27)
xmach	033044 automatic	float bin(27)
xmiso	033045 automatic	float bin(27)
xoutfitb	036647 automatic	float bin(27)
xoutfitt	033043 automatic	float bin(27)
xsteelb	036646 automatic	float bin(27)
xsteelt	033042 automatic	float bin(27)
x	000100 automatic	float bin(27)
x1cb	036632 automatic	float bin(27)
x2cb	036633 automatic	float bin(27)
x3cb	036634 automatic	float bin(27)
xcb	036631 automatic	float bin(27)
xmax	001457 automatic	float bin(27)
xmin	001456 automatic	float bin(27)
y	000472 automatic	float bin(27)
y2	001064 automatic	float bin(27)
ymax	001461 automatic	float bin(27)
ymin	001460 automatic	float bin(27)

74 86 89 92 115 569 572 577 581 599
output stream dcl 34 set ref 34 48 49 54 60 63
66 69 73 84 88 91 96 97 114 121 125 194 197 288
289 295 296 354 359 363 460 462 464 469 471 473
475 477 481 483 485 490 492 493 501 502 504 508
514 517 522 527 532 537 542 547 552 557 562 593
597 622
array dcl 20 set ref 43 231
dcl 4 set ref 187 188 208 209 210 215 216 219
219 231 390 437
dcl 4 set ref 316 320 320 322 323
dcl 4 set ref 188 192 263 285 288
dcl 21 set ref 231
dcl 4 set ref 107 115 257 473
dcl 4 set ref 344 349 352
dcl 4 set ref 345 350 352 376 423
dcl 4 set ref 249 251 252 266 274 275 280 293
305
dcl 4 set ref 253 255 256 266 275 279 280 293
305
dcl 4 set ref 258 259 263 283 284 285 288 306
307 315 363 394 441
dcl 4 set ref 107 115 315 469
dcl 4 set ref 107 115 257 475
dcl 4 set ref 261 266 267 274 280 283 293 299
359 396
dcl 4 set ref 261 267 275 279 283 293 299 397
dcl 4 set ref 107 115 325 481
dcl 24 set ref 216
array dcl 20 set ref 41 231
dcl 21 set ref 231
dcl 4 set ref 242
dcl 4 set ref 107 115 316
dcl 22 set ref 189 190 201 216
dcl 4 set ref 241 244
dcl 22 set ref 184 190 201 216
dcl 4 set ref 240 244
array dcl 588 set ref 605 609 610 615 619 620
dcl 21 set ref 172 180 184
dcl 21 set ref 173 180 184
dcl 21 set ref 174 180 184
dcl 21 set ref 171 172 172 173 173 174 174
dcl 589 set ref 596 599 600
dcl 589 set ref 596 599 600
array dcl 588 set ref 606 609 616 619
array dcl 588 set ref 607 610 617 620
dcl 589 set ref 596 599 600
dcl 589 set ref 596 599 600

NAMES DECLARED BY EXPLICIT CONTEXT.

drop_and_swap	003434	constant	entry	external dcl 1 ref 1
dsip	010255	constant	label	dcl 258 ref 258 406 412
dwtip	005732	constant	label	dcl 165 ref 165
endttip	013420	constant	label	dcl 449 ref 219 233 449
endtblp	013430	constant	label	dcl 451 ref 199 451
endtblp	013424	constant	label	dcl 450 ref 227 450
endwtip	013434	constant	label	dcl 452 ref 452
format1	003361	constant	format	ref 460 462
format2	003304	constant	format	ref 469 471 473 475 477
format3	003215	constant	format	ref 483 485
itclp	010574	constant	label	dcl 302 ref 292 297 302 404 414
llcsp	007322	constant	label	dcl 195 ref 192 195
next3	010513	constant	label	dcl 293 ref 285 293
next4	010625	constant	label	dcl 308 ref 300 308
next5	013441	constant	label	dcl 453 ref 453
plots	022002	constant	entry	internal dcl 585 ref 520 525 530 535 540 545 550 555 560 564 585
sp	004320	constant	label	dcl 63 ref 58 63
st	004147	constant	label	dcl 48 ref 48 516 566
st2	004532	constant	label	dcl 84 ref 84 99
st3	005331	constant	label	dcl 130 ref 103 130
stgr	022040	constant	label	dcl 591 ref 591 624
yesno	021627	constant	entry	internal dcl 567 ref 51 122 126 515 519 524 529 534 539 544 549 554 559 563 567 594 623

NAMES DECLARED BY CONTEXT OR IMPLICATION.

acos	builtin function	internal ref 192 285
ceil	builtin function	internal ref 155 157 158 159 161 162 204 259 284 307
exp	builtin function	internal ref 177 178 179 181 182 183
max	builtin function	internal ref 146 189 259 266 266 275 280 284 307 334 337 339
min	builtin function	internal ref 146

STORAGE REQUIREMENTS FOR THIS PROGRAM.

	Object	Text	Link	Symbol	Defs	Static
Start	0	0	23132	23206	22726	23142
Length	24410	22726	54	1166	203	6

BLOCK NAME	STACK SIZE	TYPE	WHY NONQUICK/WHO SHARES STACK FRAME
drop_and_swap	16496	external procedure	is an external procedure.
yesno	236	internal procedure	contains a format statement, and uses I/O statements.
plots	1030	internal procedure	contains a format statement, and uses I/O statements.

THE FOLLOWING EXTERNAL OPERATORS ARE USED BY THIS PROGRAM.

fx1 to fl2	r_l a	r_g a	r_e as	r_ge a	call_ext_out_desc
call_ext_out	call_int_this_desc	call_int_this	call_int_other	return	fl2_to_fx1
ext_entry	int_entry	int_entry_desc	get_end	put_end	stream_io
open	close	trunc_fx2	ceil fl	put_data_eis	put_list_eis
get_edit_eis	get_list_eis	real_to_real_rd	any_to_any_rd	divide_fx1	exp
dcl_p_dcl	real_p_real	put_field	put_field_chk	put_control	

A.3 DICTIONARY OF VARIABLES USED IN THE DROP-AND-SWAP COMPUTER MODEL

DEFINITION OF VARIABLES IN THE PROGRAM "drop-and-swap"

<u>Variable</u>	<u>Definition</u> (units)
acargo:	Annual Time Value for Cargo Cost. Total annual cost for the time value of the cargoes onboard the barge while being transported. The product of the annual discount rate times the seetime times the cargo value. (\$)
acost:	Annual Cost. Total annual cost for either drop-and-swap or integral OGTB system, including annual operating costs and annualized present value share of capital costs. (\$)
acrew:	Annual Crew Cost. Annual costs for crew, including wages and subsistence, for a tug. (\$/tug)
admin:	Annual Administrative Costs. Annual costs for administration of tug. (\$/tug)
aflowave1:	Annual Average Cargo Flow from Port 1. Average annual flow of cargo to be loaded at Port 1 and discharged at Port 2. (cargo units)
aflowave2:	Annual Average Cargo Flow from Port 2. Average annual flow of cargo to be loaded at Port 2 and discharged at Port 1. (cargo units)
aflowcap:	Annual Flow Capacity. Annual amount of cargo capacity provided by each tug, product of barge DWT and number of tug voyages. (cargo units)
afuel:	Annual Fuel Cost. Annual cost for fuel for a tug; product of the number of tug voyages, fuel consumption per voyage, and unit cost of fuel. (\$/tug)
ainsur:	Annual Insurance Premiums. Annual cost for insurance for a tug-barge unit. (\$/tug-barge)

alube: Annual Lube Oil Costs. Annual cost for lube oil for a tug; product of the number of tug voyages, lube oil consumption per voyage and unit cost of lube oil. (\$/tug)

amandr: Annual M&R Costs. Annual cost for maintenance and repair of a tug-barge unit. (\$/tug-barge)

aopcost: Annual Operating Cost. Annual costs for operating a tug-barge unit. (\$/tug-barge)

aother: Annual Other Costs. Other annual miscellaneous operating costs not included in admin, afuel, alube, ainsur, amandr, acrew, asupplies, and aporti. (\$/tug)

aport: Annual Port Charges. Annual costs for port charges for each tug, including variable and fixed charges for Port 1 and Port 2. (\$/tug)

astorcap: Annual Capital Cost for Storage Facilities. The cost for shoreside storage facilities required for the integral mode of operation. The product of the number of loading/discharging facilities, the barge size, and a cost factor ("cfixstor"). (\$)

astorop: Annual Operating Costs for Storage Facilities. The annual operating cost incurred for the operation or use of shoreside storage facilities required for the integral mode of operation. The product of the annual cargo flows and a cost factor ("cvarstor"). (\$/yr)

asupplies: Annual Supplies Cost. Annual costs for stores, supplies, and equipment for each tug-barge unit. (\$/tug-barge)

atermcap: Total Capital Cost for Loading/Discharging Facilities. The capital cost for shoreside terminal facilities. The product of the terminal facilities throughput rate and a cost factor ("cfixterm"). (\$)

atermop: Annual Operating Cost for Loading/Discharging Facilities. The annual operating costs incurred for the operation or use of the shoreside terminal facilities. The product of the annual cargo flows and a cost factor ("cvarterm"). (\$/yr)

bargeopdays: Barge Operating Days. Number of days during the year which a barge is expected to be available for operations. (days)

bbarge: Barge Beam. (ft)

best1: Storage Matrix that saves the best values found for various drop-and-swap mode variables (specified below) for a specified port separation distance and port loading and discharge rate while iterating over tug speed, barge DWT, and barge form.

best2: Same as "best1" except for integral mode.

best1(1),	best2(1)	- rfr
best1(2),	best2(2)	- dwt/1000
best1(3),	best2(3)	- speed
best1(4),	best2(4)	- acost/1000000
best1(5),	best2(5)	- ctug/1000000
best1(6),	best2(6)	- cbarge/1000000
best1(7),	best2(7)	- aopcost
best1(8),	best2(8)	- totopcost
best1(9),	best2(9)	- ctugbarges
best1(10),	best2(10)	- alube
best1(11),	best2(11)	- afuel
best1(12),	best2(12)	- amandr
best1(13),	best2(13)	- ainsur
best1(14),	best2(14)	- asupplies
best1(15),	best2(15)	- aprot
best1(16),	best2(16)	- resist
best1(17),	best2(17)	- ihp
best1(18),	best2(18)	- nrtrips
best1(19),	best2(19)	- mintug, minitb
best1(20),	best2(20)	- lbarge
best1(21),	best2(21)	- bbarge
best1(22),	best2(22)	- tbarge
best1(23),	best2(23)	- displ
best1(24),	best2(24)	- portimet/2 ⁴
best1(25),	best2(25)	- seatimet/2 ⁴
best1(26),	best2(26)	- ttript
best1(27),	best2(27)	- aflowcap
best1(28)	twait1	
	best2(28)	- cb
best1(29)	twait2	
	best2(29)	- lb
best1(30)	minbarge1	
	best2(30)	- bt
best1(31)	minbarge2	
	best2(31)	- ld
best1(32)	cb	

best1(33)	lb
best1(34)	bt
best1(35)	ld

blurb1,2,3: Subroutine Plots input parameters representing title of graph, vertical axis, and horizontal axis.

bt: Breadth-Draft Ratio. The ratio of the tug-barge unit's breadth and draft. (ft)

cbarge: Cost of Barge. (\$)

c1,c2: Coefficient arrays used only as input parameters in the calling of subprogram "power"; values are used in subroutine "prop" of subprogram "power."

cb: Block Coefficient. The block coefficient of the tug-barge unit.

cf: Coefficient of Frictional Resistance for OGTB.

cfixport1: Fixed Port Charges for Port 1. Cost of fixed port charges of tug/barge at Port 1 per voyage. (\$/voyage)

cfixport2: Fixed Port Charges for Port 2. Cost of fixed port charges of tug/barge at Port 2 per voyage. (\$/voyage)

cfixstor: Storage Facility Capital Cost Factor. The cost per long ton for shoreside storage facilities used in the integral mode. (\$/LT)

cfixterm: Terminal Facility Capital Cost Factor. The cost per long ton per day throughput rate for terminal facilities. (\$/LT-day)

cfuel: Cost of Fuel. Diesel fuel cost per ton. (\$/LT)

clube: Cost of Lube Oil. Lube oil cost per gallon. (\$/Gal)

cmacht: Tug Machinery Cost. Cost of propulsion machinery onboard the tug; function of IHP. (\$1000/tug)

countr: Index for the loadline coefficient array(11coef) used in the calculation of the variable "minfbd".

countfitb: Barge Outfit Cost. Cost of a ton of outfit material onboard the barge. (\$1000/LT)

coutfitt: Tug Outfit Cost. Cost of a ton of outfit onboard the tug. (\$1000/LT)

csteelb: Barge Steel Cost. Cost of a ton of hull steel onboard the barge. (\$1000/LT)

csteelt: Tug Steel Cost. Cost of a ton of hull steel onboard the tug. (\$1000/LT)

csubs: Cost of Subsistence. Annual subsistence cost per crew member. (\$/man)

ctug: Cost of Tug. Initial capital cost of a tug as a function of steel and outfit weight and machinery cost. (\$/tug)

ctugbarges: Cost of Tug and Barges. Total capital cost of all tugs and barges in the system. (\$)

cvarport1: Variable Port Charges for Port 1. Cost per cargo unit capacity for port charges at Port 1 per voyage. (\$/cargo unit and voyage)

cvarport2: Variable Port Charges for Port 2. Cost per cargo unit capacity for port charges at Port 2 per voyage. (\$/cargo unit and voyage)

cvarstor: Storage Facility Operating Cost Factor. The cost per long ton stored in the integral mode. (\$/LT)

cvarterm: Terminal Facility Operating Cost Factor. The cost per long ton moved through a terminal facility. (\$/LT)

cwage: Cost of Wages. Annual benefit and wage cost per crew member. (\$/man)

d: Distance Index. Index used to represent the variable distance in optimum system arrays-rfrb1, rfrb2, dwtb1,...mintugs.

dbarge: Depth of Barge. (ft)

delay1: Port 1 Delay Time. Length of delay at Port 1 prior to and after cargo operations, including docking/undocking and awaiting berth time for the barge. (hrs)

delay2: Port 2 Delay Time. Same as "delay1" except for Port 2.

delbt: Breadth-Depth Ratio Increment. The incremental B/T to be used while varying the tug-barge breadth-depth ratio from its minimum to its maximum value.

delcb: Block Coefficient Increment. The incremental Cb to be used while varying the tug-barge block coefficient from its minimum to its maximum value.

deldist: Distance Increment. The incremental distance to be used while varying the port separation distance from its minimum to its maximum value. (nautical miles)

deldwt: Deadweight Increment. The incremental deadweight to be used while varying the barge deadweight from its minimum to its maximum value. (cargo units)

dellb: Length-Breadth Ratio Increment. The incremental lb to be used while varying the tug-barge length-breadth ratio from its minimum to its maximum value.

delrate: Loading/Discharge Rate Increment. The incremental L/D rate to be used while varying the L/D from its minimum to its maximum value. (cargo units/day)

delspeed: Tug Speed Increment. The incremental tug speed to be used while varying the tug speed from its minimum to its maximum value. (kts)

dhp: Delivered Horsepower. Horsepower required to be delivered to the propeller to drive the tug-barge unit "speed" knots.

displ: Barge Displacement. (LT)

disrate: Discount Rate. Discount rate to be used in present value calculations.

distance: Port Separation Distance. Distance between Port 1 and Port 2 in miles. (nautical miles)

dq: Coefficient array used only as an input parameter in the calling of subprogram "power"; values are used in subroutine "prop" of subprogram "power."

drange: Distance Range. Number of port separation distances to be investigated.

dsopt: Drop and Swap Option. This option enables the program to investigate two types of drop-and-swap configurations. When dsopt=0, both ports have a minimum of one barge stationed for L/D. When dsopt=1, the port with lowest L/D time has no barges stationed for L/D and tug remains with barge while in port.

dt: Coefficient array used only as an input parameter in the calling of subprogram "power"; values are used in subroutine "prop" of subprogram "power."

dwt: Barge Deadweight. (cargo capacity units)

dwtb1: Array variable used to store the barge deadweight size of the optimum drop-and-swap system for given port separation distance and L/D rates. (cargo capacity units)

dwtb2: Same as dwtb1 except for integral system. (cargo capacity units)

econlife: Economic Life. Economic life of the OTGB system used in present value calculations. (yrs)

ehp: Effective Horsepower. Horsepower required to be delivered by the propeller to drive the tug-barge unit "speed" knots.

eta: Relative Rotative Efficiency Array. Array of relative rotative efficiency (x1000) values used as input parameters in the calling of subprogram "power"; values are used in subroutine "propfactors" of subprogram "power."

fbd: Barge Freeboard. (in)

fuelcons: Fuel Consumption. Amount of fuel consumed by the tug per voyage. (LT/voyage)

fuelmargin: Fuel Margin. Percentage of fuel above "fuelcons" required to be onboard tug after bunkering.

graph_type: Subroutine Plots inputs parameter used to designate type of graphical output. If equals 0, loading rate is abscissa; if equals 1, distance is abscissa.

hr: Relative Rotative Efficiency. Relative rotative efficiency (η_R) of tug-barge hull form.

hullwt75: Hull steel weight for tug-barge unit with $C_b=0.75$ as a function of lb, bt, and dwt. (LT)

hullwt80: Hull steel weight for tug-barge unit with $C_b=0.80$ as a function of lb, bt, and dwt. (LT)

hullwt85: Hull steel weight for tug-barge unit with $C_b=0.85$ as a function of lb, bt, and dwt. (LT)

i: Index Parameter used in do loop to solve for Co-efficient of Frictional Resistance for OGTB.

ibit: Subroutine "yesno" output value designating response of user to a question. If yes was signified, ibit equals 1; if no was signified, ibit equals 0.

ihp: Installed Horsepower. Horsepower of engines installed onboard tug. (horsepower)

index: Subprogram "power" Error Index. An index used to indicate if some calculation error occurred during the call of subprogram "Power".

inflafctr: Inflation Factor. A factor used to correct the total annual cost ("acost") for inflation after January 1979.

inrate: Loading/Discharging Rate Input Option Parameter. If equals 0, then the rate of loading and discharge at each port will be the same and equal at each port and will be systematically varied from the minimum to the maximum specified value. If equals 1, the loading and discharge rate at each port will be required input.

iopt: Drop and Swap/Integral Mode Option. This option is used in the main computations. When it equals 0, then drop-and-swap calculations are executed. When it equals 1, then integral mode calculations are executed.

iopt3: Detailed Output Option. This option is used to specify how much detail is desired in the output. When it equals 0, only the optimum drop-and-swap and integral system values are printed out for a given distance and loading/discharging rate. When it equals 1, drop-and-swap and integral mode

values are printed for every value of dwt, speed, distance, rate, cb, bt, and lb.

iout: Output of Input Data Option. This option is used to insure that the input parameter values are printed only once. When 0, input is printed; when 1, input is not printed.

iout2: Output of Nonconvergence Message Option. This option is used to insure that the message of nonconvergence of the waiting time routine is printed only once for a given L/D rate and port separation distance. If equals 0, message will be printed. When equals 1, message will not be printed.

iout3: Form Coefficient Under/Overflow Indicator. This variable is set to 1 if either lb, bt, or cb values are out of interpolable ranges. Otherwise its value is zero. It is used in the program to allow the user to revise his form coefficient inputs without having to input any other data.

j: Index parameter used in do loop to solve for waiting times, voyage times, and minimum tugs required in drop-and-swap mode. Also used in Yesno Subroutine.

lb: Lenth-Breadth Ratio. The ratio of tug-barge unit's length and breadth. (ft)

lbarge: Length of Barge. (ft)

lbarge75: Length of barge with Cb=0.75 as a function of lb, bt, and dwt. (ft)

lbarge80: Length of barge with Cb=0.80 as a function of lb, bt, and dwt. (ft)

lbarge85: Length of barge with Cb=0.85 as a function of lb, bt, and dwt. (ft)

ld: Length-Depth Ratio. The ratio of barge length and barge draft. (ft)

ldfact: Length-Depth Ratio Loadline Factor. This factor is used to correct the tabulated minimum free-board value (minfbd) for length-depth ratios less than 15.

litb: Length of Integrated Tug/Barge Combination. (ft)

llcoef: Minimum Loadline Coefficient Array. From this array the value of minfbd is found via linear interpolation with respect to length.

llfact: Minimum Loadline Coefficient File. The file in which llcoef is stored.

ltug: Length of Tug. (ft)

lubecons: Lube Oil Consumption. Amount of lube oil consumed by the tug per voyage. (gals/voyage)

maxb: Maximum Beam. Maximum barge beam allowed during voyage. (ft)

maxbt: Maximum Breadth-Draft Ratio. Maximum tug-barge bt value to be investigated.

maxcb: Maximum Block Coefficient. Maximum barge block coefficient to be investigated.

maxdist: Maximum Distance. Maximum port separation distance to be investigated. (nautical miles)

maxdwt: Maximum Deadweight. Maximum barge cargo deadweight capacity to be investigated. (cargo units)

maxl: Maximum Length. Maximum OGTB length allowed during voyage. (ft)

maxlb: Maximum Length-Breadth ratio. Maximum tug-barge L/B value to be investigated.

maxrate: Maximum Loading/Discharge Rate. Maximum port L/D rate to be investigated. (cargo units/day)

maxspeed: Maximum Speed. Maximum tug speed to be investigated. (kts)

maxt1: Maximum Draft Port 1. Maximum allowed draft in Port 1. (ft)

maxt2: Maximum Draft Port 2. Maximum allowed draft in Port 2. (ft)

mflowave1: Monthly Average Cargo Flow from Port 1. Average monthly flow of cargo to be loaded at Port 1 and discharged at Port 2. (cargo units)

mflowave2: Monthly Average Cargo Flow from Port 2. Average monthly flow of cargo to be loaded at Port 2 and discharged at Port 1. (cargo units)

minbargo1: Minimum Barges/Terminal Facilities at Port 1. Minimum number of terminal facilities (and barges for drop-and-swap mode only) required at Port 1 in order to be compatible with Port 1 loading/discharging rates and flow requirements. (barges/facilities)

minbargo2: Same as minbargo1 except for Port 2. (barges/facilities)

minbarges1: Storage array for "minbargo1" for the port pair trades under consideration. (barges/facilities)

minbarges2: Same as minbarges1 except for Port 2. (barges/facilities)

minbrg1,2: Storage variables for values of minbargo1 and minbargo2, respectively, used for drop-and-swap calculations when dsopt is 0.

minbt: Minimum Breadth-Draft Ratio. Minimum tug-barge B/T value to be investigated.

mincb: Minimum Block Coefficient. Minimum tug-barge Cb value to be investigated.

mindist: Minimum Distance. Minimum port separation distance to be investigated. (nautical miles)

mindwt: Minimum Deadweight. Minimum barge cargo deadweight capacity to be investigated. (cargo units)

mindwttemp: Temporary storage variable for the variable "mindwt" used for reducing the number of "dwt" iterations required.

minfbd: Minimum Freeboard. The uncorrected (or block coefficient corrected) minimum freeboard value as obtained after linear interpolation of the array llcoef.

minitb: Minimum OGTB. The minimum number of OGTB's required in the integral mode to provide sufficient flow capacity. (tug-barges)

minlb: Minimum Length-Breadth ratio. Minimum tug-barge

L/B value to be investigated.

minrate: Minimum Loading/Discharge Rate. Minimum port L/D rate to be investigated. (cargo units/day)

minspeed: Minimum Speed. Minimum tug operating speed to be investigated. (kts)

mintug: Minimum Tugs. Minimum number of tugs required in the drop-and-swap mode to provide sufficient flow capacity. (tugs)

mintugs1: Storage array for the minimum number of tugs required by the optimal drop-and-swap system for a given L/D rate and port separation distance. (tugs)

mintugs2: Storage array for the minimum number of OGTB's required by the optimal integral system for a given port L/D rate and port separation distance. (OGTBs)

multifctr1-3: Multiple Ship Cost Reduction Factors. Factors used to correct single unit costs for multi-unit orders for tugs and barges.

nextchar: Yesno Subroutine input variables used to test if yes or no response was given by user.

nrcrew: Number of Crew. Number of crew members required for a tug. (men)

nrtrips: Number of Trips. Number of voyages tug or OGTB can make during a year. (voyages/year)

o1,o2: Subroutine Plots Input Parameters representing the arrays of obscissa and ordinate coordinates of the points to be graphed.

output_type: Output Type Option. Allows user to designate whether printed output is desired. If equals 1, printed output is provided. If equals 0, printed output is not provided.

parameters: Name of file containing the semifixed parametric values used by the program.

percent: Array used to store the percentage that port time takes of the total voyage time for the port pair trades under consideration.

plot_: Subroutine used to graph output data.

plot_scale: Subroutine used to set abscissa and ordinate scale minimum and maximum values.

plot_setup: Subroutine used to set up the proper environment for the plot_ subroutine.

portimet: Tug Porttime. Time tug or OGTB remains in port during a voyage. (hrs)

power: Subprogram used to calculate the "dhp" and "ehp" as a function of the tug-barge system's principal dimensions and speed.

powerdata: File used to store data required by subprogram "power".

propef: Open Water Propeller Efficiency (η_o). The ratio of the power delivered by the propeller and the power delivered to the propeller in open water.

pvf: Present Value Factor. Factor used in present value calculations; a function of the discount rate and economic life for the system.

r: Iteration index used for port loading/discharging rates.

rate: Loading/Discharging Rates. Daily amount of cargo which can be either loaded into or discharged out of a barge located at either port. Refers to each barge of the minimum required at that port. (cargo units/day)

rc: Residual Resistance Array. Array of residual resistance coefficients used as input parameter for subprogram "power"; values are used in subroutine "resist" of subprogram "power."

resist: Resistance. Total still water hydrodynamic resistance of the OGTB combination.

rfr: Required Freight Rate. The amount of freight per cargo unit carried required to cover system operating and capital costs (present value annualized). (\$/cargo unit)

rfrb1: Storage array for the minimum required freight rate found for a drop-and-swap system with

specified port L/D rate and port separation distance. (\$/cargo unit)

rfrb2: Same as for rfrb1 except for integral mode. (\$/cargo unit)

rload1: Loading Rate at Port 1. Daily cargo loading rate at Port 1. (cargo units/day)

rload2: Loading Rate at Port 2. Daily cargo loading rate at Port 2. (cargo/units/day)

rlubeoil: Lube Oil Consumption Rate. Amount of lube oil consumed per hour. (gal/hr)

rportfuel: Inport Fuel Consumption Rate. Amount of Diesel fuel consumed per hour while tug is in port. (ton/hr)

rrange: Rate Range. Number of L/D rates to be investigated.

rseafuel: At Sea Fuel Consumption Rate. Amount of diesel fuel to be consumed per hour by the tug while steaming. (tons/hr)

runload1: Unloading Rate at Port 1. Daily cargo discharge rate per terminal at Port 1. (cargo units/day)

runload2: Unloading Rate at Port 2. Daily cargo discharge rate per terminal at Port 2. (cargo units/day)

seatimet: Tug Seetime. Time tug or OGTB is at sea; including delay time and linkage time (drop-and-swap mode only). (hours per voyage)

servmargin: Service Margin. Engine horsepower service margin used to calculate IHP.

sfc: Specific Fuel Consumption. Amount of diesel fuel consumed per horsepower per hour by the tug at sea. (lbs/hp-hr)

shp: Shaft Horsepower. Shaft horsepower required by tug to push OGTB through the water at the specified speed. (horsepower)

shrfact: Sheer Correction Factor. Correction to the free-board calculation due to lack of sheer expected in the barge form.

speed: Tug Speed. Tug designed service speed. (kts)

speedb1: Storage array for the speed of the drop-and-swap system for a given port L/D rate and port separation distance. (kts)

speedb2: Same as speedb1 except for integral mode. (kts)

sysin: System input file

sysprint: System output file

t: Thrust Deduction Fraction Array. Array of one minus the thrust deduction fraction (x1000) values used as input parameter in the calling of subprogram "power"; values are used in subroutine "propfactors" of subprogram "power."

tbarge: Barge Draft. (ft)

tdwt: Total Deadweight of OGTB. Includes cargo deadweight capacity plus fuel and miscellaneous weights. (cargo units)

temp: Temporary storage location used for various intermediate calculations.

th: Thrust Deduction Fraction. Thrust deduction fraction (t) of the tug-barge form.

tlink: Linking Time. Time required to link a tug with a barge for drop-and-swap operations. (hrs)

totcapcost: Total Capital Cost. Total system capital costs for tugs, barges, terminal facilities, (and storage facilities for integral mode only). (\$)

totopcost: Total Operating Costs. Total OGTB system operating costs. For drop-and-swap mode it includes M&R and insurance costs for those barges in excess of the number of tugs. (\$)

tport1: L/D Time Spent in Port 1. Amount of time required for L/D operations at Port 1 per barge. (hrs)

tport2: L/D Time Spent in Port 2. Amount of time required for L/D operations at Port 2 per barge (hrs)

ttript: Tug Total Voyage Time. Amount of time required

by tug for a voyage. (days)

tugopdays: Tug Operating Days. Number of days during the year in which a tug is expected to be available for operations. (days)

tunlink: Unlinking Time. Time required to disconnect barge from tug for drop-and-swap operations. (hrs)

twait1: Port 1 Waiting Time. Time tug is required to wait for barge at Port 1 during drop-and-swap operations. (hrs)

twait2: Port 2 Waiting Time. Time tug is required to wait for barge at Port 2 during drop-and-swap operations. (hrs)

vcargo: Cargo Value. Value of cargo per long ton. (\$/LT)

volwt: Cargo Cubic Capacity. The amount of cargo that could fit within the cubic volume of the barge. (LT)

w: Wake Fraction Array. Array of one minus the wake fraction (x1000) values used as an input parameter in the calling of subprogram "power"; values are used in subroutine "propfactors" of subprogram "power."

wa: Wake Fraction. Wake fraction (W_T) of the tug-barge form.

wmacht: Tug Machinery Weight. (LT)

wmisc: Tug Miscellaneous Weight. (LT)

woutfitb: Barge Outfit Weight. (LT)

woutfitt: Tug Outfit Weight. (LT)

wsteelb: Barge Hull Steel Weight. (LT)

wsteelt: Tug Steel Weight. (LT)

x: Subroutine Plots variable used to represent abscissa array values.

xcb: Quadratic Interpolation Coefficient. Parameter used in the quadratic interpolation with respect

to block coefficient for the value of lbarge and wsteelb.

x1cb: Same as for xcb.

x2cb: Same as for xcb.

x3cb: Same as for xcb.

xmax,xmin: Subroutine Plots variables used to represent minimum and maximum range of abscissa axis.

y1,y2: Subroutine Plots variables used to represent ordinate array values for drop-and-swap and integral modes respectively.

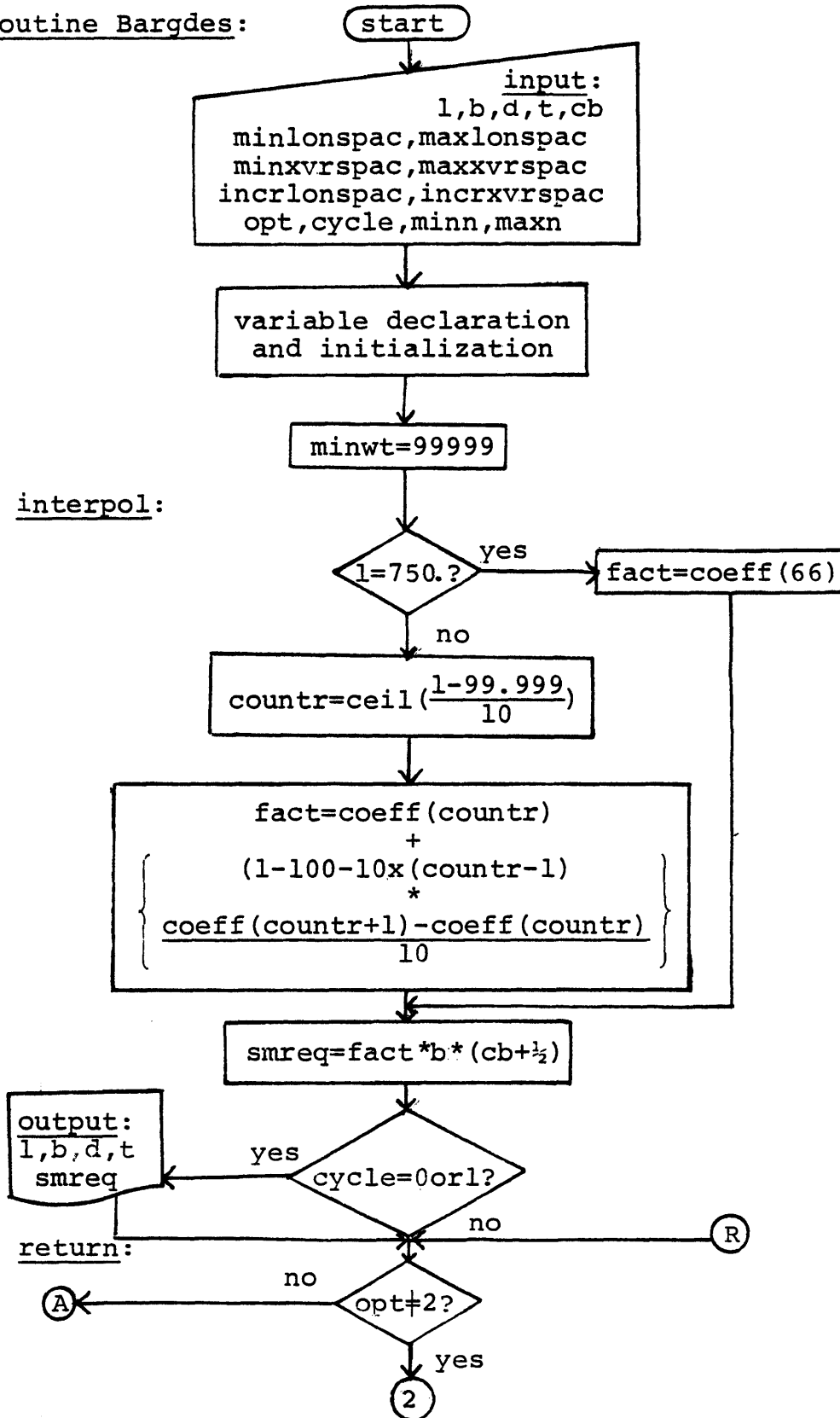
ymax,ymin: Subroutine Plots variables used to represent minimum and maximum range of ordinate axis.

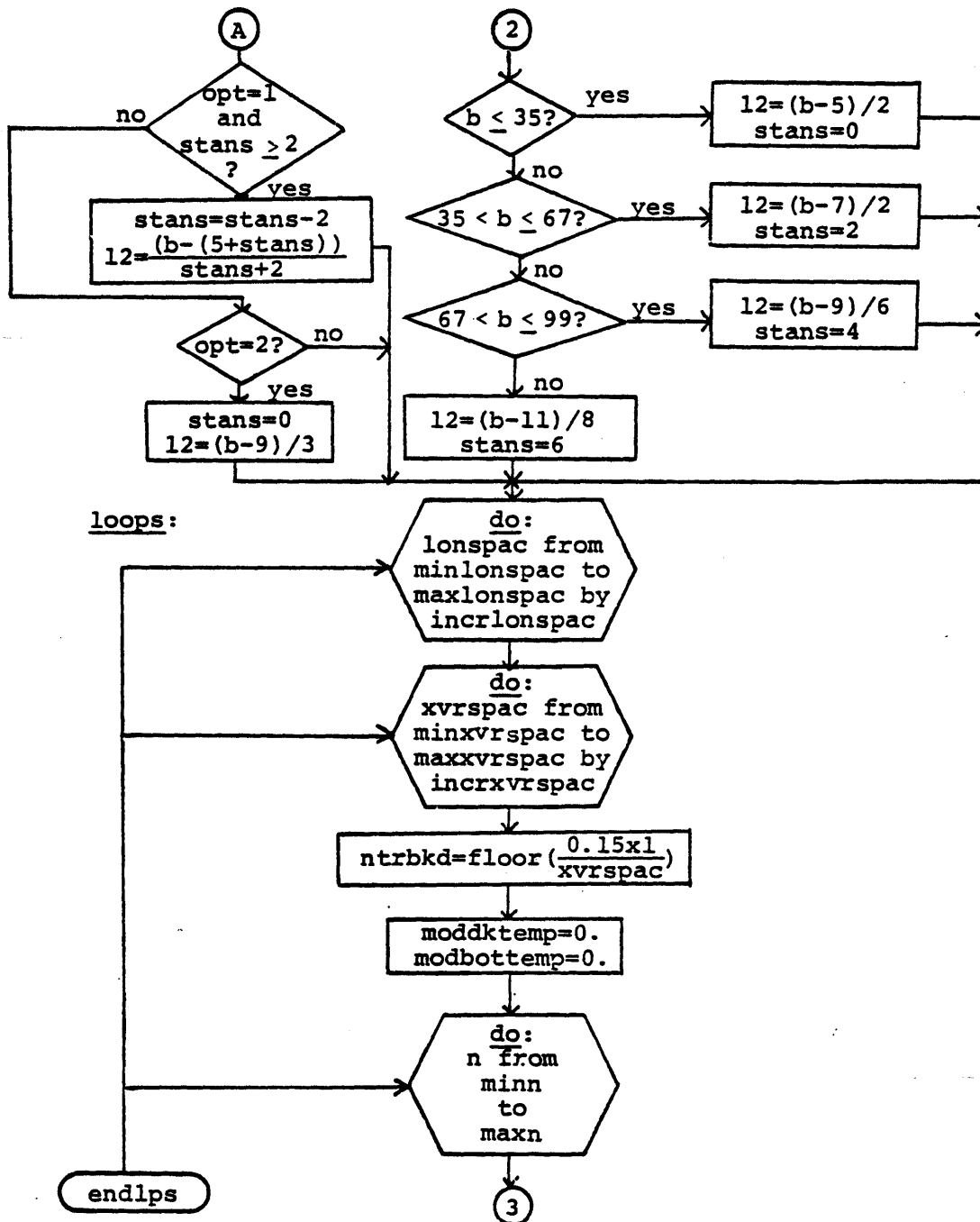
APPENDIX B

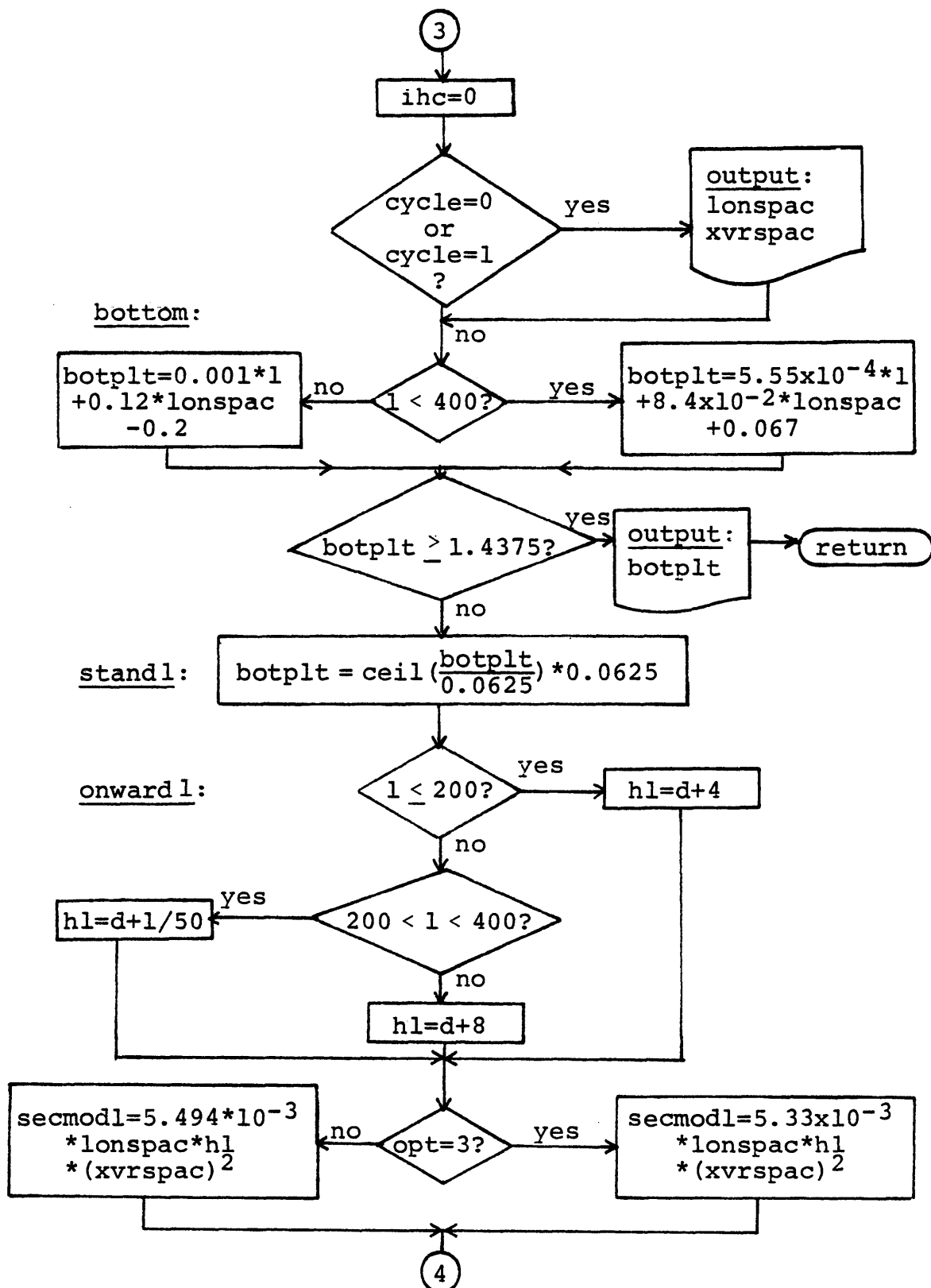
DOCUMENTATION FOR THE BARGE DESIGN MODEL

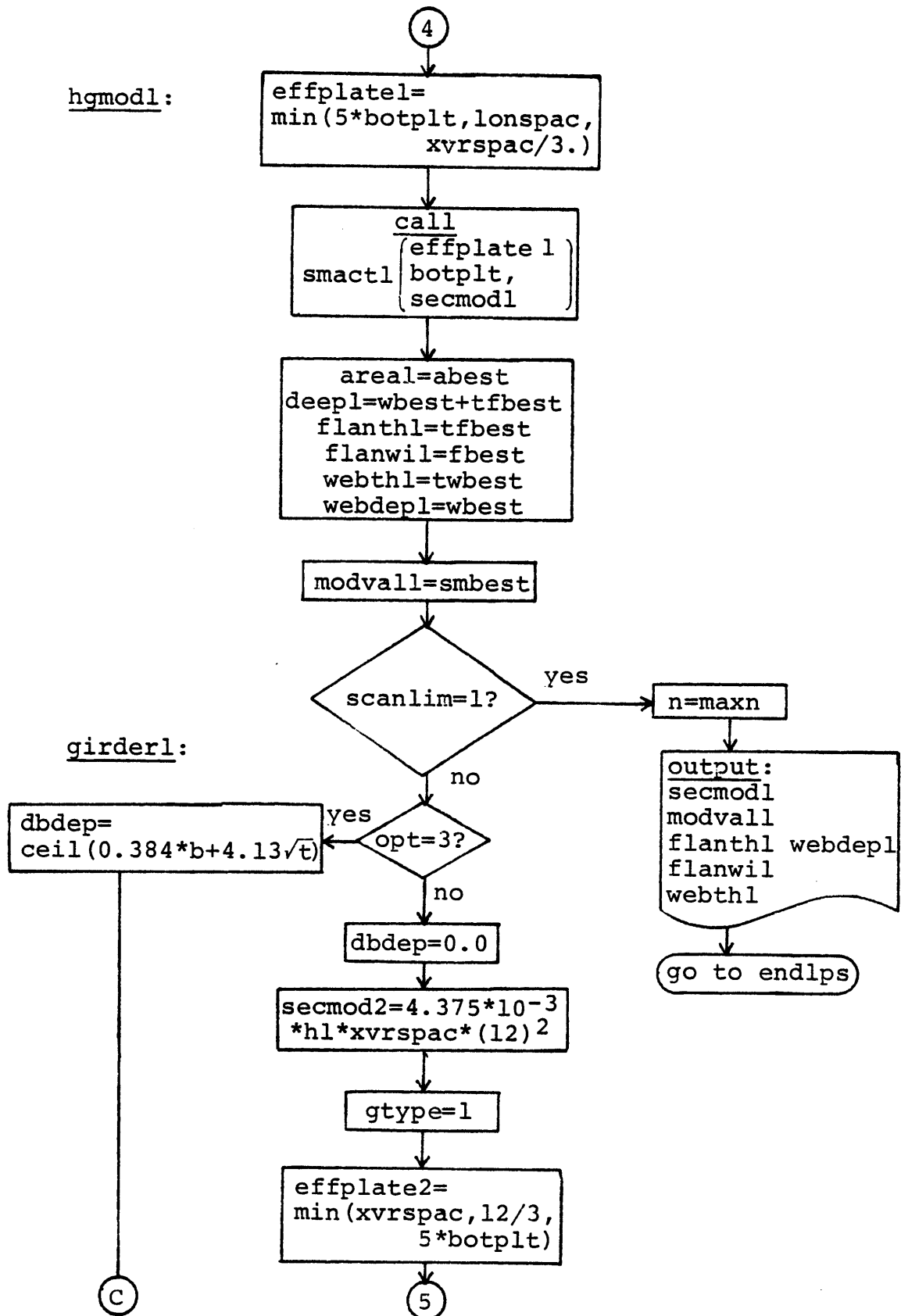
B.1 FLOWCHART OF THE BARGE DESIGN MODEL

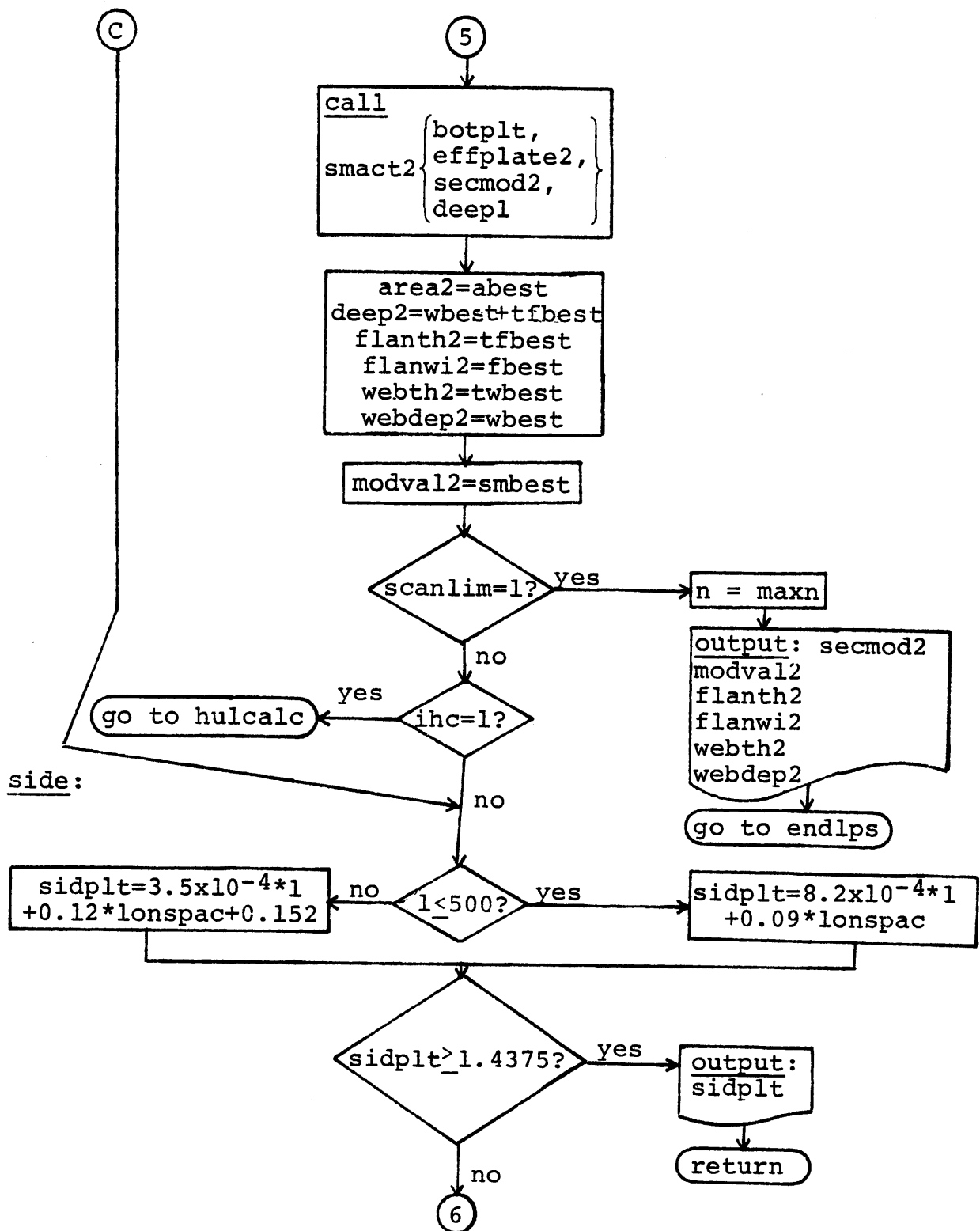
Subroutine Bargdes:





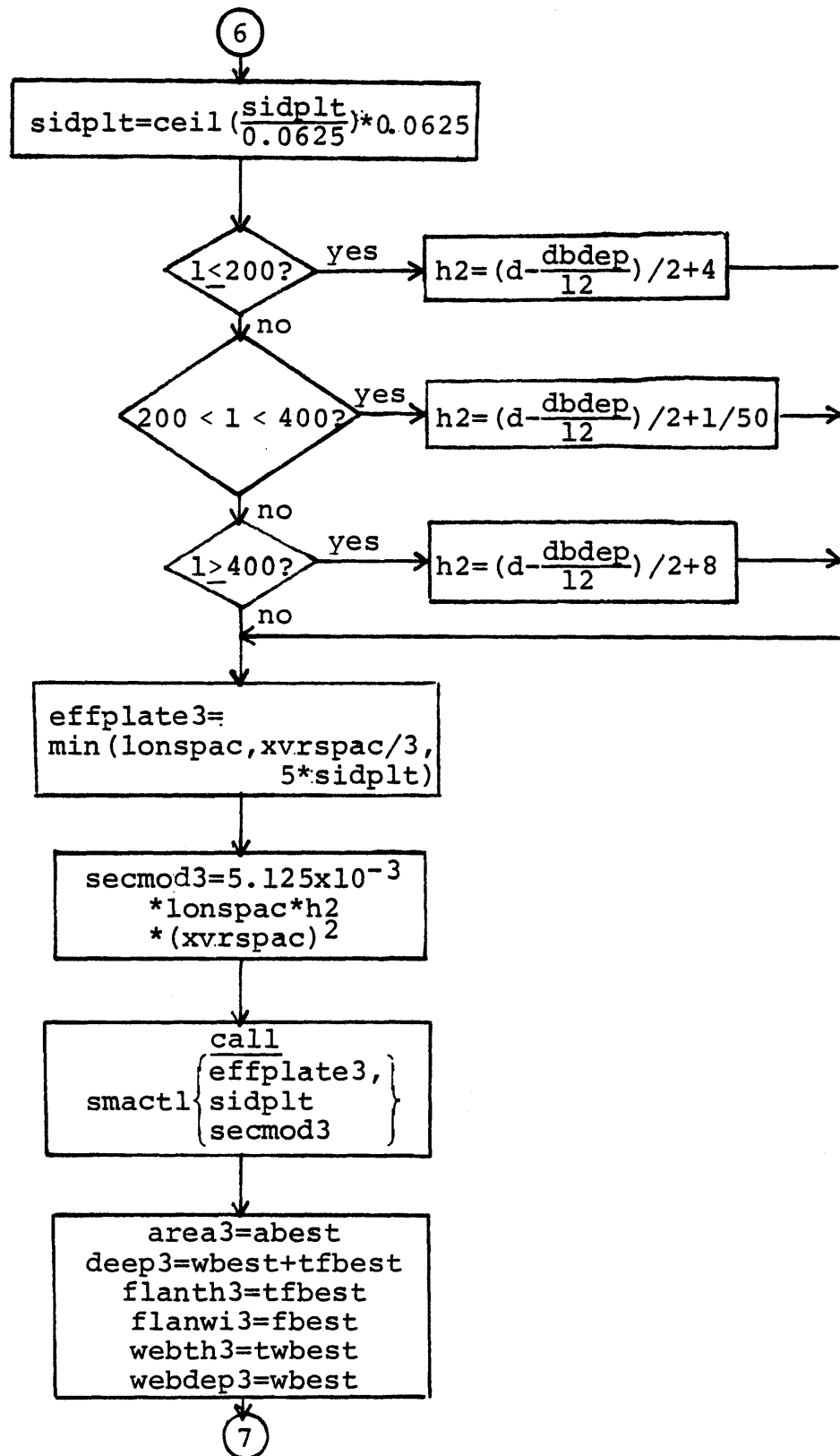




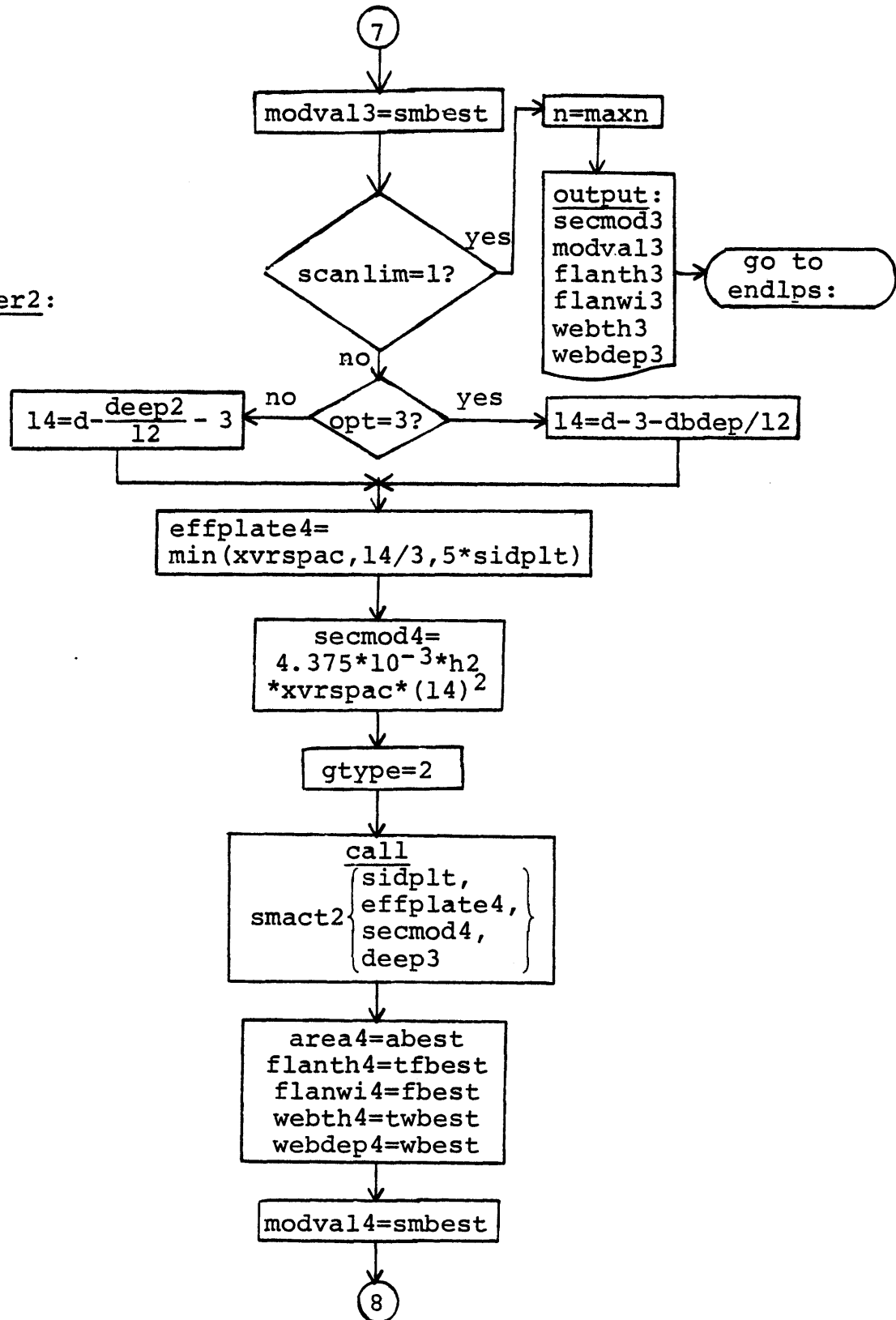


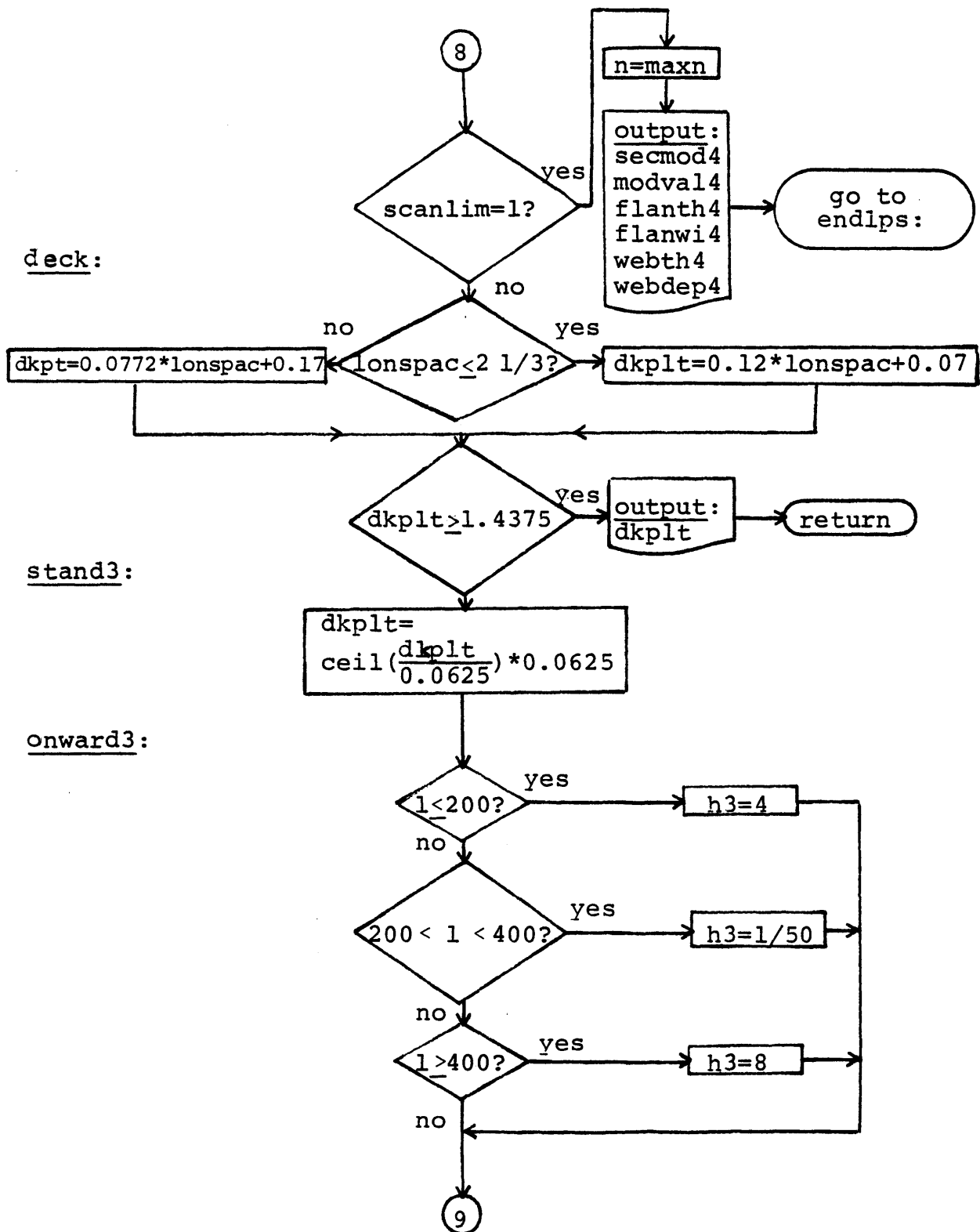
stand2:

onward2:



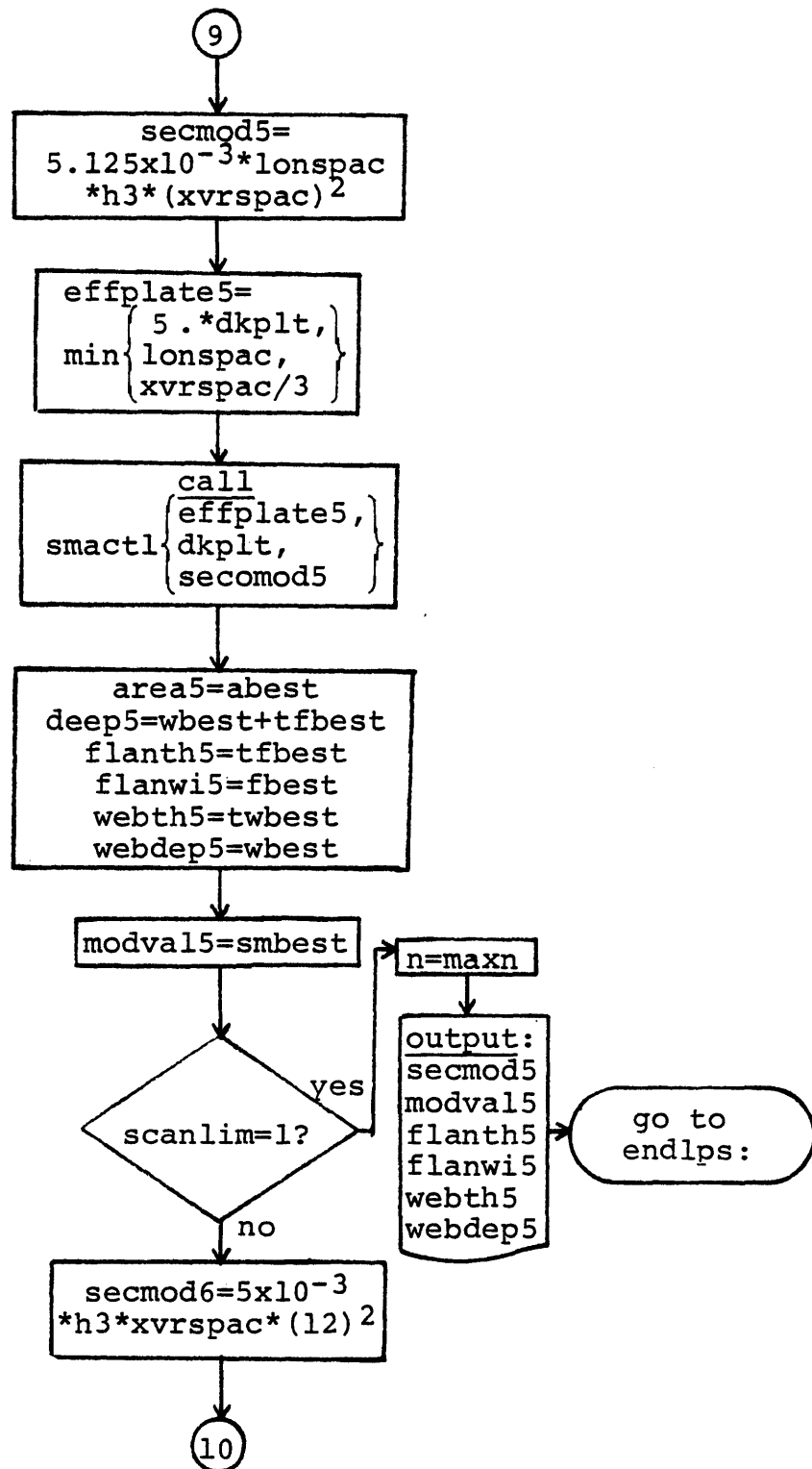
girder2:

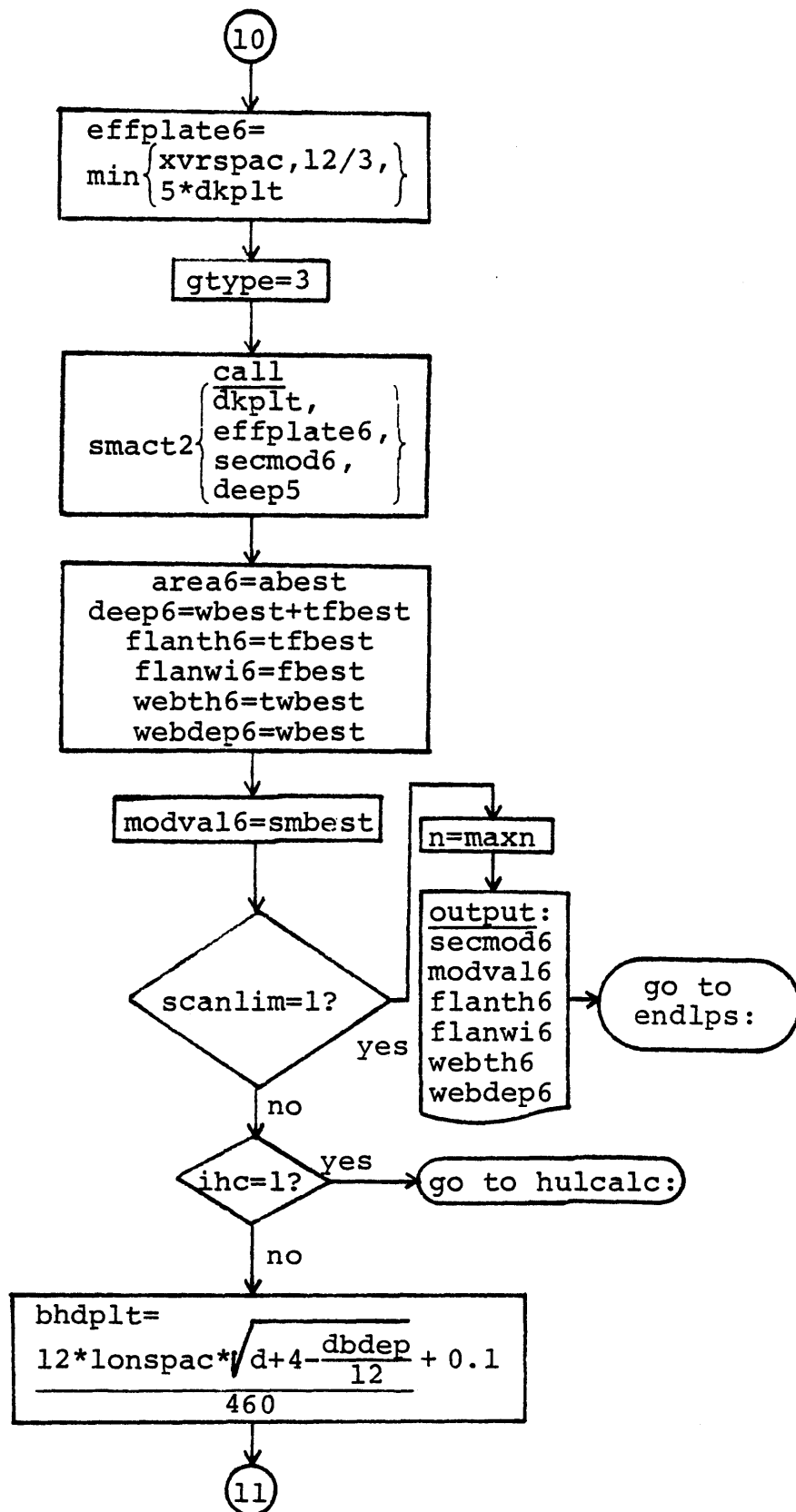




hgmod3:

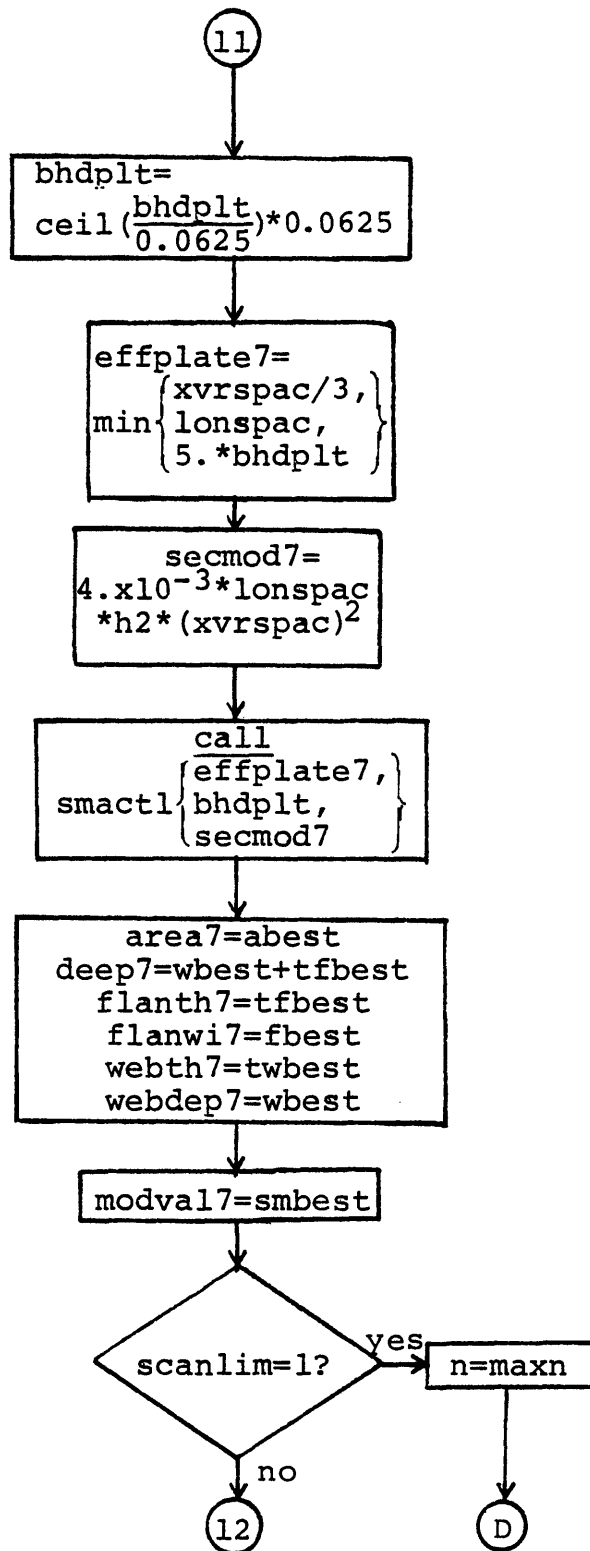
girder3:

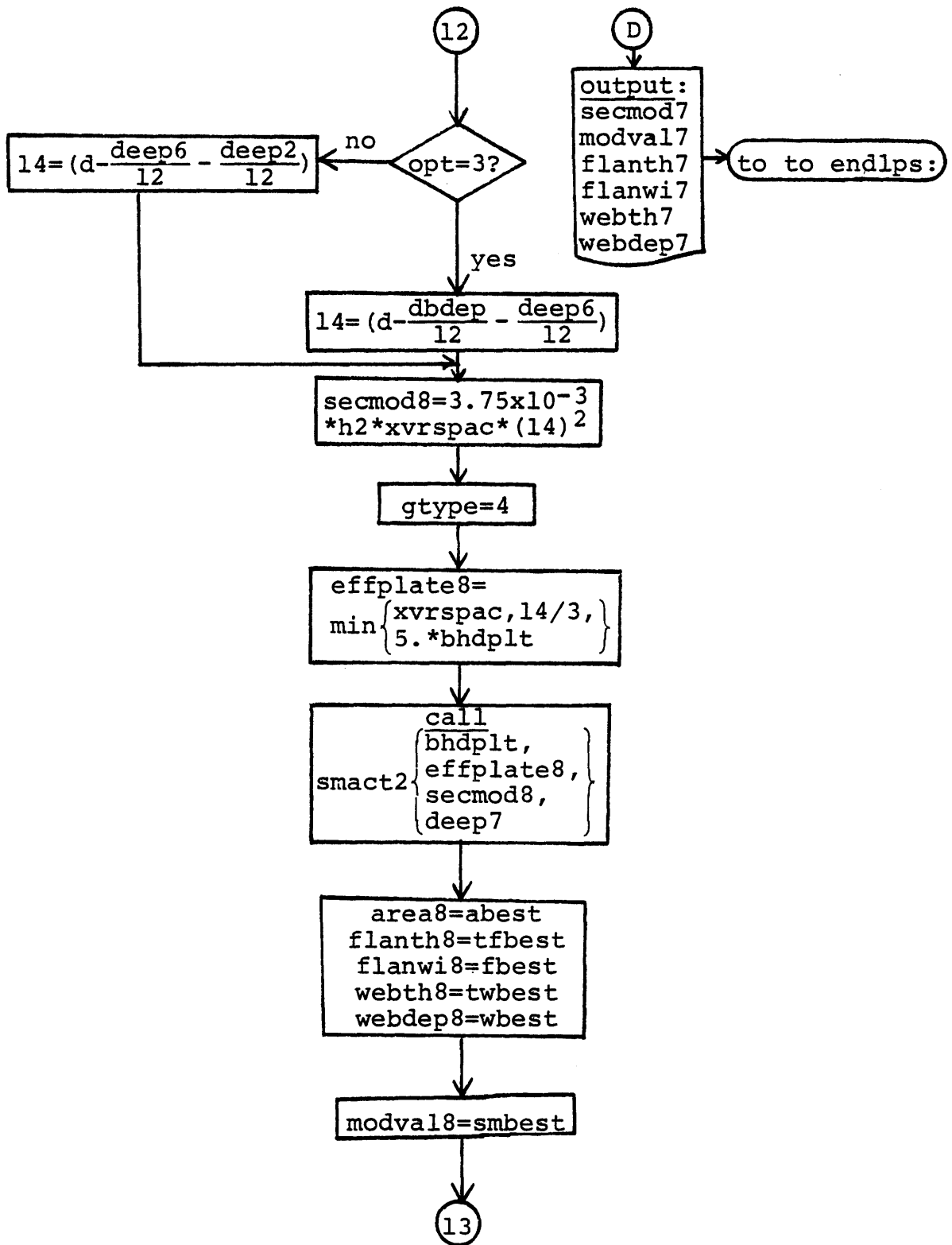


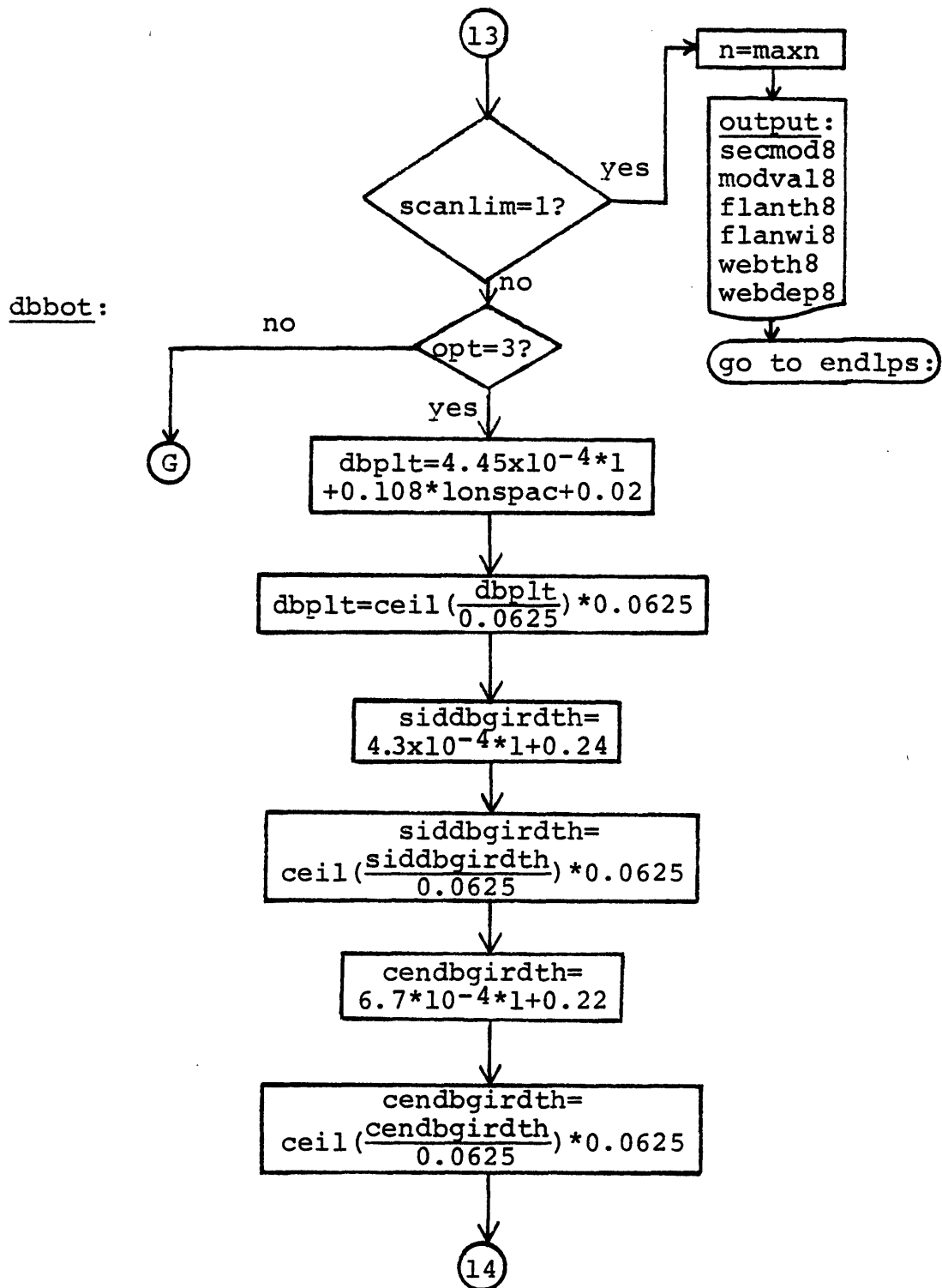


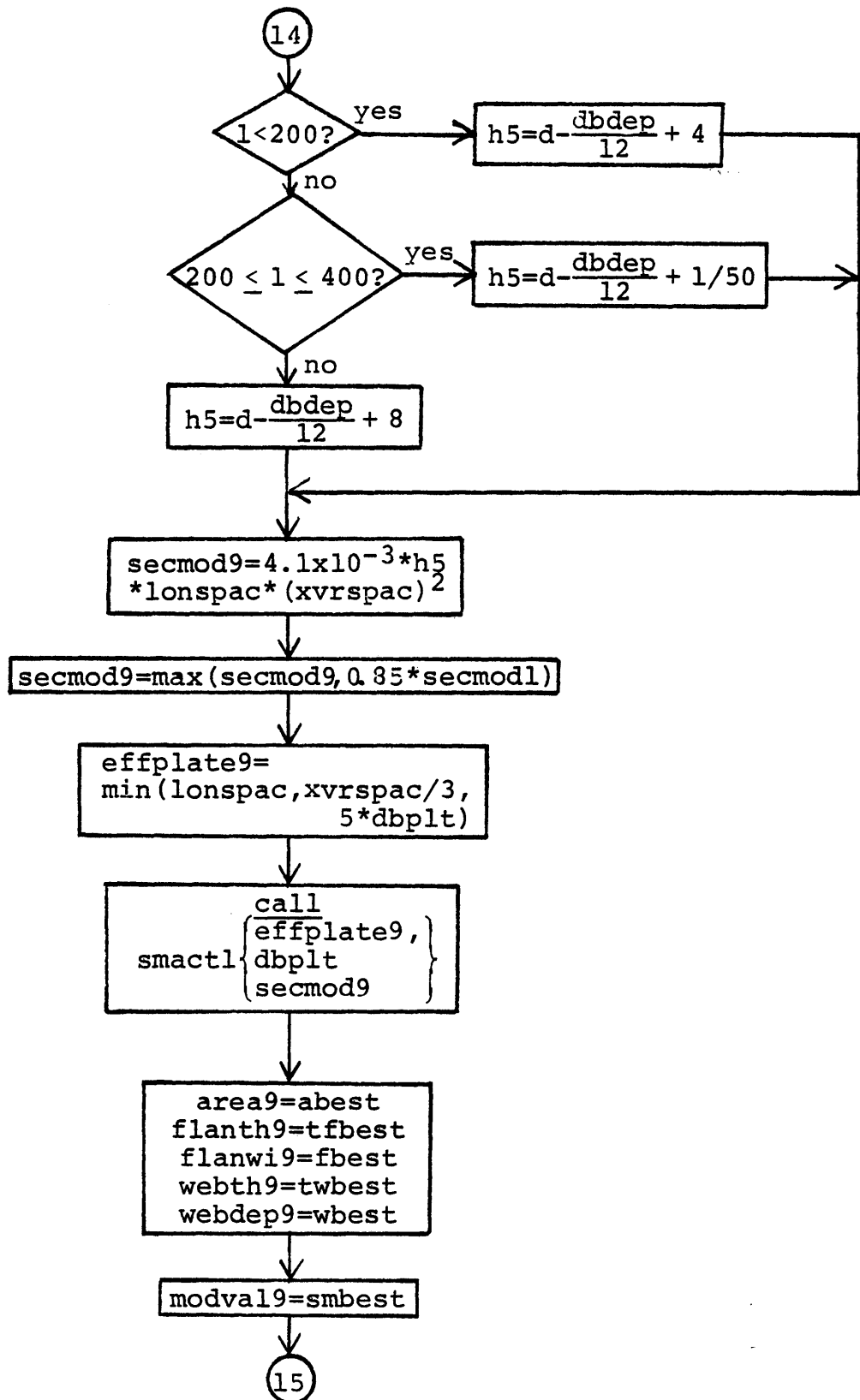
bhds:

stand4:



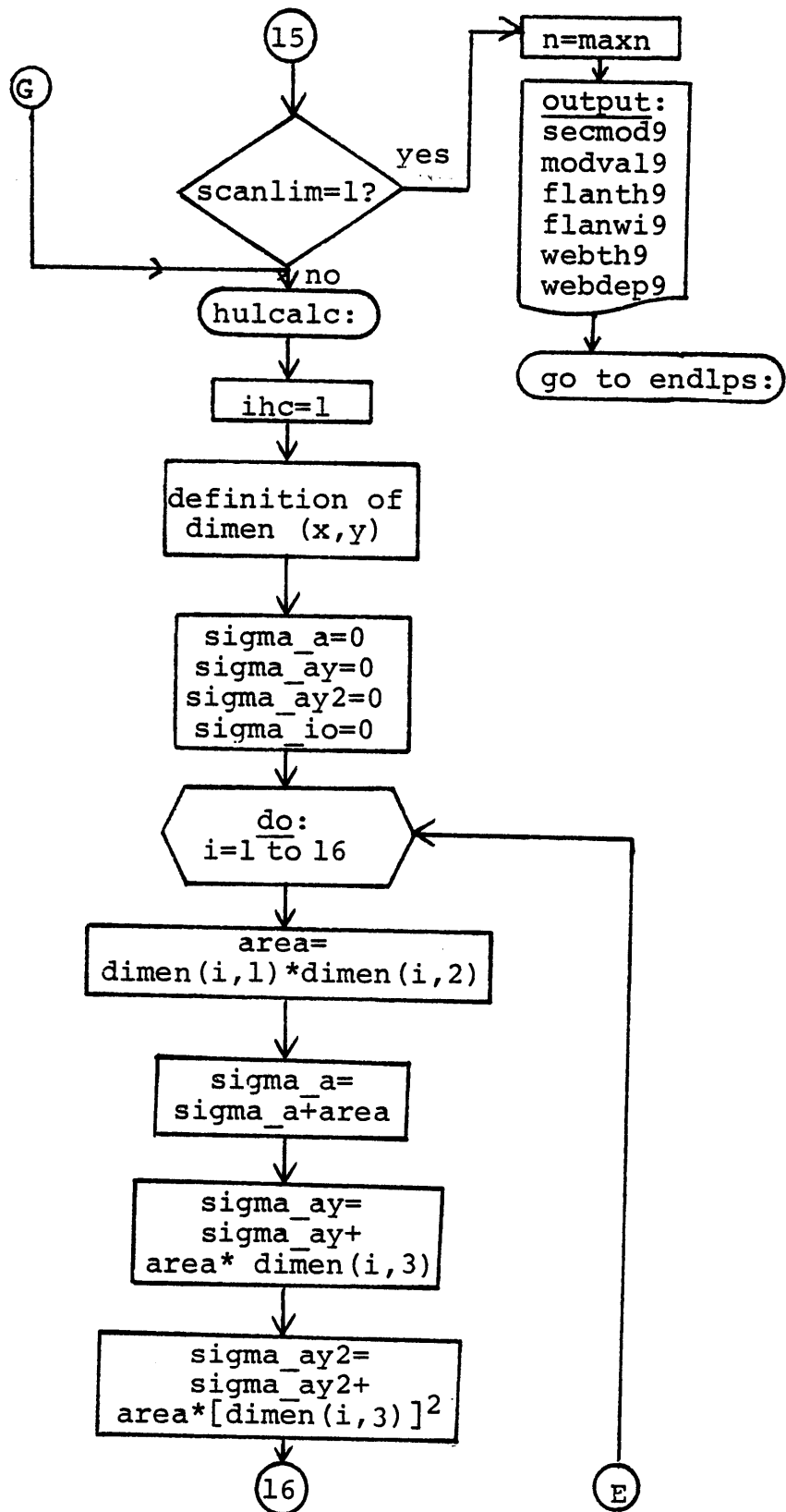


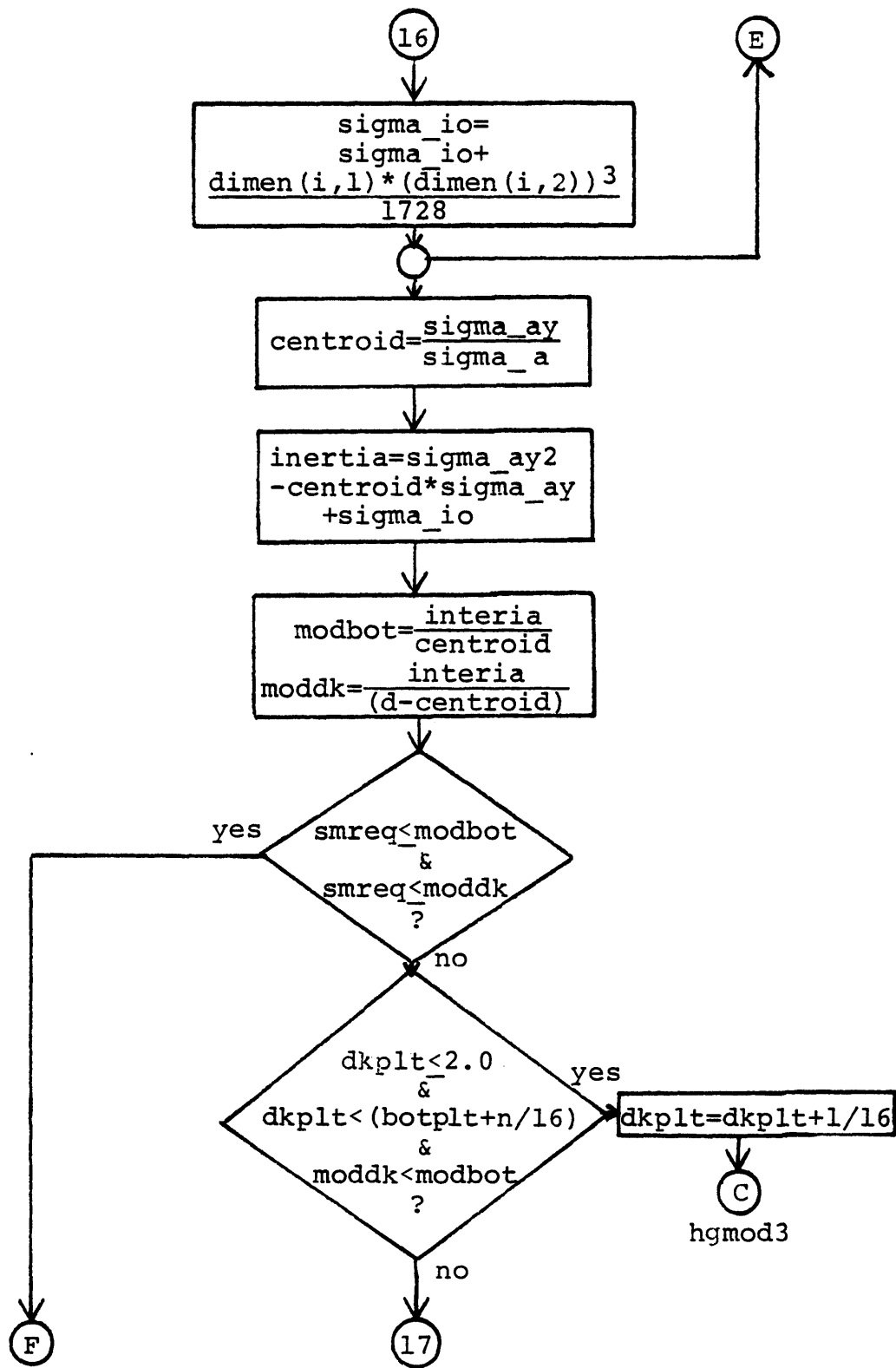


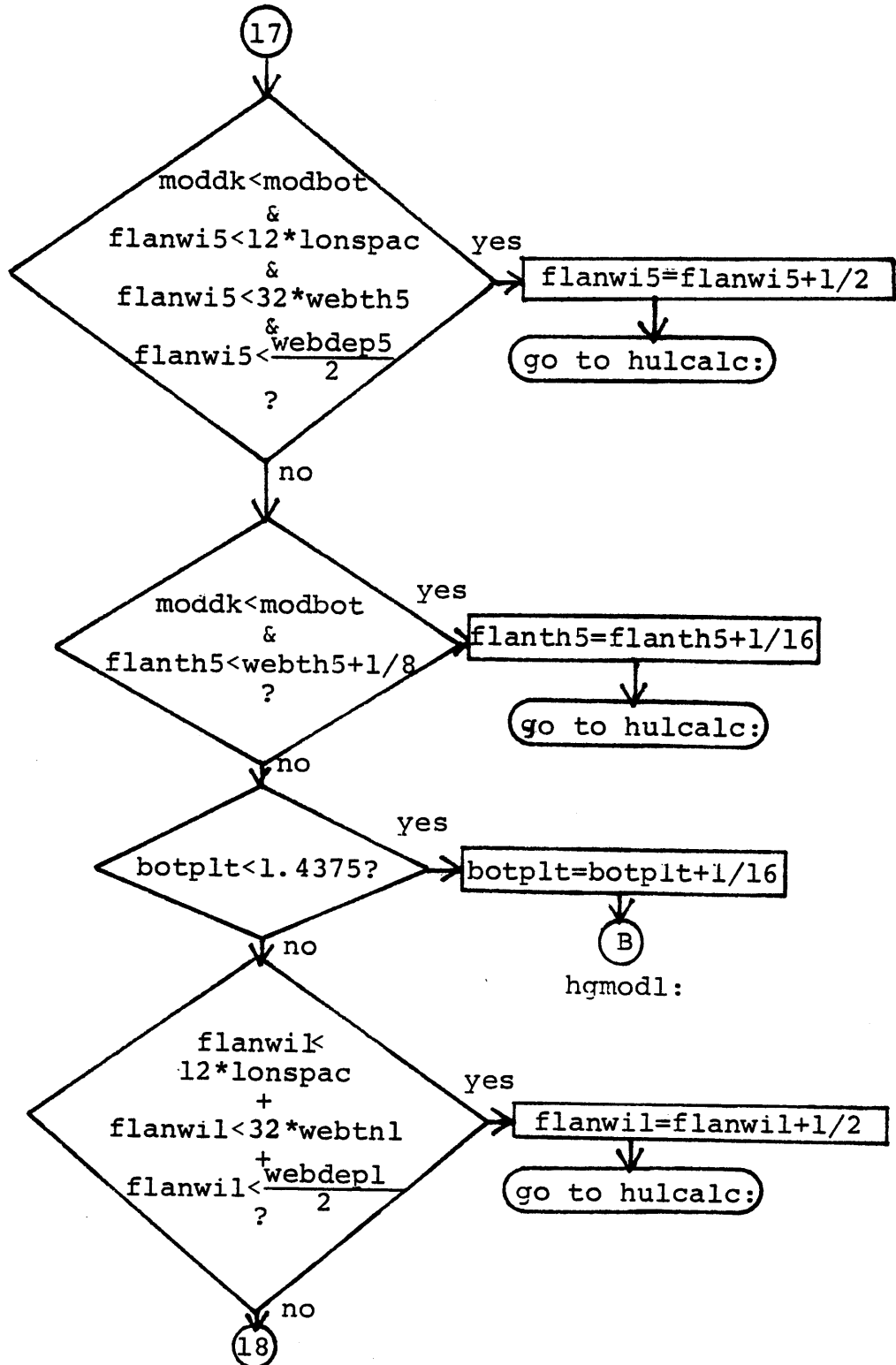


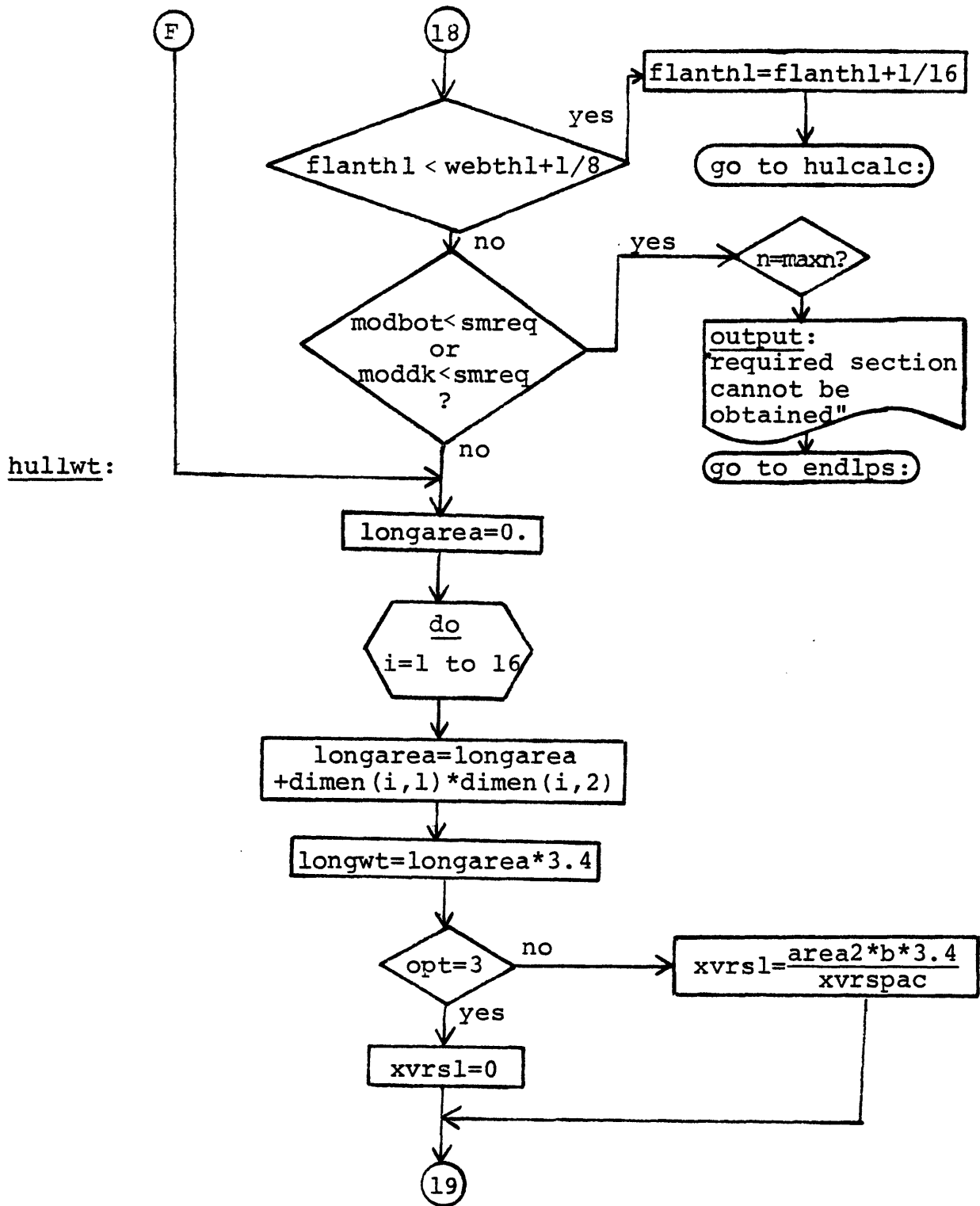
hulcalc:

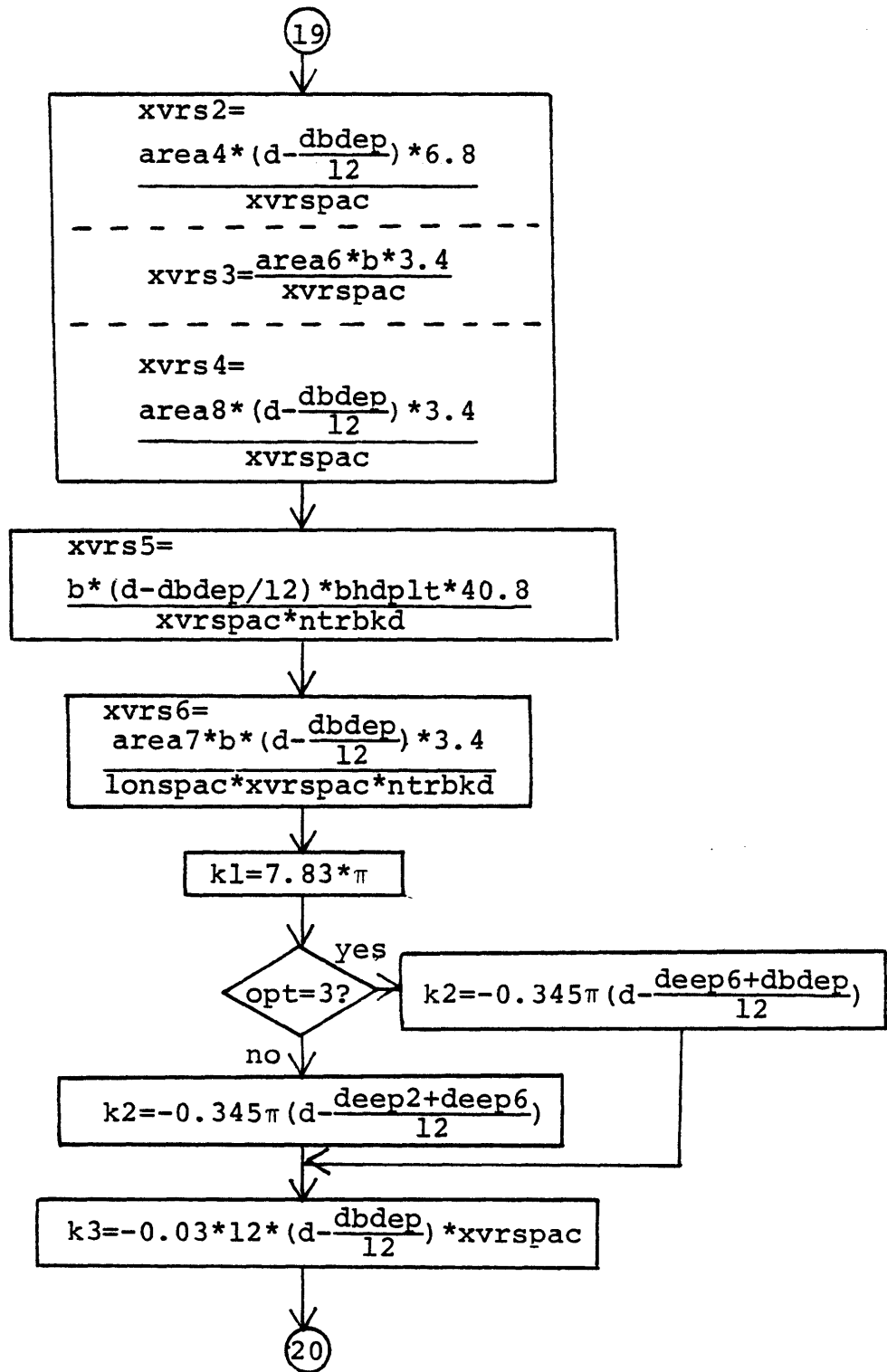
hullsm:

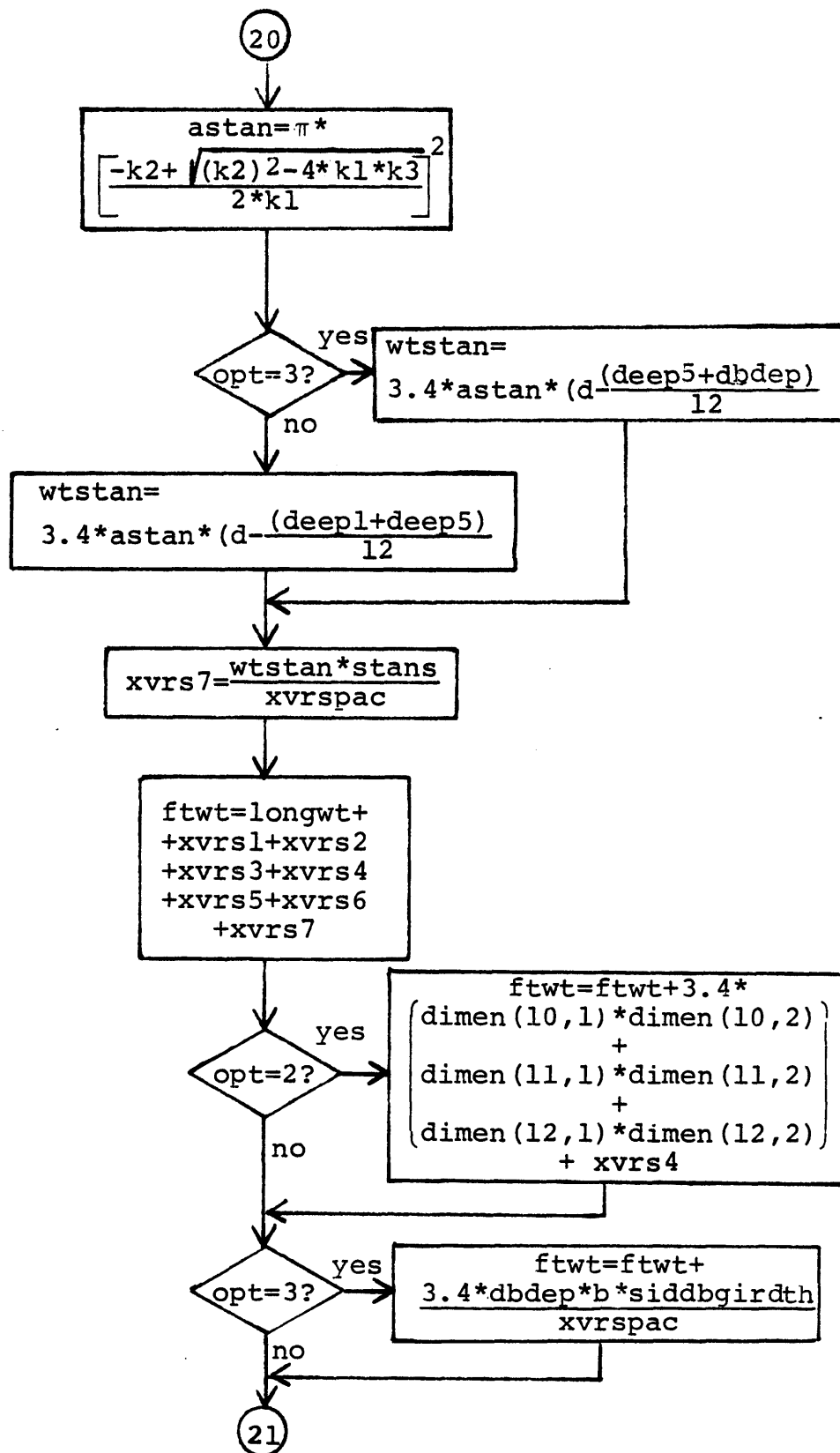


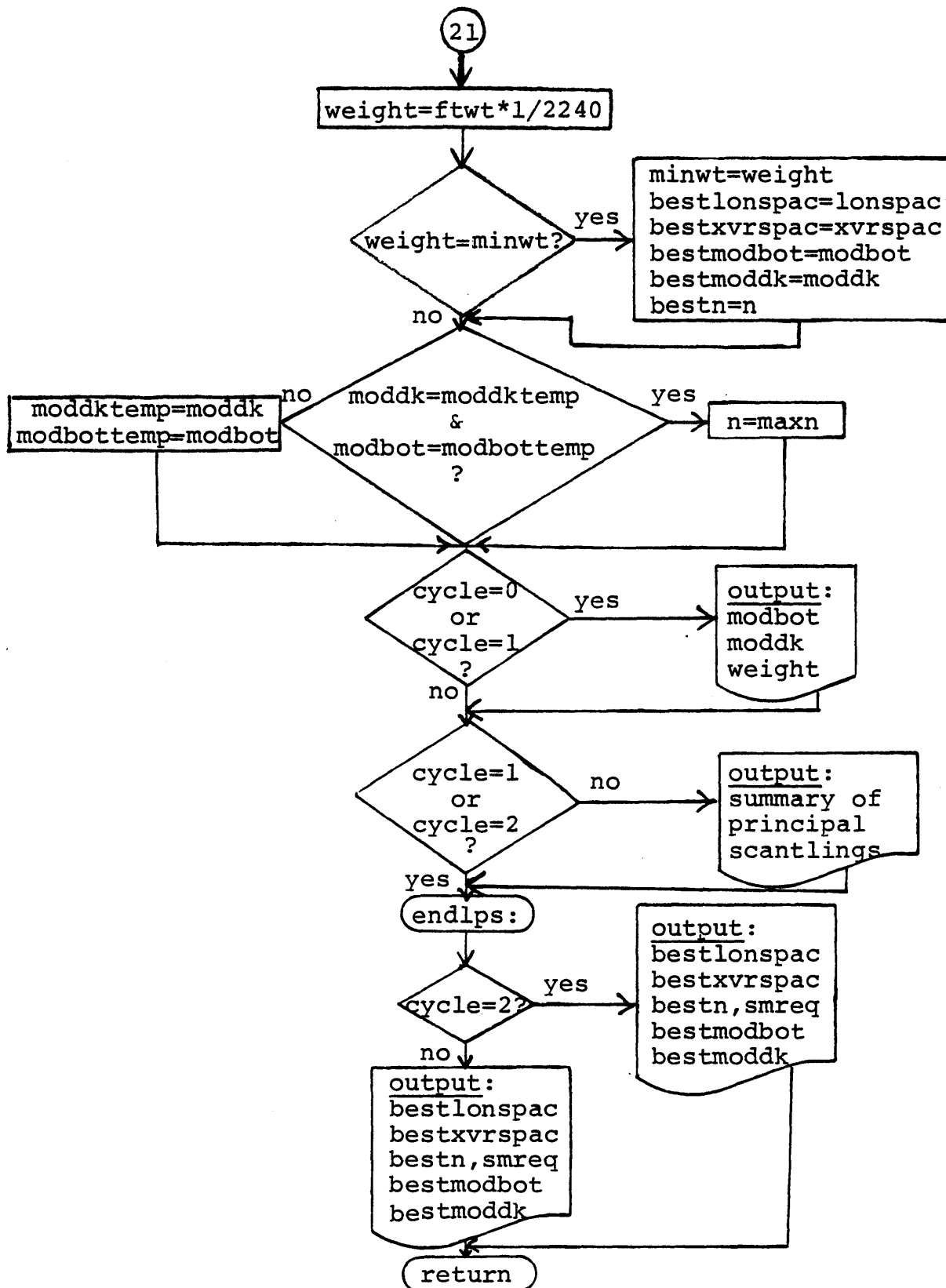












```

1 bargdes6: proc(l,b,d,t,cb,minlonspac,maxlonspac,incrlnonspac,minxvrspac,maxxvrspac,incrxxvrspac,minn,maxn,
2   coeff,cycle,opt,minwt,bestlonspac,bestxxvrspac,bestn);
3   dcl (l,b,d,t,cb,lonspac,xvrspac) dec float;
4   dcl (minlonspac, maxlonspac,incrlnonspac,minxvrspac,
5     maxxvrspac,incrxxvrspac) dec float;
6   dcl (botplt,h1,secmod1,area1,deep1,flanwi1,flanth1,webth1,
7     webdep1,modval1,12,secmod2,area2,deep2,modval2) dec float;
8   dcl (sidplt,h2,secmod3,area3,deep3,flanth3,flanwi3,webth3,
9     flanwi2,flanth2,webth2,webdep2,flanth4,flanwi4,webth4,
10    webdep4,flanth6,flanwi6,webth6,webdep6,flanth8,flanwi8,
11    webth8,webdep8,
12    webdep3,modval3,14,area4,secmod4,modval4) dec float;
13   dcl (dkplt,h3,secmod5,area5,deep5,flanth5,flanwi5,webth5,
14     webdep5,modval5,secmod6,area6,deep6,modval6) dec float;
15   dcl (bhdplt,secmod7,area7,deep7,flanth7,flanwi7,webth7,
16     webdep7,modval7,secmod8,area8,modval8) dec float;
17   dcl (ltemp,smreq,fact,sigma_a,sigma_ay,sigma_ay2,sigma_io,
18     centroid,inertia,modbot,moddk) dec float;
19   dcl (longarea,longwt,xvrs1,xvrs2,xvrs3,xvrs4,xvrs5,xvrs6,xvrs7,effplate1,effplate2,effplate3,
20     effplate4,effplate5,effplate6,effplate7,effplate8,ftwt,weight,astan,wtstan) dec float;
21   dcl (dbplt,dbdep,sidbgirdth,cendbgirdth,h5,effplate9,secmod9,modval9,
22     area9,flanth9,flanwi9,webth9,webdep9) dec float;
23   dcl (minwt,bestlonspac,bestxxvrspac,bestmodbot,bestmoddk) dec float;
24   dcl (coeff(66), dimen(16,3)) dec float;
25   dcl (abest,area,fbest,smbest,tfbest,twbest,wbest) dec float;
26   dcl (scanlim,countr,i,n,pi,ihc,stans,gtype,cycle,opt,minn,maxn,bestopt,bestn,ntrbkd) bin fixed;
27   dcl (k1,k2,k3,moddktemp,moddbottemp) dec float;
28   dcl sysprint file stream output;
29   pi=3.141592654;
30   minwt=99999.;
31   interpol: if l=750. then fact=coeff(66);
32   else do;
33     countr=ceil((1-99.9999)/10.);
34     fact=coeff(countr)+(1-100-10*(countr-1))*
35       (coeff(countr+1)-coeff(countr))/10;
36   end;
37   onward: smreq=fact*b*(cb+5e-1);
38   if cycle=0 ; cycle=1 then put skip edit("length=",l," beam=",b," depth=",d, " draft=",
39     t," required modulus=",smreq) (col(2),a,f(3),a,f(6,2),
40     a,f(5,2),a,f(5,2),a,f(6));
41   return;
42   if opt=2 then do;
43     if b<=35 then do;
44       l2=(b-5.)/2.;
45       stans=0;
46     end;
47     if b>35 & b<=67 then do;
48       l2=(b-7.)/4.;
49       stans=2;
50     end;
51     if b>67 & b<=99 then do;
52       l2=(b-9.)/6.;
53       stans=4;
54     end;
55     if b>99. then do;
56       l2=(b-11.)/8.;
57       stans=6;
58     end;
59     go to loops;

```

```

60     end;
61     if opt=1 & stans>=2 then do;
62         stans=stans-2;
63         l2=(b-(5.+stans))/(stans+2.);
64         go to loops;
65     end;
66     if opt = 2 then do;
67         stans=0;
68         l2=(b-9.)/3.;
69     end;
70 loops: do lonspac = minlonspac to maxlonspac by incr1lonspac;
71     do xvrspac = minxvrspac to maxxvrspac by incr1xvrspac;
72     ntrbk1 = floor(0.15*1/xvrspac);
73     moddktemp = 0.;
74     modbottemp = 0.;
75     do n = minn to maxn;
76         ihc=0;
77         if cycle=0 ; cycle = 1 then put skip(2) edit("longitudinal frame spacing = ",lonspac,
78             " transverse web spacing = ",xvrspac)(col(2),a,f(5,2),a,f(5,2));
79     bottom: if l<400 then
80         botplt=5.55e-4*1+8.4e-2*lonspac+6.7e-2;
81         if l>=400 then
82             botplt=1e-3*1+1.2e-1*lonspac-2e-1;
83             if botplt >= 1.4375e0 then
84                 do;
85                     put data(botplt);
86                     return;
87                 end;
88 stand1: botplt=ceil(botplt/.0625)*0.0625;
89 onward1: if l<=200 then
90         h1=d+4;
91         if l>200&l<400 then
92             h1=d+1/50;
93         if l>=400 then
94             h1=d+8;
95         if opt=3 then secmod1=5.494e-3*lonspac*h1*xvrspac*xvrspac;
96         else secmod1=5.33e-3*lonspac*h1*xvrspac*xvrspac;
97     hgmod1: effplate1=min(5.*botplt,lonspac,xvrspac/3.);
98     call smact1(effplate1,botplt,secmod1);
99     area1=abest;
100     deep1=wbest+tfbest;
101     flanth1=tfbest;
102     flawn1=fbest;
103     webth1=twbest;
104     webdep1=wbest;
105     modval1=smbest;
106     if scanlim=1 then do;
107         n = maxn;
108         put data(secmod1,modval1,flanth1,flawn1,webth1,
109             webdep1);
110         go to endlps;
111     end;
112 girder1: if opt = 3 then do;
113     dbdep = ceil(0.384*b + 4.13*sqrt(t));
114     go to side;
115     end;
116     else dbdep = 0.0;
117     secmod2=4.375e-3*h1*xvrspac*l2*l2;
118     gtype=1;
119     effplate2=min(xvrspac,l2/3.,5.*botplt);

```

```

120      call smact2(botplt,effplate2,secmod2,deep1);
121      area2=abest;
122      deep2=wbest+tfbest;
123      flanth2=tfbest;
124      flawni2=fbest;
125      webth2=twbest;
126      webdep2=wbest;
127      modval2=smbest;
128      if scanlim=1 then do;
129          n = maxn;
130          put data(secmod2,modval2,flanth2,flawni2,webth2,webdep2);
131          go to endlps;
132      end;
133      if ihc = 1 then go to hulcalc;
134      side: if l<=500 then
135          sidplt=8.2e-4*1+9e-2*lonspac;
136      if l>500 then
137          sidplt=3.5e-4*1+1.2e-1*lonspac+1.52e-1;
138      if sidplt >= 1.4375e0 then
139          do;
140              put data(sidplt);
141              return;
142          end;
143      stand2: sidplt=ceil(sidplt/.0625)*0.0625;
144      onward2: if l<=200. then
145          h2=(d-dbdep/12.)/2. + 4.;
146      if l>200&l<400 then
147          h2=(d-dbdep/12.)/2. + 1/50;
148      if l>=400 then
149          h2=(d-dbdep/12.)/2. + 8.;
150      effplate3=min(lonspac,xvrspac/3.,5.*sidplt);
151      secmod3=5.125e-3*lonspac*h2*xvrspac*xvrspac;
152      call smact1(effplate3,sidplt,secmod3);
153      area3=abest;
154      deep3=wbest+tfbest;
155      flanth3=tfbest;
156      flawni3=fbest;
157      webth3=twbest;
158      webdep3=wbest;
159      modval3=smbest;
160      if scanlim=1 then do;
161          n = maxn;
162          put data(secmod3,modval3,flanth3,flawni3,webth3,webdep3);
163          go to endlps;
164      end;
165      girder2: if opt =3 then l4 = d -3. - dbdep/12.;
166      else l4 = d-(deep2/12. + 3.);
167      effplate4=min(xvrspac,l4/3.,5.*sidplt);
168      secmod4=4.375e-3*h2*xvrspac*l4*l4;
169      gtype=2;
170      call smact2(sidplt,effplate4,secmod4,deep3);
171      area4=abest;
172      flanth4=tfbest;
173      flawni4=fbest;
174      webth4=twbest;
175      webdep4=wbest;
176      modval4=smbest;
177      if scanlim=1 then do;
178          n = maxn;
179          put data(secmod4,modval4,flanth4,flawni4,webth4,webdep4);

```

```

180          go to endlps;
181      end;
182      deck: if lonspac<=2.3333 then
183          dkplt=1.2e-1*lonspac+7e-2;
184          else
185              dkplt= 7.72e-2*lonspac+1.7e-1;
186              if dkplt >= 1.4375 then
187                  do;
188                      put data(dkplt);
189                      return;
190                  end;
191      stand?: dkplt=ceil(dkplt/.0625)*0.0625;
192      onward?: if l<=200 then
193          h3=4;
194          if l>200&l<400 then
195              h3=1/50;
196          if l>=400 then
197              h3=8;
198          secmod5=5.125e-3*lonspac*h3*xvrspac*xvrspac;
199      hgmod?: effplate5=min(5.*dkplt,lonspac,xvrspac/3.);
200      call smact1(effplate5,dkplt,secmod5);
201      area5=abest;
202      deep5=wbest+tfbest;
203      flanth5=tfbest;
204      flanwi5=fbest;
205      webth5=twbest;
206      webdep5=wbest;
207      modval5=smbest;
208      if scanlim=1 then do;
209          n = maxn;
210          put data(secmod5,modval5,flanth5,flanwi5,webth5,webdep5);
211          go to endlps;
212      end;
213      girder?: secmod6=5e-3*h3*xvrspac*12*12;
214      effplate6 = min(xvrspac,12/3.0,5.0*dkplt);
215      gtype=3;
216      call smact2(dkplt,effplate6,secmod6,deep5);
217      area6=abest;
218      deep6=wbest+tfbest;
219      flanth6=tfbest;
220      flanwi6=fbest;
221      webth6=twbest;
222      webdep6=wbest;
223      modval6=smbest;
224      if scanlim=1 then do;
225          n = maxn;
226          put data(secmod6,modval6,flanth6,flanwi6,webth6,webdep6);
227          go to endlps;
228      end;
229      if ihc = 1 then go to hulcalc;
230      bhds: bhdplt=12.*lonspac*sqrt(d+4-dbdep/12.)/460.+1.e-1;
231      stand4: bhdplt=ceil(bhdplt/.0625)*0.0625;
232      effplate7=min(lonspac,xvrspac/3.,5.*bhdplt);
233      secmod7=4.1e-3*lonspac*h2*xvrspac*xvrspac;
234      call smact1(effplate7,bhdplt,secmod7);
235      area7=abest;
236      deep7=wbest+tfbest;
237      flanth7=tfbest;
238      flanwi7=fbest;
239      webth7=twbest;

```

```

240 webdep7=wbest;
241 modval7=smbest;
242 if scanlim=1 then do;
243   n = maxn;
244   put data(secmod7,modval7,flanth7,flanwi7,webth7,webdep7);
245   go to endlps;
246 end;
247 girder4: if opt =3 then l4 = d - dbdep/12. - deep6/12.;
248           else l4 = (d - deep6/12. - deep2/12.);
249 secmod8=.75e-3*h2*xvrspac*l4*l4;
250 gtype=4;
251 effplate8=min(xvrspac,l4/3.,5.*bhdplt);
252 call smact2(bhdplt,effplate8,secmod8,deep7);
253 area8=abest;
254 flanth8=tfbest;
255 flanwi8=fbest;
256 webth8=twbest;
257 webdep8=wbest;
258 modval8=smbest;
259 if scanlim=1 then do;
260   n = maxn;
261   put data(secmod8,modval8,flanth8,flanwi8,webth8,webdep8);
262   go to endlps;
263 end;
264 dbbot: if opt =3 then do;
265   dbplt = 4.45e-4*1 + 0.108*lonspac + 0.02;
266   dbplt = ceil(dbplt/0.0625)*0.0625;
267   siddbgirdth = 4.3e-4*1 + 0.24;
268   siddbgirdth = ceil(siddbgirdth/0.0625)*0.0625;
269   cendbgirdth = 6.7e-4*1 + 0.22;
270   cendbgirdth = ceil(cendbgirdth/0.0625)*0.0625;
271   if l<200. then h5 = d - dbdep/12. + 4.;
272   if l>=200. & l<=400. then h5 = d - dbdep/12. + 1/50.;
273   if l>400. then h5 = d - dbdep/12. + 8.;
274   secmod9 = 4.1e-3*h5*lonspac*xvrspac**2;
275   secmod9 = max(0.85*secmod1,secmod9);
276   effplate9 = min(lonspac,xvrspac/3.,5.*dbplt);
277   call smact1(effplate9,dbplt,secmod9);
278   area9 = abest;
279   flanth9 = tfbest;
280   flanwi9 = fbest;
281   webth9 = twbest;
282   webdep9 = wbest;
283   modval9 = smbest;
284   if scanlim = 1 then do;
285     put data(secmod9,modval9,flanth9,flanwi9,webth9,webdep9);
286     n = maxn;
287     go to endlps;
288   end;
289 end;
290 hulcalc: ihc=1;
291 dimen = 0.0;
292 dimen(1,1)=12*b;
293 dimen(1,2)=botplt;
294 dimen(1,3)=botplt/24;
295 dimen(2,1)=webth1*b/lonspac;
296 dimen(2,2)=webdep1;
297 dimen(2,3)=(botplt+webdep1/2)/12;
298 dimen(3,1)=flanwi1*b/lonspac;
299 dimen(3,2)=flanth1;

```

```

300      dimen(3,3)=(botplt+webdep1+flanth1/2)/12;
301      dimen(4,1)=2*sidplt;
302      dimen(4,2)=12*d;
303      dimen(4,3)=d/2;
304      dimen(5,1)=webdep3*2*d/lonspac;
305      dimen(5,2)=webth3;
306      dimen(5,3)=d/2;
307      dimen(6,1)=flanth3*2*d/lonspac;
308      dimen(6,2)=flanwi3;
309      dimen(6,3)=d/2;
310      dimen(7,1)=12*b;
311      dimen(7,2)=dkplt;
312      dimen(7,3)=d-(dkplt/24);
313      dimen(8,1)=webth5*b/lonspac;
314      dimen(8,2)=webdep5;
315      dimen(8,3)=d-(dkplt+webdep5/2)/12;
316      dimen(9,1)=flanwi5*b/lonspac;
317      dimen(9,2)=flanth5;
318      dimen(9,3)=d-(dkplt+webdep5+flanth5/2)/12;
319      dimen(10,1)=bhdplt;
320      dimen(10,2)=12*d - dbdep;
321      dimen(10,3)=d/2. + dbdep/24.;
322      dimen(11,1)=webdep7*(d-dbdep/12.)/lonspac;
323      dimen(11,2)=webth7;
324      dimen(11,3)=d/2 + dbdep/24.;
325      dimen(12,1)=flanth7*d/lonspac;
326      dimen(12,2)=flanwi7;
327      dimen(12,3)=d/2 + dbdep/24.;
328      if opt =3 then do;
329          dimen(13,1) = 12.*b;
330          dimen(13,2) = dbplt;
331          dimen(13,3) = (botplt + dbdep + dbplt/2.)/12.;
332          dimen(14,1) = webth9*b/lonspac;
333          dimen(14,2) = webdep9;
334          dimen(14,3) = (botplt + dbdep - webdep9/2.)/12.;
335          dimen(15,1) = flanwi9*b/lonspac;
336          dimen(15,2) = flanth9;
337          dimen(15,3) = (botplt + dbdep -webdep9 - flanth9/2.)/12.;
338          dimen(16,1) = 2*siddbgirdth*(ceil(b/30.) - 1) + cendbgirdth;
339          dimen(16,2) = dbdep;
340          dimen(16,3) = (botplt + dbdep/2.)/12.;
341      end;
342      hullsm: sigma_a=0;
343      sigma_ay=0;
344      sigma_ay2=0;
345      sigma_io=0;
346      do i=1 to 16;
347          area=dimen(i,1)*dimen(i,2);
348          sigma_a=sigma_a+area;
349          sigma_ay=sigma_ay+area*dimen(i,3);
350          sigma_ay2=sigma_ay2+area*dimen(i,3)*dimen(i,3);
351          sigma_io=sigma_io+dimen(i,1)*dimen(i,2)**3/1728;
352      end;
353      centroid=sigma_ay/sigma_a;
354      inertia=sigma_ay2+sigma_io-centroid*sigma_ay;
355      modbot=inertia/centroid;
356      moddk=inertia/(d-centroid);
357      if smreq<=modbot&smreq<=moddk then
358          go to hullwt;
359      if dkplt<(botplt + n*0.0625) & moddk<modbot & dkplt<2. then do;

```

```

360         dkplt=dkplt+.0625;
361         go to hgmod3;
362     end;
363     if flanwi5<webdep5/2 & flanwi5<32.*webth5 & moddk<modbot & flanwi5<lonspac*12. then
364     do;
365         flanwi5=flanwi5+.5;
366         go to hulcalc;
367     end;
368     if flanth5<(webth5+0.125)&moddk<modbot then
369     do;
370         flanth5=flanth5+.0625;
371         go to hulcalc;
372     end;
373     if botplt<1.4375 then
374     do;
375         botplt=botplt+.0625;
376         go to hgmod1;
377     end;
378     if flanwi1<webdep1/2 & flanwi1<32.*webth1 & flanwi1<12.*lonspac then
379     do;
380         flanwi1=flanwi1+.5;
381         go to hulcalc;
382     end;
383     if flanth1<(webth1+0.125) then
384     do;
385         flanth1=flanth1+.0625;
386         go to hulcalc;
387     end;
388     if modbot<smreq!moddk<smreq then do;
389         if n= maxn then put edit("required section modulus cannot be obtained",
390             " with reasonable scantlings increases",lonspac,xvrspac)(col(2),a,a,2 f(6,2));
391         go to endlps;
392     end;
393     put skip(2);
394     put edit("results of hull section modulus calculations")(col(15),a);
395     go to endlps;
396     hullwt: longarea=0;
397     do i=1 to 16;
398         longarea=longarea+dimen(i,1)*dimen(i,2);
399     end;
400     longwt=longarea*3.4;
401     if opt ^= 3 then xvrsl=area2*b*3.4/xvrspac;
402     else xvrsl = 0.;
403     xvrsl2=area4*(d - dbdep/12.)*6.8/xvrspac;
404     xvrsl3=area6*b*3.4/xvrspac;
405     xvrsl4=area8*(d - dbdep/12.)*3.4/xvrspac;
406     xvrsl5=b*(d-dbdep/12.)*bhdplt*40.8/(xvrspac*ntrbkd);
407     xvrsl6=area7*(b/lonspac)*(d-dbdep/12.)*3.4/(xvrspac*ntrbkd);
408     k1=7.83*pi;
409     if opt = 3 then k2 = -0.345*pi*(d-(deep6+dbdep)/12.);
410     else k2 = -0.345*pi*(d-(deep2+deep6)/12.);
411     k2=-0.67*12*(d-dbdep/12.)*xvrspac;
412     astan=pi*((-k2+sqrt(k2*k2-4*k1*k3))/(2*k1))**2;
413     if opt ^= 3 then wtstan = 3.4*astan*(d-(deep1+deep5)/12.);
414     else wtstan = 3.4*astan*(d-(deep5+dbdep)/12.);
415     xvrsl7=wtstan*stans/xvrspac;
416     ftwt=longwt+xvrsl+xvrsl2+xvrsl3+xvrsl4+xvrsl5+xvrsl6+xvrsl7;
417     if opt = 2 then
418         ftwt=ftwt+3.4*(dimen(10,1)*dimen(10,2)+dimen(11,1)*dimen(11,2)+dimen(12,1)*dimen(12,2))-xvrsl4;
419     if opt = 3 then ftwt = ftwt + 3.4*dbdep*sidbgbirth*b/xvrspac;

```



```

420     weight=ftwt*1/2240.;
421     if weight<minwt then do;
422         minwt=weight;
423         bestlonspac=lonspac;
424         bestxvrspac=xvrspac;
425         bestmodbot=modbot;
426         bestmoddk=moddk;
427         bestn=n;
428     end;
429     if moddk=moddktemp & modbot=modbottemp then n=maxn;
430     else do;
431         modbottemp=modbot;
432         moddktemp=moddk;
433     end;
434     if cycle=0 ; cycle=1 then put skip edit("bottom modulus=",modbot,"deck modulus=",moddk,
435         "estimated bare hull weight=",weight)
436         (col(2),? (a,f(6),x(2)));
437 smact1: procedure(s,tp,smr);
438     dcl (tp,tw,w,tf,f,p,s,smr,ap,aw,af,yp,yw,yf,ym,im,smf,am) dec float;
439     if tp>=.375 then
440         tw=tp-.125;
441     else tw=tp;
442     w=.5;
443     tf=tp;
444     f = max(tw,0.750);
445     wbest,twbest,fbest,tfbest,smf,smbest=0;
446     scanlim=0.0;
447     p=12*s;
448 smcalc: ap=p*tp;
449     aw=w*tw;
450     af=f*tf;
451     yp=tp/2;
452     yw=tp+w/2;
453     yf=tp+w+tf/2;
454     ym=(ap*yp+aw*yw+af*yf)/(ap+aw+af);
455     im=ap*yp*yp+aw*yw*yw+af*yf*yf+(tw*w*w*w)/12-
456         ym*(ap*yp+aw*yw+af*yf);
457     smf=im/(tp+w+tf-ym);
458     if smf>=smr then
459 loop1: do;
460         am=aw+af;
461         abest=am;
462         wbest=w;
463         twbest=tw;
464         fbest=f;
465         tfbest=tf;
466         smbest=smf;
467         go to endsml;
468     end;
469     if w<18.&w<5*f & w<75.*tw then
470     do;
471         w=w+.5;
472         go to smcalc;
473     end;
474     if f=tw then f=ceil(f/0.25)*0.25;
475     if f<w/2 & f<32.*tw & f<lonspac*12. then
476     do;
477         f=f+.5;
478         go to smcalc;
479     end;

```

```

480     if tf<tp then
481         do;
482             tf=tf+.0625;
483             go to smcalc;
484         end;
485     if tw<tp then
486         do;
487             tw=tw+.0625;
488             go to smcalc;
489         end;
490     else do;
491         put edit("scantling limits reached in smact1")(col(2),a);
492         if cycle = 2 then put data(lonspac,xvrspac,n);
493         scanlim=1;
494         go to loop1;
495     end;
496 endsml: end smact1;
497 smact2: procedure(tp,s,smr,dep);
498     dcl (tp,tw,w,tf,f,s,smr,ap,aw,af,yp,yw,yf,ym,im,smf,am,smp,dep) dec float;
499     scanlim=0.0;
500     if gtype=2 ; gtype=4 then w=1.5*14;
501     if gtype=1 then w=2.4*12;
502     if gtype=3 then w=1.5*12;
503     w=ceil(max(w,2.5*dep)/.25)*0.25;
504     if l<=200 then tw=0.34;
505     if l>200&l<270 then tw=0.34+(1-200)/3500;
506     if l>=270&l<700 then tw=0.36 + (1-270)/3000;
507     if l>=700 then tw=0.50;
508     if gtype=4 then tw=min(0.44,1.2e-1+1.e-2*w);
509     if tp >= 0.625 then tw = max(tw, tp - .25);
510         else tw = max(tw, tp);
511     tw=ceil(max(tw,w/75)/.0625)*0.0625;
512     tf=tp;
513     f = ceil(min(w/5.0,20.0*tw)/0.25)*0.25;
514     wbest,twbest,fbest,tfbest,smf,smp,smbest=0;
515 smcalc: ap=12*s*tp;
516     aw=w*tw;
517     af=f*tf;
518     yp=tp/2;
519     yw=tp+w/2;
520     yf=tp+w+tf/2;
521     ym=(ap*yp+aw*yw+af*yf)/(ap+aw+af);
522     im=ap*yp*yp+aw*yw*yw+af*yf*yf+(tw*w*w)/12-
523         ym*(ap*yp+aw*yw+af*yf);
524     smf=im/(tp+w+tf-ym);
525     smp=im/ym;
526     if smf>=smr & smp>=smr then
527 loop1:     do;
528         am=aw+af;
529         abest=am;
530         wbest=w;
531         twbest=tw;
532         fbest=f;
533         tfbest=tf;
534         if smf>=smp then
535             sbest=smp;
536         else sbest=smf;
537     if ((gtype=1 ; gtype=3) & w>48.) ; w>88. then do;
538         put skip list("transverse web depth limit exceeded for girder type:",gtype);
539         scanlim = 1.0;

```

```

540         if cycle=2 then n=maxn;
541         return;
542     end;
543         go to ends2;
544     end;
545     if w<47. & w<=5.*f & w<=75.*tw then
546     do;
547         w=w+1;
548         go to smcalc;
549     end;
550     if f<=w/2.0 & f<=32.*tw then
551     do;
552         f=f+1;
553         go to smcalc;
554     end;
555     if tf<(tw+0.5) & tf<1.4375 then do;
556         tf = tf + 0.0625;
557         go to smcalc;
558     end;
559     if tw<(tp+0.5) & tw<1.4375 then do
560         tw = tw + 0.0625;
561         go to smcalc;
562     end;
563     if ((gtype=2 | gtype=4) & w<87.) & w<=5.*f & w<=75.*tw then do;
564         w = w + 1;
565         go to smcalc;
566     end;
567     if gtype=4 & tw<1.4375 then do;
568         tw = tw + 0.0625;
569         go to smcalc;
570     end;
571     else do;
572         put edit("scantlings limits reached in smact2")(col(2),a);
573         if cycle = 2 then put data(lonspac,xvrspac,n);
574         scanlim=1;
575         go to loop1;
576     end;
577     ends2: end smact2;
578     if cycle = 1 | cycle = 2 then go to endlps;
579     put skip(2) edit("summary of principal scantlings")(col(20),a);
580     put skip(2);
581     put edit("bottom shell - ",botplt," in. side shell - ",sidplt,
582     " in. bulkhead plating - ",bhdplt," in. deck plate - ",dkplt)
583     (col(2),a,f(6,4),a,f(6,4),a,f(6,4),a,f(6,4));
584     if opt = 3 then put edit(" in. doub-bot shell - ",dbplt) (a,f(6,4),a);
585     put edit("bottom longitudinals: flange- ",flanwi1," x ",flanth1,
586     "; web- ",webdep1," x ",webth1)(col(2),a,x(2),f(5,2),a,f(6,4),
587     a,f(5,2),a,f(6,4));
588     if opt=3 then put edit("bottom transverses: flange- ",flanwi2," x ",flanth2,
589     "; web- ",webdep2," x ",webth2)(col(2),a,x(4),f(5,2),a,f(6,4),
590     a,f(5,2),a,f(6,4));
591     put edit("side longitudinals: flange- ",flanwi3," x ",flanth3,
592     "; web- ",webdep3," x ",webth3)(col(2),a,x(4),f(5,2),a,f(6,4),
593     a,f(5,2),a,f(6,4));
594     put edit("side transverses: flange- ",flanwi4," x ",flanth4,
595     "; web- ",webdep4," x ",webth4)(col(2),a,x(6),f(5,2),a,f(6,4),
596     a,f(5,2),a,f(6,4));
597     put edit("bulkhead longitudinals: flange- ",flanwi7," x ",flanth7,
598     "; web- ",webdep7," x ",webth7)(col(2),a,f(5,2),a,f(6,4),
599     a,f(5,2),a,f(6,4));

```

```

600      put edit("bulkhead transverses: flange- ",flanwi8," x ",flanth8,
601      "; web- ",webdep8," x ",webth8)(col(2),a,x(2),f(5,2),a,f(6,4),
602      a,f(5,2),a,f(6,4));
603      put edit("deck longitudinals: flange- ",flanwi5," x ",flanth5,
604      "; web- ",webdep5," x ",webth5)(col(2),a,x(4),f(5,2),a,f(6,4),
605      a,f(5,2),a,f(6,4));
606      put edit("deck transverses: flange- ",flanwi6," x ",flanth6,
607      "; web- ",webdep6," x ",webth6)(col(2),a,x(6),f(5,2),a,f(6,4),
608      a,f(5,2),a,f(6,4));
609      if opt = 3 then put edit("doub-bott longitudinals: flange-",flanwi9," x ",flanth9,"; web- ",webdep9,
610      " x ",webth9) (col(2),a,f(5,2),a,f(6,4),a,f(5,2),a,f(6,4));
611      endlps: end; end; end;
612      if cycle=2 then put edit(bestlonspac,bestxvrspac,bestn,smreq,bestmodbot,bestmoddk)(f(5,2),f(4),x(3),f(2),3 f(6));
613      else put data(bestlonspac,bestxvrspac,bestn,smreq,bestmodbot,bestmoddk);
614      return;
615      end bargdes6;

```

NAMES DECLARED IN THIS COMPILATION:

IDENTIFIER	OFFSET	LOC STORAGE CLASS	DATA TYPE	ATTRIBUTES AND REFERENCES
NAMES DECLARED BY DECLARE STATEMENT.				
abest	001070	automatic	float dec(10)	dcl 25 set ref 93 121 153 171 201 217 235 253 278 461 529
af	000122	automatic	float dec(10)	dcl 498 set ref 517 521 521 522 522 524
af	000125	automatic	float dec(10)	dcl 438 set ref 450 454 454 455 455 460
an	000152	automatic	float dec(10)	dcl 438 set ref 50 461
am	000147	automatic	float dec(10)	dcl 498 set ref 528 529
ap	000117	automatic	float dec(10)	dcl 438 set ref 448 454 454 455 455
ap	000114	automatic	float dec(10)	dcl 498 set ref 515 521 521 522 522
area	001073	automatic	float dec(10)	dcl 25 set ref 347 348 348 350
area1	000117	automatic	float dec(10)	dcl 6 set ref 99
area2	000152	automatic	float dec(10)	dcl 6 set ref 121 401
area3	000174	automatic	float dec(10)	dcl 8 set ref 153
area4	000304	automatic	float dec(10)	dcl 8 set ref 171 403
area5	000326	automatic	float dec(10)	dcl 13 set ref 201
area6	000356	automatic	float dec(10)	dcl 13 set ref 217 404
area7	000375	automatic	float dec(10)	dcl 15 set ref 235 407
area8	000425	automatic	float dec(10)	dcl 15 set ref 253 405
area9	000623	automatic	float dec(10)	dcl 21 set ref 278
astan	000565	automatic	float dec(10)	dcl 19 set ref 412 413 414
an	000117	automatic	float dec(10)	dcl 498 set ref 516 521 521 522 522 524
an	000122	automatic	float dec(10)	dcl 438 set ref 449 454 454 455 455 460
b		parameter	float dec(10)	dcl 3 ref 1 37 38 43 44 47 47 43 51 51 52 55 56 63 68 113 232 295 298 310 313 315 329 332 335 338 401 404 405 407 419
bestionspac		parameter	float dec(10)	dcl 23 set ref 1 423 612 613
bestmodcot	000642	automatic	float dec(10)	dcl 23 set ref 425 612 613
bestmodum	000645	automatic	float dec(10)	dcl 23 set ref 426 612 613
bestn	001126	automatic	fixed bin(17,0)	dcl 26 set ref 427 612 613
bestxvrspac		parameter	float dec(10)	dcl 23 set ref 1 424 612 613
ondp1f	000367	automatic	float dec(10)	dcl 15 set ref 230 231 231 232 234 251 252 319 406 581
botp1f	000106	automatic	float dec(10)	dcl 6 set ref 79 81 83 85 88 88 97 98 119 120 293 294 297 300 331 334 337 346 359 373 375 375 581
cb		parameter	float dec(10)	dcl 3 ref 1 37
cenlogirdth	000604	automatic	float dec(10)	dcl 21 set ref 269 270 270 339
centroid	000460	automatic	float dec(10)	dcl 17 set ref 353 354 355 356
coeff		parameter	float dec(10)	array dcl 24 ref 1 31 34 34 34
countn	001116	automatic	fixed bin(17,0)	dcl 26 set ref 33 34 34 34 34
cyclic		parameter	fixed bin(17,0)	dcl 26 set ref 1 38 38 77 77 434 434 578 578 612 492 540 573
d		parameter	float dec(10)	dcl 3 ref 1 38 89 91 93 144 146 144 165 166 230 247 248 271 272 273 302 303 334 336 327 309 312 315 318 320 321 322 324 325 327 356 403 405 436 407 409 410 411 413 414
dboep	000576	automatic	float dec(10)	dcl 21 set ref 113 116 144 146 148 165 230 247 271 272 273 320 321 322 324 327 331 334 337 339 340 403 415 406 407 409 411 414 419
dpol1	000573	automatic	float dec(10)	dcl 21 set ref 265 266 266 276 277 330 331 584
deep1	000122	automatic	float dec(10)	dcl 6 set ref 100 120 413
deep2	000155	automatic	float dec(10)	dcl 6 set ref 122 166 248 410
deep3	000177	automatic	float dec(10)	dcl 8 set ref 154 170
deep5	000331	automatic	float dec(10)	dcl 13 set ref 202 216 413 414
deep6	000361	automatic	float dec(10)	dcl 13 set ref 218 247 248 409 413
deep7	000403	automatic	float dec(10)	dcl 15 set ref 236 252

k1	001130	automatic	float dec(10)	dcl 27 set ref 408 412 412
k2	001133	automatic	float dec(10)	dcl 27 set ref 409 410 412 412 412
k3	001136	automatic	float dec(10)	dcl 27 set ref 411 412
i		parameter	float dec(10)	dcl 3 ref 1 31 33 34 38 72 79 79 91 81 89 91 91 91 93 134 134 135 136 144 146 146 146 146 146 146 194 194 194 195 265 267 269 271 272 272 272 273 420 534 505 505 505 506 506 506 506 507 dcl 6 set ref 44 48 52 56 63 68 117 117 117 213 213 214 411 501 502 dcl 8 set ref 16 166 167 168 169 247 248 249 249 251 500 dcl 19 set ref 396 398 398 400 dcl 19 set ref 400 416 dcl 3 set ref 70 77 79 91 95 96 97 134 136 150 151 182 192 194 198 199 230 232 233 265 274 276 295 298 314 307 313 316 322 325 332 335 363 378 389 407 423 475 492 573 dcl 4 ref 1 70 dcl 26 ref 1 75 107 129 161 178 209 225 243 260 286 389 429 540 dcl 4 ref 1 71 dcl 4 ref 1 79 dcl 26 ref 1 75 dcl 23 set ref 1 30 421 422 dcl 4 ref 1 71 dcl 17 set ref 355 357 359 363 368 388 425 429 431 434 dcl 27 set ref 74 429 431 dcl 17 set ref 356 357 359 363 368 389 426 429 432 434 dcl 27 set ref 73 429 432 dcl 6 set ref 105 108 dcl 6 set ref 127 130 dcl 8 set ref 153 162 dcl 8 set ref 176 179 dcl 13 set ref 207 210 dcl 13 set ref 223 226 dcl 15 set ref 241 244 dcl 15 set ref 258 261 dcl 21 set ref 283 285 dcl 26 set ref 75 107 129 161 174 209 225 243 260 286 353 389 427 429 492 540 573 dcl 26 set ref 72 406 407 dcl 26 ref 1 41 61 66 95 112 165 247 264 328 401 409 413 417 419 544 588 609 dcl 438 set ref 447 448 dcl 26 set ref 24 438 439 410 412 dcl 498 ref 497 515 dcl 438 ref 437 447 dcl 26 set ref 106 128 160 177 208 224 242 259 284 446 493 499 539 574 dcl 6 set ref 95 96 98 100 275 dcl 6 set ref 117 120 130 dcl 8 set ref 151 152 162 dcl 8 set ref 168 170 179 dcl 13 set ref 198 206 210 dcl 13 set ref 213 216 226 dcl 15 set ref 233 234 244 dcl 15 set ref 249 252 261 dcl 21 set ref 274 275 275 277 285
12	000144	automatic	float dec(10)	
14	000331	automatic	float dec(10)	
longarea	000474	automatic	float dec(10)	
longwt	000477	automatic	float dec(10)	
longspac	000100	automatic	float dec(10)	
maxionspac		parameter	float dec(10)	
maxn		parameter	fixed bin(17,0)	
maxxvrspace		parameter	float dec(10)	
minionspac		parameter	float dec(10)	
minn		parameter	fixed bin(17,0)	
minwt		parameter	float dec(10)	
minxvrspace		parameter	float dec(10)	
modoot	000466	automatic	float dec(10)	
modocftemp	001144	automatic	float dec(10)	
modok	000471	automatic	float dec(10)	
modoktemp	001141	automatic	float dec(10)	
modval1	000141	automatic	float dec(10)	
modval2	000160	automatic	float dec(10)	
modval3	000276	automatic	float dec(10)	
modval4	000312	automatic	float dec(10)	
modval5	000350	automatic	float dec(10)	
modval6	000364	automatic	float dec(10)	
modval7	000417	automatic	float dec(10)	
modval8	000430	automatic	float dec(10)	
modval9	000629	automatic	float dec(10)	
n	001120	automatic	fixed bin(17,0)	
ntrknd	001127	automatic	fixed bin(17,0)	
opt		parameter	fixed bin(17,0)	
p	000114	automatic	float dec(10)	
pi	001121	automatic	fixed bin(17,0)	
s		parameter	float dec(10)	
s		parameter	float dec(10)	
scanlim	001115	automatic	fixed bin(17,0)	
sectod1	000114	automatic	float dec(10)	
sectod2	000147	automatic	float dec(10)	
sectod3	000171	automatic	float dec(10)	
sectod4	000307	automatic	float dec(10)	
sectod5	000323	automatic	float dec(10)	
sectod6	000353	automatic	float dec(10)	
sectod7	000372	automatic	float dec(10)	
sectod8	000422	automatic	float dec(10)	
sectod9	000415	automatic	float dec(10)	

sigcbjrdth	000601 automatic	float dec(10)	dcl 21 set ref 267 268 268 338 419
sigclt	000163 automatic	float dec(10)	dcl 8 set ref 134 136 136 140 143 143 150 152 167 170 301 541
signa_3	000444 automatic	float dec(10)	dcl 17 set ref 342 344 348 353
signa_ay	000447 automatic	float dec(10)	dcl 17 set ref 343 343 345 353 354
signa_ay2	000452 automatic	float dec(10)	dcl 17 set ref 344 350 350 354
signa_io	000455 automatic	float dec(10)	dcl 17 set ref 345 351 351 354
smo:st	001101 automatic	float dec(10)	dcl 25 set ref 105 127 159 176 207 223 241 258 243 445 466 514 534 536
smf	000147 automatic	float dec(10)	dcl 438 set ref 5 457 458 466
smf	000144 automatic	float dec(10)	dcl 498 set ref 514 524 526 534 536
smc	000152 automatic	float dec(10)	dcl 498 set ref 514 525 526 534 534
smr	parameter	float dec(10)	dcl 438 ref 437 458
smr	parameter	float dec(10)	dcl 498 ref 497 526 526
smreq	000430 automatic	float dec(10)	dcl 17 set ref 37 39 357 357 388 381 612 613
stans	001123 automatic	fixed bin(17,0)	dcl 26 set ref 45 44 53 57 61 62 62 63 63 67 415
sysprint	000016 constant	file	output stream dcl 24 set ref 26 38 77 45 138 130 140 162 179 184 210 226 244 261 296 384 393 394 434 579 580 581 584 585 588 591 594 597 600 601 606 609 612 613 491 492 538 572 573
t	parameter	float dec(10)	dcl 3 ref 1 3d 113
tt	000106 automatic	float dec(10)	dcl 498 set ref 512 517 520 524 533 553 555 556 556
tt	000106 automatic	float dec(10)	dcl 438 set ref 443 450 453 457 465 483 482 482
ttdest	001104 automatic	float dec(10)	dcl 25 set ref 100 101 122 123 154 155 172 202 203 218 219 236 237 254 279 445 465 514 533
tp	parameter	float dec(10)	dcl 438 ref 437 439 439 441 443 448 451 452 453 457 480 485
tp	parameter	float dec(10)	dcl 498 ref 497 509 509 510 512 515 518 519 520 524 559
tn	000100 automatic	float dec(10)	dcl 498 set ref 504 505 506 537 508 539 509 513 510 511 511 513 516 522 531 545 551 555 559 559 559 559 563 567 568 568
tn	000100 automatic	float dec(10)	dcl 438 set ref 439 441 444 444 455 463 469 474 475 485 487 487
tndest	001107 automatic	float dec(10)	dcl 25 set ref 103 125 157 174 205 221 239 256 281 445 463 514 531
n	000103 automatic	float dec(10)	dcl 438 set ref 442 449 452 453 455 455 455 457 462 469 469 469 471 471 475
n	000103 automatic	float dec(10)	dcl 498 set ref 500 501 502 503 503 504 511 513 516 519 520 522 522 522 524 536 537 537 545 545 545 547 547 550 563 563 563 564 564
ndest	001112 automatic	float dec(10)	dcl 25 set ref 100 104 122 126 154 154 175 202 206 218 222 236 240 257 282 445 462 514 530
neojep1	000136 automatic	float dec(10)	dcl 6 set ref 104 139 296 297 300 378 585
neojep2	000224 automatic	float dec(10)	dcl 8 set ref 126 130 544
neojep3	000273 automatic	float dec(10)	dcl 8 set ref 158 162 314 591
neojep4	000240 automatic	float dec(10)	dcl 3 set ref 175 179 594
neojep5	000345 automatic	float dec(10)	dcl 13 set ref 206 210 314 315 315 363 603
neojep6	000254 automatic	float dec(10)	dcl 8 set ref 222 226 606
neojep7	000414 automatic	float dec(10)	dcl 15 set ref 240 244 322 597
neojep8	000273 automatic	float dec(10)	dcl 8 set ref 257 261 600
neojep9	000037 automatic	float dec(10)	dcl 21 set ref 282 285 333 334 337 609
neoth1	000133 automatic	float dec(10)	dcl 6 set ref 103 108 295 378 383 535
neoth2	000221 automatic	float dec(10)	dcl 8 set ref 125 130 588
neoth3	000210 automatic	float dec(10)	dcl 8 set ref 157 162 305 591
neoth4	000235 automatic	float dec(10)	dcl 6 set ref 174 179 594
neoth5	000342 automatic	float dec(10)	dcl 13 set ref 205 210 313 363 368 603
neoth6	000251 automatic	float dec(10)	dcl 8 set ref 221 226 606
neoth7	000411 automatic	float dec(10)	dcl 15 set ref 239 244 323 597

widthh	000205	automatic	float	dec(10)
widthy	000034	automatic	float	dec(10)
weight	000562	automatic	float	dec(10)
widthan	000576	automatic	float	dec(10)
xvrs1	000502	automatic	float	dec(10)
xvrs2	000505	automatic	float	dec(10)
xvrs3	000513	automatic	float	dec(10)
xvrs4	000513	automatic	float	dec(10)
xvrs5	000516	automatic	float	dec(10)
xvrs6	000521	automatic	float	dec(10)
xvrs7	000524	automatic	float	dec(10)
xvrspace	000103	automatic	float	dec(10)

yl	000133	automatic	float	dec(10)
yl	000136	automatic	float	dec(10)
ym	000136	automatic	float	dec(10)
ym	000141	automatic	float	dec(10)
yp	000130	automatic	float	dec(10)
yp	000125	automatic	float	dec(10)
yn	000133	automatic	float	dec(10)
yn	000130	automatic	float	dec(10)

NAMES DECLARED BY DECLARE STATEMENT AND NEVER REFERENCED.

destopt	001125	automatic	fixed	bin(17,0)
ftemp	000433	automatic	float	dec(10)

NAMES DECLARED BY EXPLICIT CONTEXT.

barjdeso	001220	constant	entry	
bnos	004547	constant	label	
bottom	002320	constant	label	
bottom	005436	constant	label	
jack	004114	constant	label	
endips	013120	constant	label	
enjsm1	014236	constant	label	
enjsm2	015072	constant	label	
girder1	002724	constant	label	
girder2	003051	constant	label	
girder3	004442	constant	label	
girder4	005154	constant	label	
nyrod1	002545	constant	label	
nyrod3	004261	constant	label	
nulcalc	006124	constant	label	
nulism	007172	constant	label	
nulint	010006	constant	label	
interool	001261	constant	label	
loop1	013747	constant	label	
loop1	015227	constant	label	
loops	002036	constant	label	
onward	001427	constant	label	
onward1	002430	constant	label	
onward2	003311	constant	label	
onward3	004177	constant	label	
return	001617	constant	label	
sice	003205	constant	label	
smact1	013332	constant	entry	
smact2	014240	constant	entry	
smcalc	014707	constant	label	

dcl 8 set ref 256 261 600
dcl 21 set ref 281 285 332 609
dcl 15 set ref 420 421 422 434
dcl 19 set ref 413 414 415
dcl 19 set ref 401 402 416
dcl 19 set ref 403 416
dcl 19 set ref 404 415
dcl 19 set ref 405 416 417
dcl 19 set ref 406 416
dcl 19 set ref 407 416
dcl 19 set ref 415 416
dcl 3 set ref 71 72 77 95 95 96 96 97 117 119
150 151 151 167 168 168 168 194 213 214 232 233
233 249 251 274 276 389 401 413 434 435 406 407
411 415 419 424 492 573
dcl 498 set ref 520 521 522 522 522
dcl 438 set ref 453 454 455 455 455
dcl 498 set ref 521 522 524 525
dcl 438 set ref 454 455 457
dcl 438 set ref 451 454 455 455 455
dcl 498 set ref 518 521 522 522 522
dcl 438 set ref 452 454 455 455 455
dcl 498 set ref 519 521 522 522 522

dcl 26
dcl 17

external dcl 1 ref 1
dcl 230 ref 230
dcl 79 ref 79
dcl 264 ref 264
dcl 182 ref 182
dcl 611 ref 110 131 163 180 211 227 245 262 287
jy1 395 578 611
dcl 496 ref 467 496
dcl 577 ref 543 577
dcl 112 ref 112
dcl 165 ref 165
dcl 213 ref 213
dcl 247 ref 247
dcl 97 ref 97 376
dcl 199 ref 199 361
dcl 290 ref 133 229 290 366 371 381 386
dcl 342 ref 342
dcl 396 ref 357 396
dcl 31 ref 31
dcl 458 ref 458 494
dcl 526 ref 526 575
dcl 70 ref 59 64 70
dcl 37 ref 37
dcl 89 ref 89
dcl 144 ref 144
dcl 192 ref 192
dcl 41 ref 41
dcl 134 ref 114 134
internal dcl 437 ref 98 152 200 234 277 437
internal dcl 497 ref 120 170 216 252 497
dcl 515 ref 515 548 553 557 561 565 564

smcalc	013-52	constant	label	dcl 443 ref 443 472 478 483 498
stand1	002412	constant	label	dcl 88 ref 88
stand2	003273	constant	label	dcl 143 ref 143
stand3	004101	constant	label	dcl 191 ref 191
stand4	004737	constant	label	dcl 231 ref 231

NAMES DECLARED BY CONTEXT OR IMPLICATION.

ceil	builtin function	internal ref 33 88 113 143 191 231 266 268 270 338 474 503 511 513
floor	builtin function	internal ref 72
max	builtin function	internal ref 275 444 503 509 510 511
min	builtin function	internal ref 97 119 150 167 199 214 232 251 276 508 513
sqr	builtin function	internal ref 113 230 412

STORAGE REQUIREMENTS FOR THIS PROGRAM.

	Object	Text	Link	Symbol	Defs	Static
Start	0	0	16044	16066	15753	16094
Length	24042	15753	22	7737	71	2

BLOCK NAME	STACK SIZE	TYPE	WHY NONQUICK/WHO SHARES STACK FRAME
barides6	914	external procedure	is an external procedure.
smact11	298	internal procedure	contains a format statement, and uses I/O statements.
smact12	301	internal procedure	contains a format statement, and uses I/O statements.

THE FOLLOWING EXTERNAL OPERATORS ARE USED BY THIS PROGRAM.

fxl_to_f12	r_l_a	r_l_a	r_e_a	r_l_e_a	r_g_e_a
call_ext_out	call_int_this	return	ext_entry	int_entry	put_end
stream_to	put_data_eis	put_list_eis	real_to_real_rd	dsqrt	cell_cec
floor_dac	put_field	put_field_cnk	put_control		

THE FOLLOWING EXTERNAL ENTRIES ARE CALLED BY THIS PROGRAM.

decimal_exp_

THE FOLLOWING EXTERNAL VARIABLES ARE USED BY THIS PROGRAM.

sysprint
sysprint.fso

LINE	LOC	LINE	LOC	LINE	LOC	LINE	LOC	LINE	LOC	LINE	LOC
1	001204	28	001240	29	001252	30	001254	31	001261	33	001276
37	001427	38	001446	41	001617	43	001623	44	001630	45	001643
48	001657	49	001672	51	001674	52	001707	53	001722	55	001724
57	001743	59	001745	61	001746	62	001757	63	001761	64	002016
67	002021	68	002022	70	002036	71	002111	72	002165	73	002213
75	002221	76	002230	77	002231	79	002320	81	002346	83	002372
86	002411	88	002412	89	002430	91	002441	93	002470	95	002501
97	002545	98	002606	99	002620	100	002625	101	002631	102	002635
104	002646	105	002652	106	002656	107	002661	108	002664	110	002723
113	002730	114	002774	116	002775	117	003009	118	003020	119	003022
121	003076	122	003102	123	003106	124	003112	125	003117	126	003123
128	003134	129	003137	130	003142	131	003201	133	003202	134	003205
139	003253	140	003257	141	003272	143	003273	144	003311	146	003344
150	003450	151	003511	152	003531	153	003543	154	003547	155	003553
157	003564	158	003571	159	003576	160	003603	161	003606	162	003611
165	003651	166	003676	167	003710	168	003756	169	003776	170	004000
172	004020	173	004025	174	004031	175	004035	176	004041	177	004046
179	004054	180	004113	182	004114	184	004131	186	004141	188	004145
										189	004160

B.3 DICTIONARY OF VARIABLES USED IN THE BARGE DESIGN MODEL

DEFINITION OF VARIABLES IN THE SUBPROGRAM "bargdes"

<u>Variable</u>	<u>Definition</u> (units)
abest:	Optimum frame area (includes web area and flange area) (sq in)
absfact:	File containing data for coeff array
area:	Total longitudinal sectional area of midship crossection (includes longitudinal frames and plates) (sq in)
area1:	Sectional area of bottom longitudinal frame (sq in)
area2:	Sectional area of bottom transverse girder (sq in)
area3:	Sectional area of side longitudinal frame (sq in)
area4:	Sectional area of side transverse girder (sq in)
area5:	Sectional area of deck longitudinal frame (sq in)
area6:	Sectional area of deck transverse girder (sq in)
area7:	Sectional area of bulkhead longitudinal frame (sq in)
area8:	Sectional area of bulkhead transverse girder (sq in)
area9:	Sectional area of double bottom longitudinal frame (sq in)
astan:	Stanchion cross sectional area (sq in)
b:	Molded breadth of barge (ft)
bestlonspac:	Longitudinal frame spacing for minimum hull weight (ft)

bestmodbot: Bottom section modulus for minimum hull weight
 bestmoddk: Deck section modulus for minimum hull weight
 bestn: Value of n for minimum hull weight
 bestxvrspac: Transverse girder spacing for minimum hull weight
 bhdplt: Thickness of bulkhead plate (in)
 botplt: Thickness of bottom plate (in)
 cb: Block coefficient of barge
 cendbgirdth: Thickness of double bottom center girder (in)
 centroid: Centroid of longitudinal midship crossection area
 coeff: Array containing ABS factors used in section modulus calculation
 countr: Incremental counter used in do loops
 cycle: Index indicating which type of output is desired. If 0, a complete summary of hull scantlings is provided. If 1, only the hull weight and hull section modulus is provided for each case of transverse and longitudinal spacing and value of n. If 2, only the minimum hull weight for all the iterations is provided.
 d: Molded depth of barge (ft)
 dbdep: Depth of double bottom (in)
 dbplt: Thickness of double bottom plate (in)
 deep1: Depth of bottom longitudinal frame (in)
 deep2: Depth of bottom transverse girder (in)
 deep3: Depth of side longitudinal frame (in)
 deep5: Depth of deck longitudinal frame (in)
 deep6: Depth of deck transverse girder (in)
 deep7: Depth of bulkhead longitudinal frame (in)
 dimen: See attached sheet

dkplt: Thickness of deck plate (in)
 effplate1: Effective plate width to be used in section modulus calculation of bottom longitudinals (ft)
 effplate2: Effective plate width to be used in section modulus calculation of bottom transverses (ft)
 effplate3: Effective plate width to be used in section modulus calculation of side longitudinals (ft)
 effplate4: Effective plate width to be used in section modulus calculation of side transverses (ft)
 effplate5: Effective plate width to be used in section modulus calculation of deck longitudinals (ft)
 effplate6: Effective plate width to be used in section modulus calculation of deck transverses (ft)
 effplate7: Effective plate width to be used in section modulus calculation of bulkhead longitudinals (ft)
 effplate8: Effective plate width to be used in section modulus calculation of bulkhead transverses (ft)
 effplate9: Effective plate width to be used in section modulus calculation of double bottom longitudinals (ft)
 fact: ABS factor used in hull section modulus calculations
 fbest: Optimum flange depth of member (in)
 flanth1: Thickness of flange of bottom longitudinal frame (in)
 flanth2: Thickness of flange of bottom transverse girder (in)
 flanth3: Thickness of flange of side longitudinal frame (in)
 flanth4: Thickness of flange of side transverse girder (in)
 flanth5: Thickness of flange of deck longitudinal frame (in)

flanth6:	Thickness of flange of deck transverse girder (in)
flanth7:	Thickness of flange of bulkhead longitudinal frame (in)
flanth8:	Thickness of flange of bulkhead transverse girder (in)
flanth9:	Thickness of flange of double bottom longitudinal frame (in)
flanwi1:	Depth of flange of bottom longitudinal frame (in)
flanwi2:	Depth of flange of bottom transverse girder (in)
flanwi3:	Depth of flange of side longitudinal frame (in)
flanwi4:	Depth of flange of side transverse girder (in)
flanwi5:	Depth of flange of deck longitudinal frame (in)
flanwi6:	Depth of flange of deck transverse girder (in)
flanwi7:	Depth of flange of bulkhead longitudinal frame (in)
flanwi8:	Depth of flange of bulkhead transverse girder (in)
flanwi9:	Depth of flange of double bottom longitudinal frame (in)
ftwt:	Weight per longitudinal foot of midship section (lb/ft)
gtype:	Index used to indicate in smact2 type of transverse girder for which section modulus is being calculated (1--bottom; 2--side; 3--deck; 4--bulkhead).
h1:	Head to be used for bottom longitudinal frame/transverse girder section modulus calculation (ft)
h2:	Head to be used for side and longitudinal bulkhead longitudinal frame/transverse girder section modulus calculation (ft)

h3: Head to be used for deck longitudinal frame/
 transverse girder section modulus calculation
 (ft)

h5: Head to be used for double bottom longitudinal
 frame section modulus calculation (ft)

i: Index used in do loops

ihc: Index used to indicate if deck/bottom
 longitudinal and transverse section modulus cal-
 culation is being made for the first time (ihc=0)
 or is called for by statements in hull section
 modulus calculation (ihc=1) which results from
 bottom/deck plate increases

incrlonspac: Longitudinal spacing increment used in do loops

incrsvrspac: Transverse spacing increment used in do loops

inertia: Moment of inertia of longitudinal midsection area

k1, k2, k3: Coefficients used in calculation of astan

1: Length between perpendiculars of barge (ft)

12: Unsupported span of deck and bottom transverse
 girders (ft)

14: Unsupported span of side and bulkhead transverse
 girders (ft)

longarea: Sectional area of longitudinals and plates in
 midship cross-section (sq in)

longwt: Weight per foot for longitudinals and plates in
 midship cross-section (lbs/foot)

lonspac: Spacing between longitudinals (ft)

maxlonspac: Maximum longitudinal spacing to be considered in
 the run (ft)

maxn: Upper bound of iterative loop for n

maxxvrspac: Maximum transverse spacing to be considered in
 the run (ft)

minn: Lower bound of iterative loop for n

minlonspac: Minimum longitudinal spacing to be considered in

the run (ft)

minwt: The minimum hull weight calculated for all variations in longitudinal and transverse spacing and value of n (LT)

minxvrspac: Minimum transverse spacing to be considered in the run (ft)

modbot: Calculated section modulus to bottom plate (hull girder section modulus)

modbottemp: Temporary storage variable for bottom section modulus used in reducing iterations of n values

moddk: Calculated section modulus to deck plate (hull girder section modulus)

moddktemp: Temporary storage variable for deck section modulus used in reducing iterations of n values

modval1: Calculated section modulus of bottom longitudinal frame

modval2: Calculated section modulus of bottom transverse girder

modval3: Calculated section modulus of side longitudinal frame

modval4: Calculated section modulus of side transverse girder

modval5: Calculated section modulus of deck longitudinal frame

modval6: Calculated section modulus of deck transverse girder

modval7: Calculated section modulus of bulkhead longitudinal frame

modval8: Calculated section modulus of bulkhead transverse girder

modval9: Calculated section modulus of double bottom longitudinal frame

ntrbkcd: Number of transverses between transverse bulkheads

n: Index used to indicate how many 1/16's inch deckplate may exceed bottom plate in hull section modulus calculations

opt: Index used to indicate which longitudinal section option is being considered. If opt=0 then deck/bottom transverse span is limited to less than 15'. If opt=1, then transverse span is increased by reducing number of stanchions by two in transverse cross-section. If opt=2 then two longitudinal bulkheads are used, with no stanchions. If opt=3 then opt=0 applies but with double bottom.

scanlim: Switch to indicate scantlings limit reached in smact1 or smact2 (scanlim = 1 if scantling limits exceeded to obtain required section modulus).

secmod1: Required section modulus of bottom longitudinal frame

secmod2: Required section modulus of bottom transverse girder

secmod3: Required section modulus of side longitudinal frame

secmod4: Required section modulus of side transverse girder

secmod5: Required section modulus of deck longitudinal frame

secmod6: Required section modulus of deck transverse girder

secmod7: Required section modulus of bulkhead longitudinal frame

secmod8: Required section modulus of bulkhead transverse girder

secmod9: Required section modulus of double bottom longitudinal frame

siddbgirdth: Thickness of double bottom side girders (in)

sidplt: Thickness of side shell plating (in)

sigm_a: Sum of longitudinal member (longitudinal frames and plates) cross-sectional areas (sq in)

sigma_ay:	Sum of first vertical moments about baseline (ft-in ²)
sigma_ay2:	Sum of second vertical moments about baseline (ft-in) ²
sigma_io:	Sum of longitudinal member individual inertias about their respective center of gravity (in ⁴)
smbest:	Optimum section modulus of member
smreq:	Required hull girder section modulus
stans:	The number of stanchions used in supporting a deck/bottom transverse girder
t:	Loaded draft of barge (ft)
tfbest:	Optimum flange thickness of member (in)
twbest:	Optimum web thickness of member (in)
wbest:	Optimum web depth of member (in)
webdep1:	Depth of the web of bottom longitudinal frame (in)
webdep2:	Depth of the web of bottom transverse girder (in)
webdep3:	Depth of the web of side longitudinal frame (in)
webdep4:	Depth of the web of side transverse girder (in)
webdep5:	Depth of the web of deck longitudinal frame (in)
webdep6:	Depth of the web of deck transverse girder (in)
webdep7:	Depth of the web of bulkhead longitudinal frame (in)
webdep8:	depth of the web of bulkhead transverse girder (in)
webdep9:	Depth of the web of double bottom longitudinal frame (in)
webth1:	Thickness of the web of bottom longitudinal frame (in)

webth2:	Thickness of the web of bottom transverse girder (in)
webth3:	Thickness of the web of side longitudinal frame (in)
webth4:	Thickness of the web of side transverse girder (in)
webth5:	Thickness of the web of deck longitudinal frame (in)
webth6:	Thickness of the web of deck transverse girder (in)
webth7:	Thickness of the web of bulkhead longitudinal frame (in)
webth8:	Thickness of the web of bulkhead transverse girder (in)
webth9:	Thickness of the web of double bottom longitudinal frame (in)
weight:	Estimated hull steel weight of barge (LT)
wtstan:	Weight per stanchion (lbs)
xvrs1:	Weight per longitudinal foot for bottom transverse girders (lbs/ft)
xvrs2:	Weight per longitudinal foot for side transverse girders (lbs/ft)
xvrs3:	Weight per longitudinal foot for deck transverse girders (lbs/ft)
xvrs4:	Weight per longitudinal foot for bulkhead transverse girders (lbs/ft)
xvrs5:	Weight per longitudinal foot for transverse bulkheads (lbs/ft)
xvrs6:	Weight per longitudinal foot for transverse bulkhead longitudinals (lbs/ft)
xvrs7:	Weight per longitudinal foot for stanchions (lbs/ft)
xvrspac:	Spacing between transverse girders (ft)

DEFINITION OF THE ARRAY DIMEN

Dimensions	Member	Definition
(1,1)	Bottom Plate	$12xb$
(1,2)	" "	$botplt$
(1,3)	" "	$\frac{botplt}{24}$
(2,1)	Bottom Longitudinal Web	$\frac{webthlxb}{lonspac}$
(2,2)	" " "	$webdepl$
(2,3)	" " "	$\frac{botplt+webdepl}{2}$ 12
(3,1)	Bottom Longitudinal Flange	$\frac{flanwil \times b}{lonspac}$
(3,2)	" " "	$flanthl$
(3,3)	" " "	$\frac{botplt+webdepl+flanthl}{2}$ 12
(4,1)	Side Plate	$2 \times sidplt$
(4,2)	" "	$12 \times d$
(4,3)	" "	$\frac{d}{2}$
(5,1)	Side Longitudinal Web	$\frac{webdep3x2xd}{lonspac}$
(5,2)	" " "	$webth3$
(5,3)	" " "	$\frac{d}{2}$
(6,1)	Side Longitudinal Flange	$\frac{flanth3x2xd}{lonspac}$
(6,2)	" " "	$flanwi3$
(6,3)	" " "	$\frac{d}{2}$

Dimensions	Member	Definition
(7,1)	Deck Plate	$12xb$
(7,2)	" "	$dkplt$
(7,3)	" "	$d - \frac{dkplt}{24}$
(8,1)	Deck Longitudinal	$\frac{webth5xb}{lonspac}$
(8,2)	" "	$webdep5$
(8,3)	" "	$d - \frac{(dkplt + webdep5)}{2}$ 12
(9,1)	Deck Longitudinal Flange	$\frac{flanwi5xb}{longspac}$
(9,2)	" "	$flanth5$
(9,3)	" "	$d - \frac{dkplt + webdep5 + flanth5}{2}$ 12
(10,1)	Bulkhead Plate	$bhdplt$
(10,2)	" "	$12xd - dbdep$
(10,3)	" "	$\frac{d}{2} + \frac{dbdep}{24}$
(11,1)	Bulkhead Longitudinal Web	$\frac{webdep7x(d - dbdep)}{12}$ $lonspac$
(11,2)	" "	$webth7$
(11,3)	" "	$\frac{d}{2} + \frac{dbdep}{24}$
(12,1)	Bulkhead Longitudinal Flange	$\frac{flanth7xd}{lonspac}$
(12,2)	" "	$flanwi7$
(12,3)	" "	$\frac{d}{2} + \frac{dbdep}{24}$

Dimensions	Member	Definition
(13,1)	Double Bottom Plate	$12 \times b$
(13,2)	" " "	$dbplt$
(13,3)	" " "	$\frac{botplt+dbdep+\underline{dbdplt}}{2}$ $\underline{\hspace{1cm}}12$
(14,1)	Double Bottom Longitudinal Web	$\frac{webth9xb}{lonspac}$
(14,2)	" " "	$webdep9$
(14,3)	" " "	$\frac{botplt+dbdep-\underline{web9}}{2}$ $\underline{\hspace{1cm}}12$
(15,1)	Double Bottom Longitudinal Flange	$\frac{flanwi9xb}{lonspac}$
(15,2)	" " "	$flanth9$
(15,3)	" " "	$\frac{botplt+dbdep-webdep9-\underline{flanth9}}{2}$ $\underline{\hspace{1cm}}12$
(16,1)	Double Bottom Center And Side Girders	$2xsiddbgirdthx(\text{ceil}(\frac{b}{30})-1)$ $+ cendbgirdth$
(16,2)	" " "	$dbdep$
(16,3)	" " "	$\frac{botplt + \underline{dbdep}}{2}$ $\underline{\hspace{1cm}}12$

dimen (i,1) ← horizontal dimension of member i

dimen (i,2) ← vertical dimension of member i

dimen (i,3) ← vertical distance of center of area of member i
from baseline (bottom of bottom plate)

DEFINITION OF VARIABLES USED IN
SUBROUTINES "smact1" AND "smact2"

af: Cross sectional area of flange (sq in)

am: Cross sectional area of member--flange and web (sq in)

ap: Cross section area of plate that is considered effective
in section modulus calculation (sq in)

aw: Cross sectional area of web (sq in)

dep: Subroutine input parameter used to specify longitudinal
depth in associated transverse girder (in)

f: Flange depth (in)

im: Moment of inertia of member

p: Longitudinal spacing (in)

s: Subroutine input parameter used to specify effective
width of plate (ft)

smf: Section modulus of member to the flange

smp: Section modulus of member to the plate

smr: Subroutine input parameter used to specify required sec-
tion modulus of member

tf: Flange thickness (in))

tp: Subroutine input parameter used to specify plate thick-
ness (in)

tw: Web thickness (in)

w: Web depth (in)

yf: Vertical distance from plate outer surface to flange cen-
ter of gravity (in)

ym: Vertical distance from plate outer surface to member cen-
ter of gravity (in)

yp: Vertical distance from plate outer surface to plate cen-
ter of gravity (in)

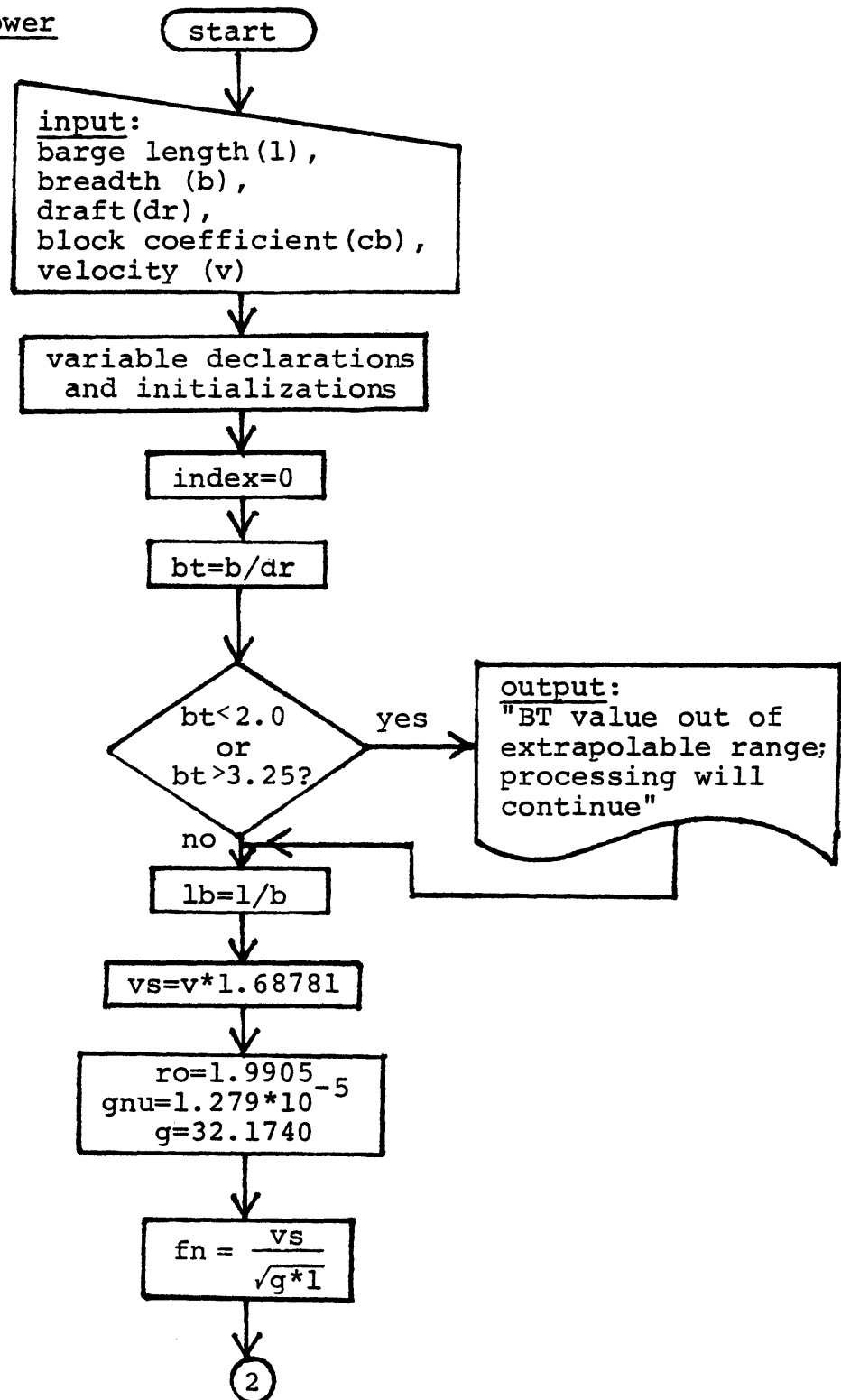
yw: Vertical distance from plate outer surface to web center
of gravity (in)

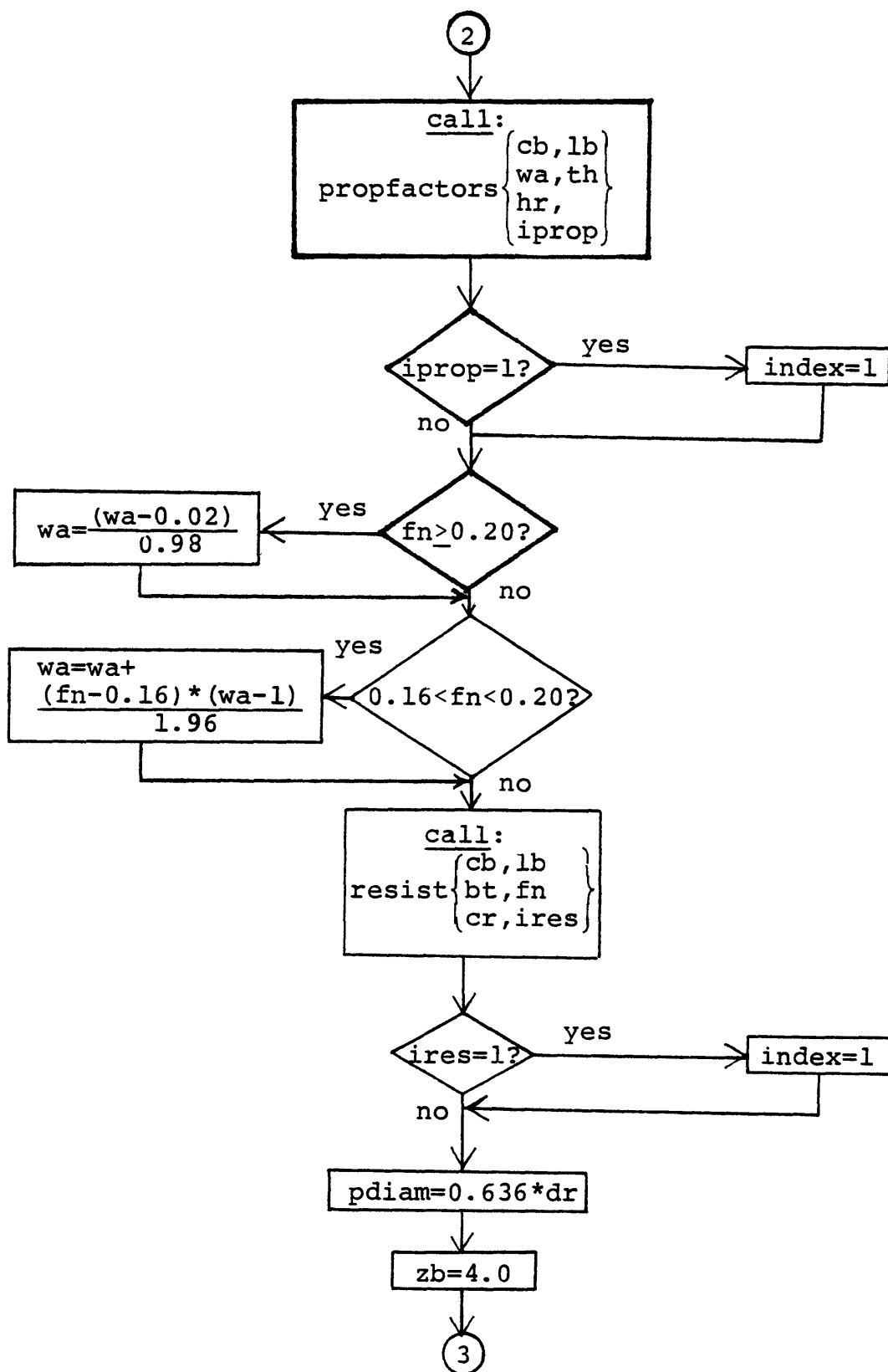
APPENDIX C

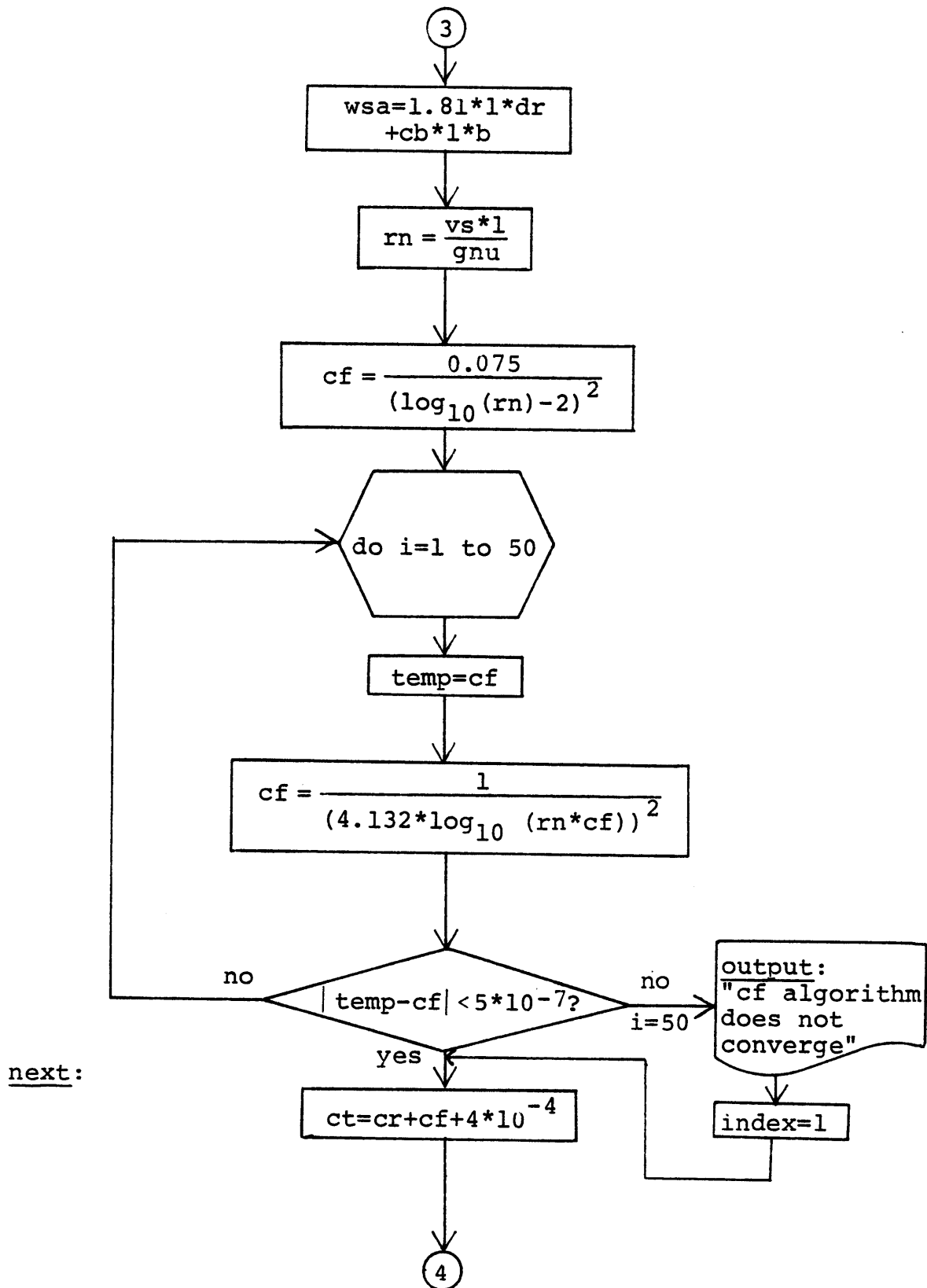
DOCUMENTATION FOR THE BARGE POWERING MODEL

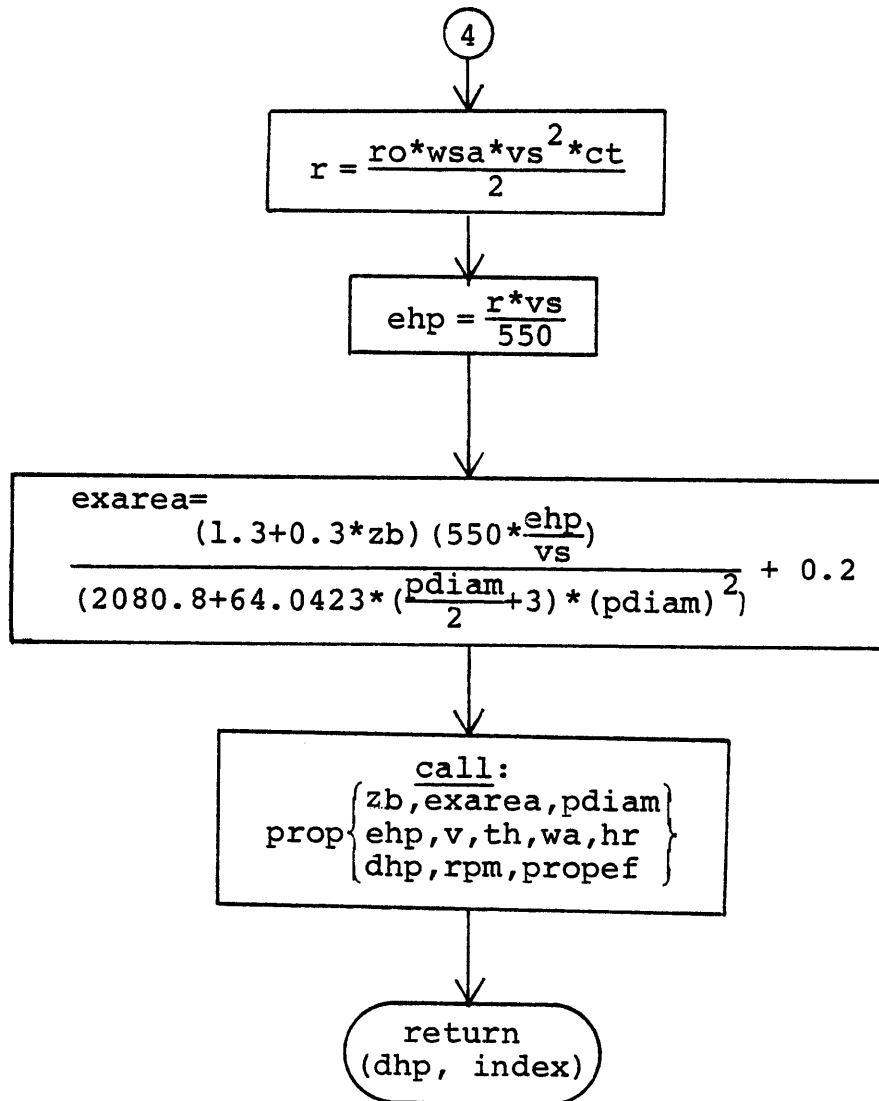
C.1 FLOWCHART OF THE BARGE POWERING MODEL

Subprogram: Power

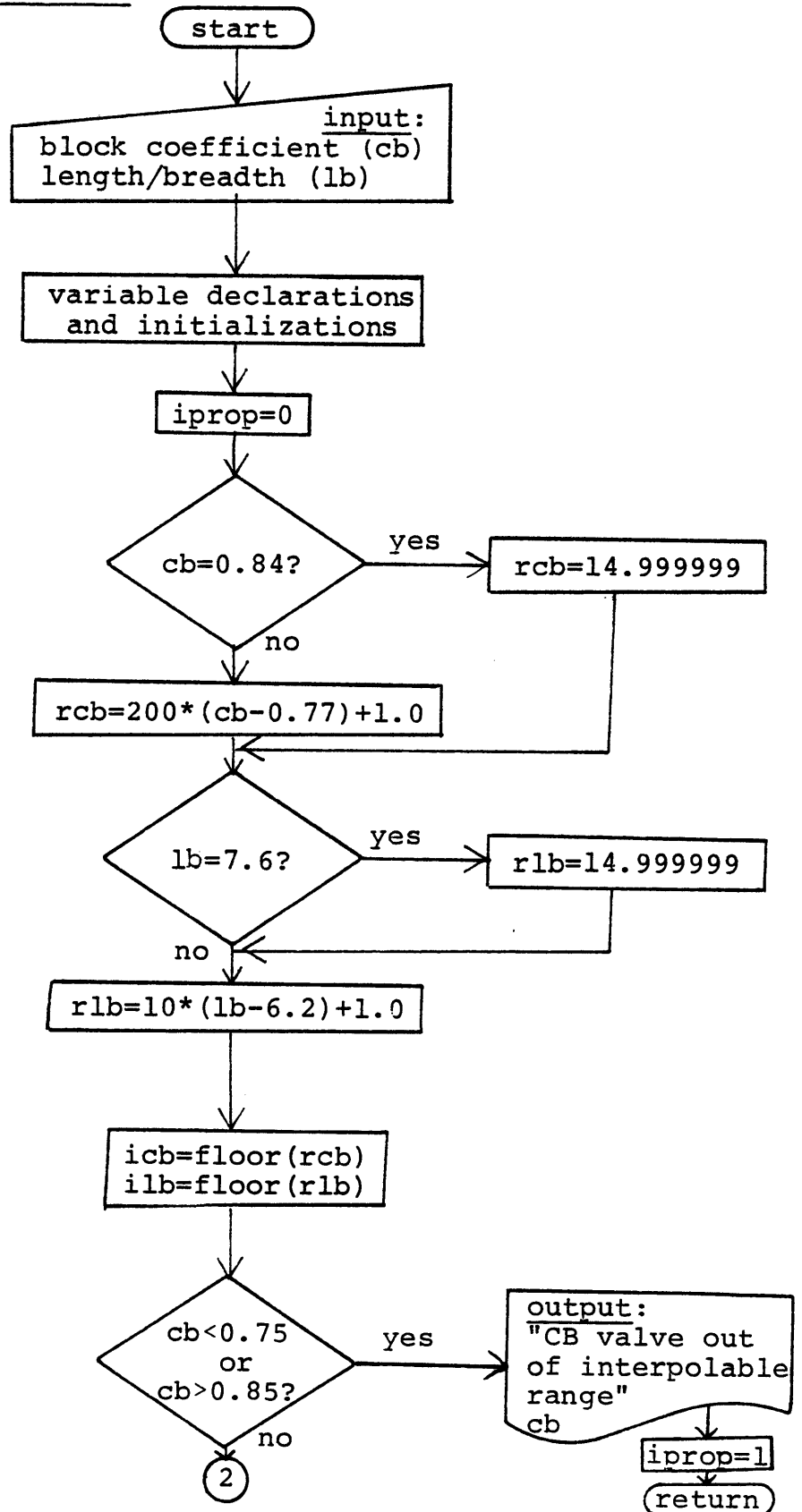


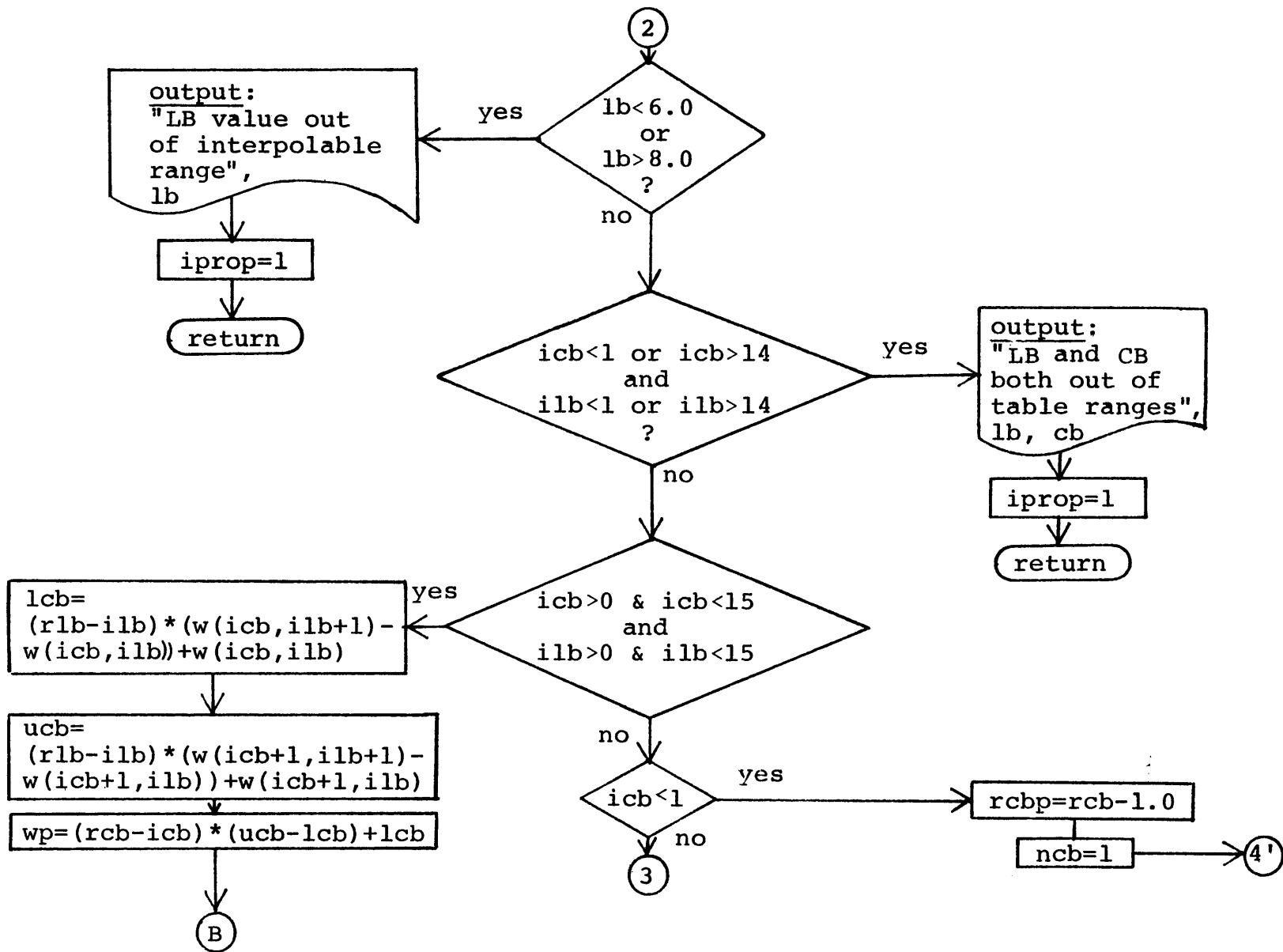


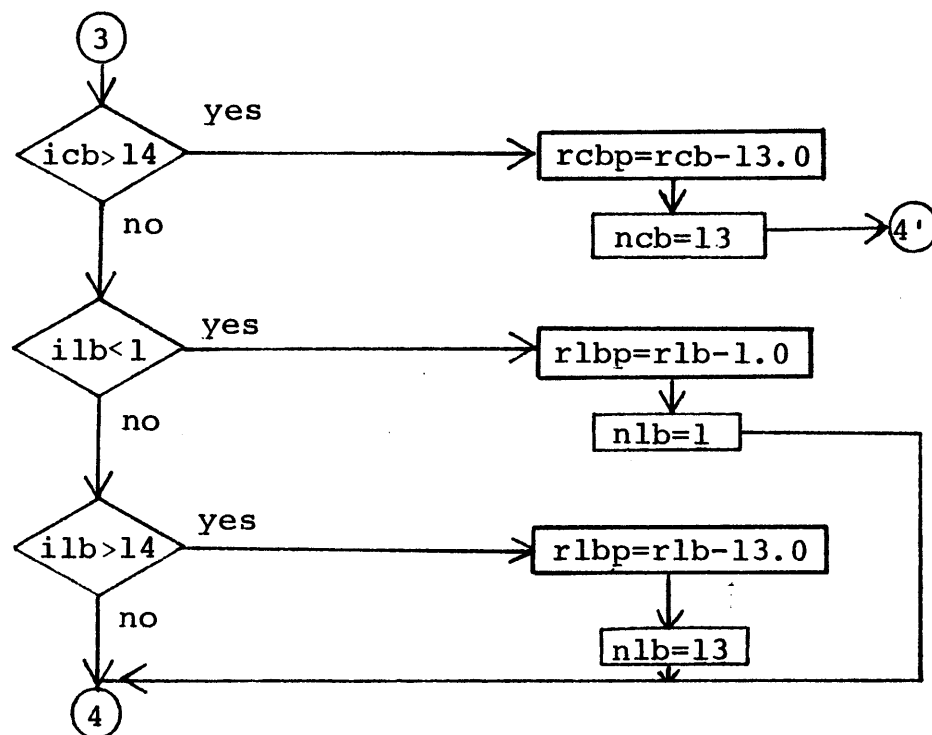
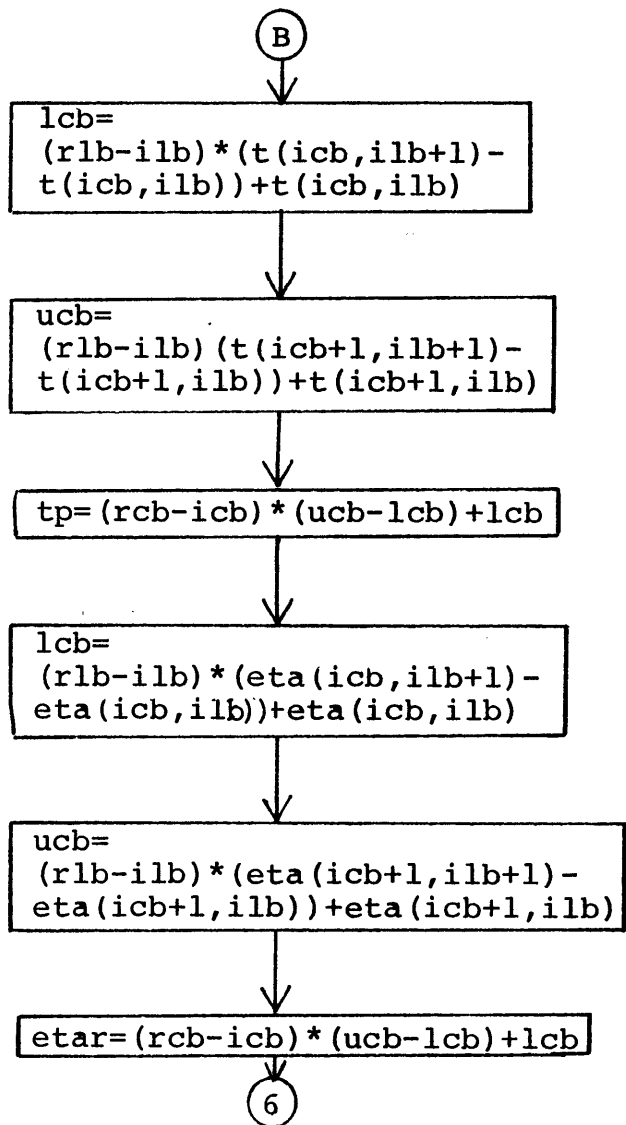


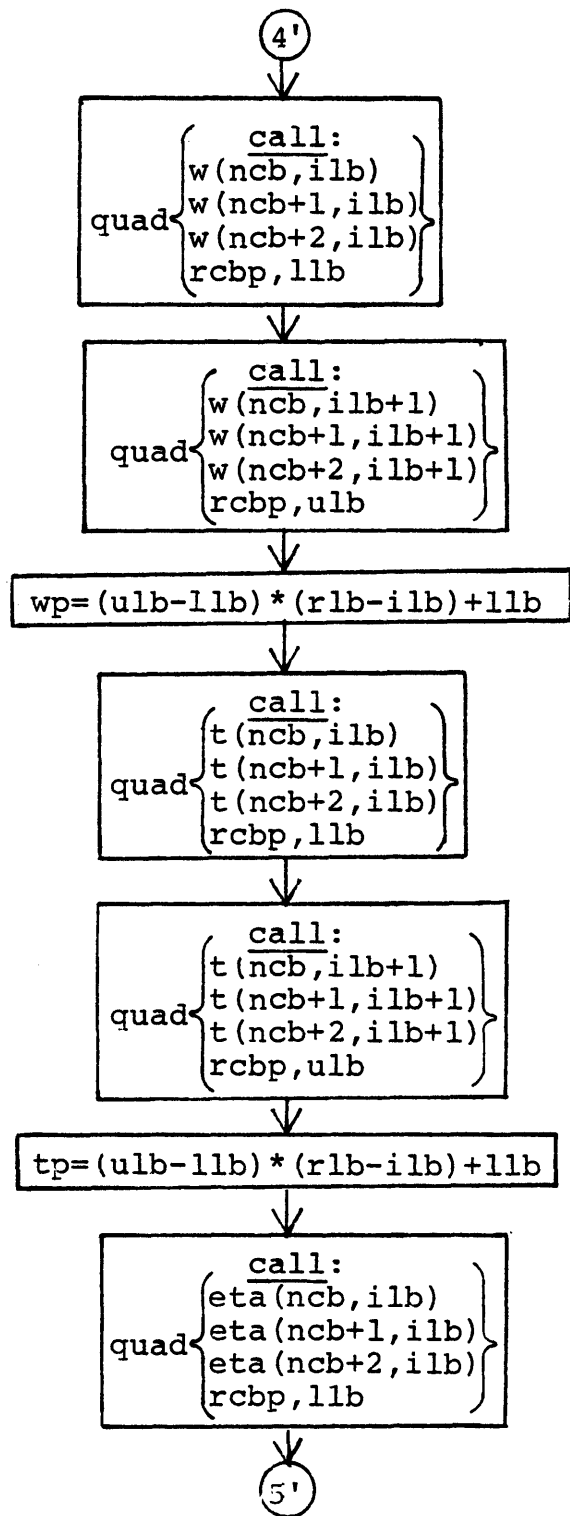
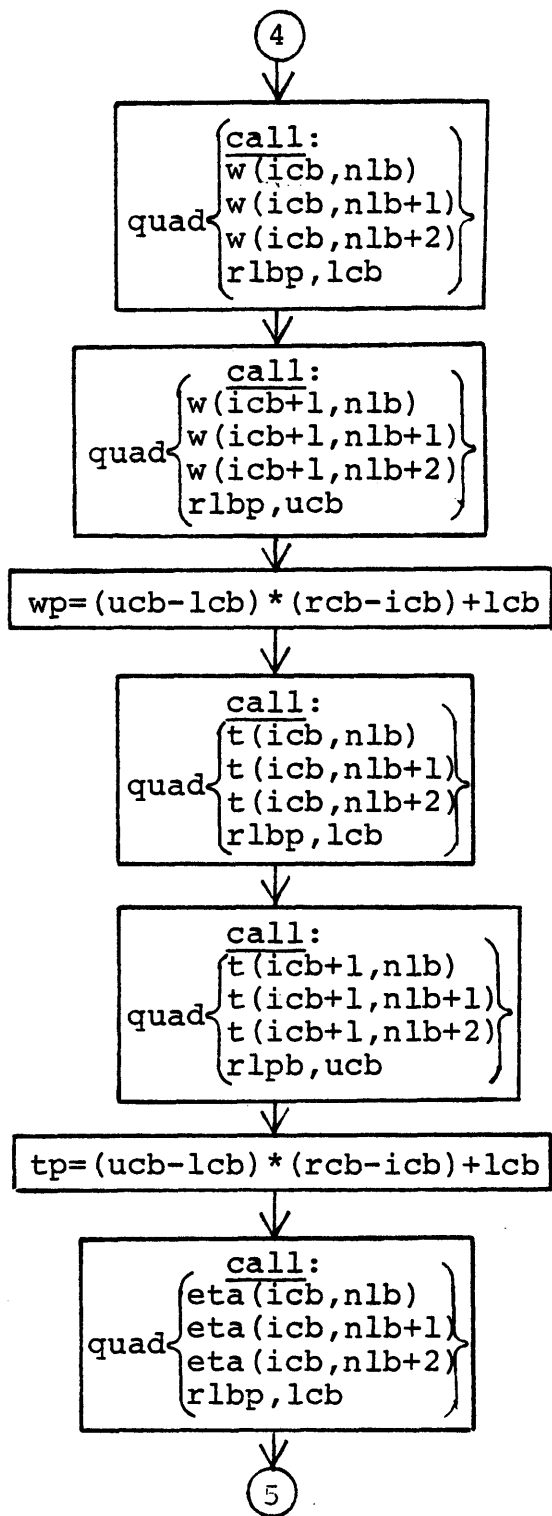


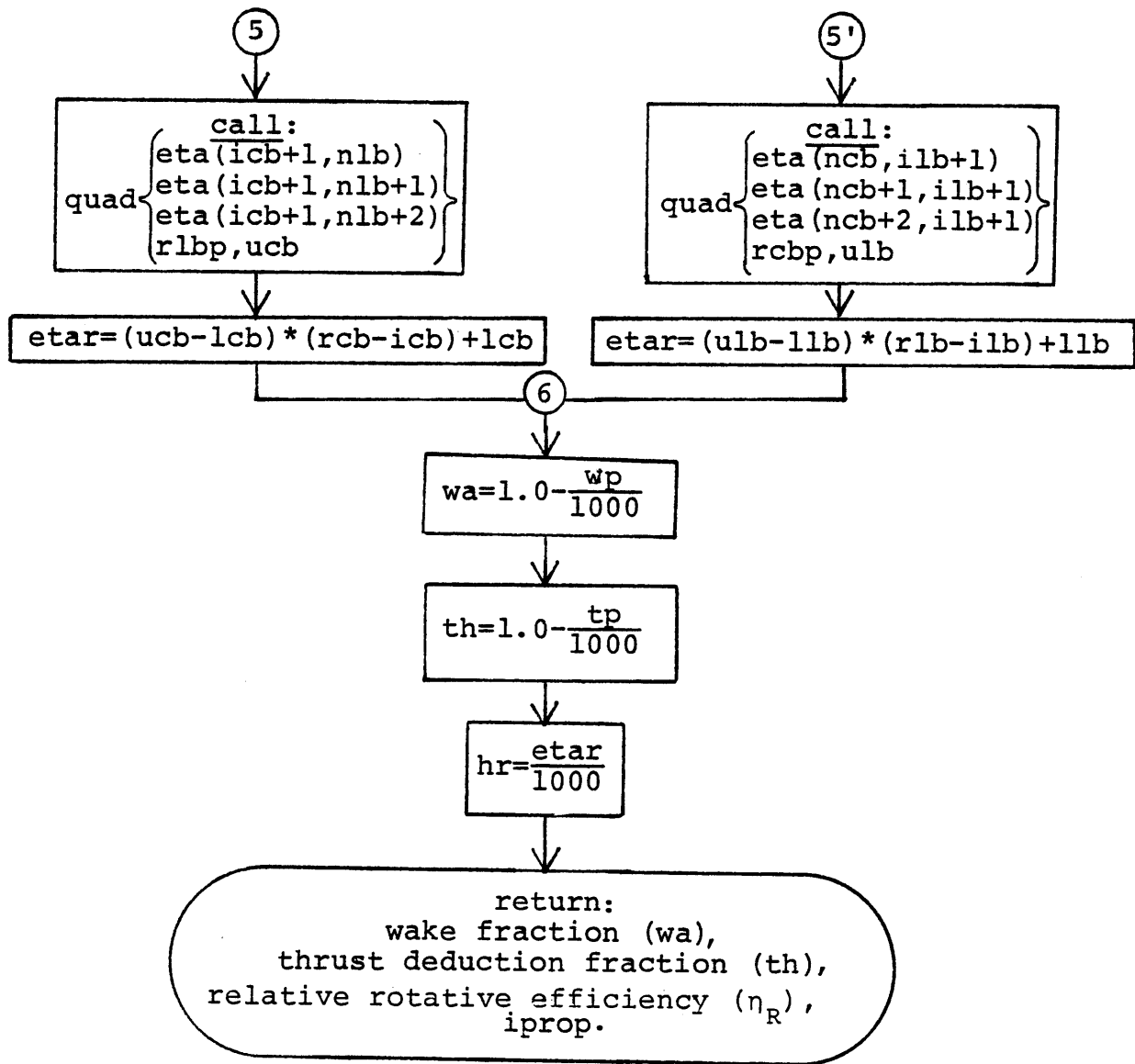
Subroutine Propfactors



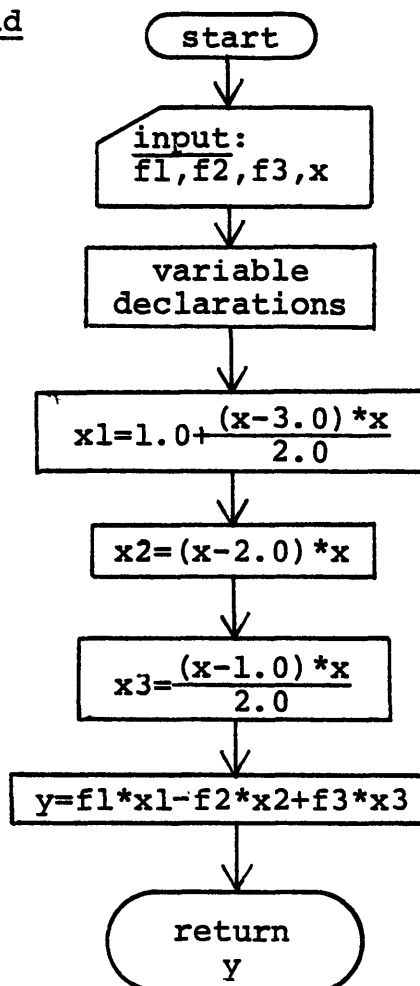








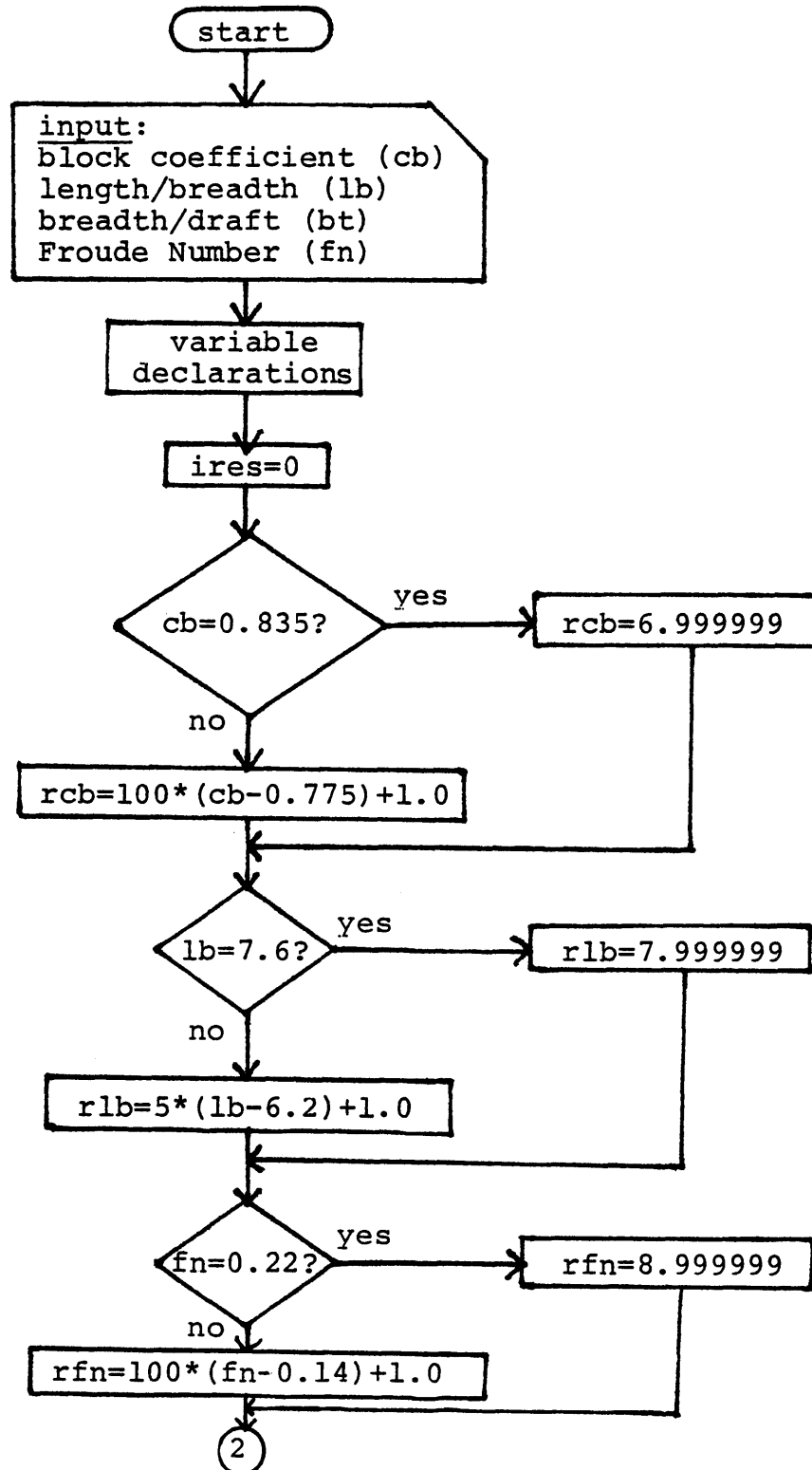
Subroutine: Quad

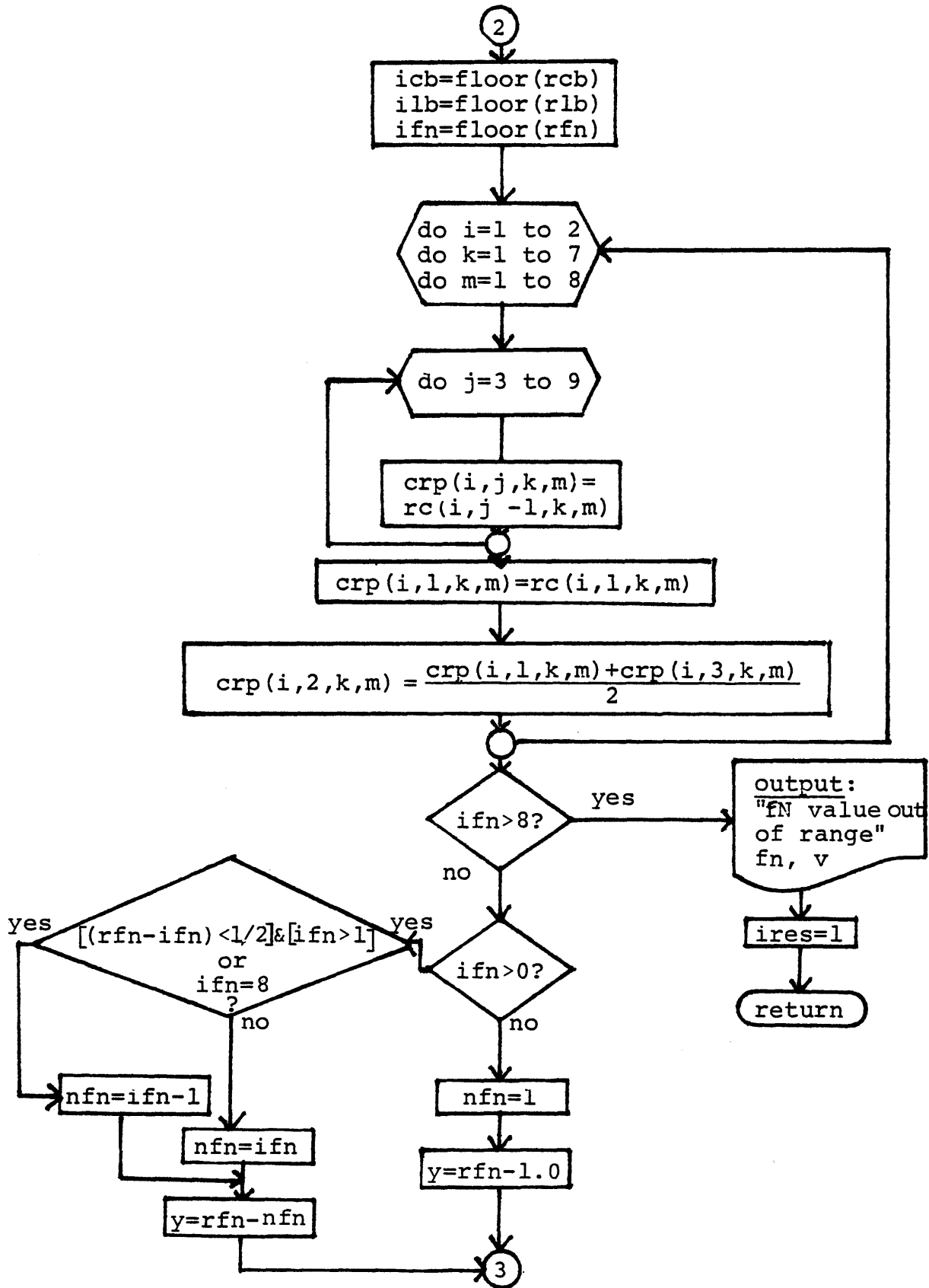


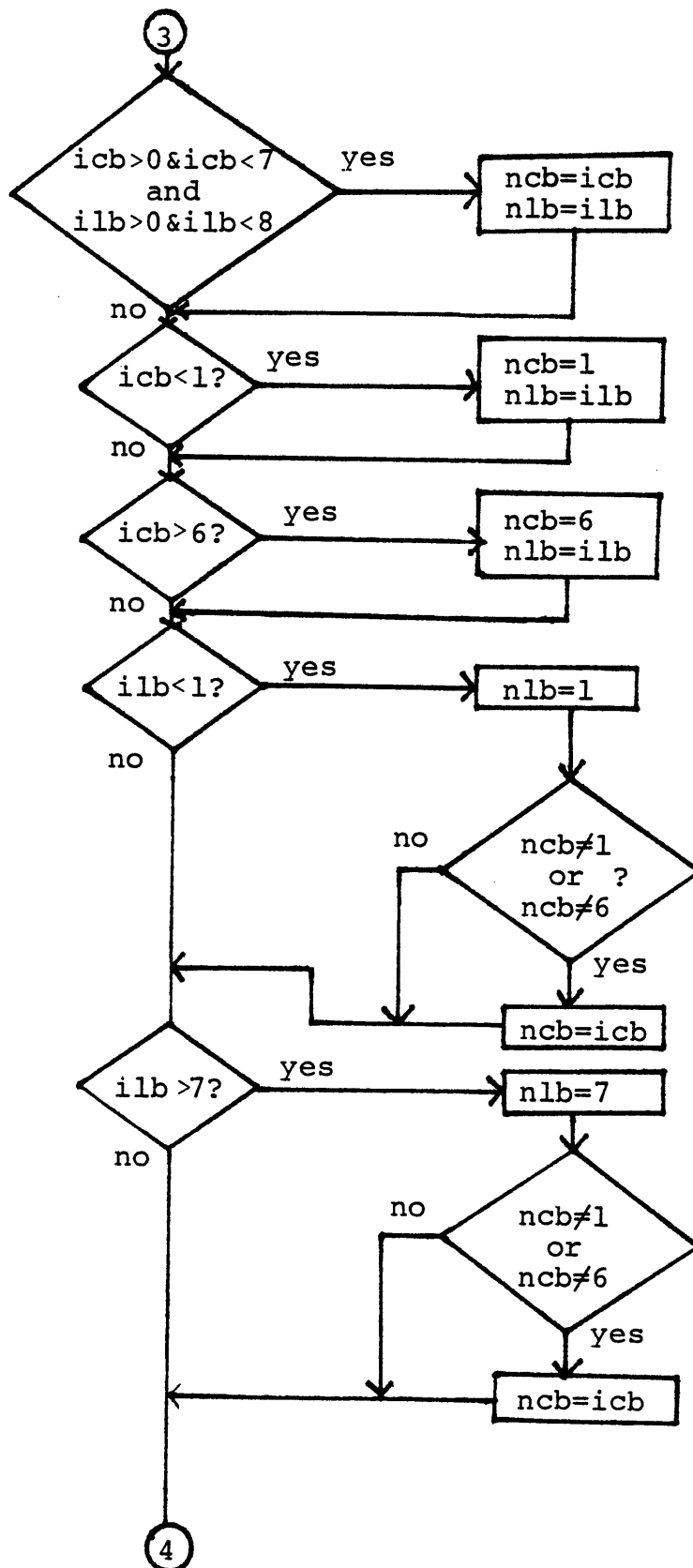
Definition of the variables involved in "quad", a subroutine for quadratic interpolation and extrapolation.

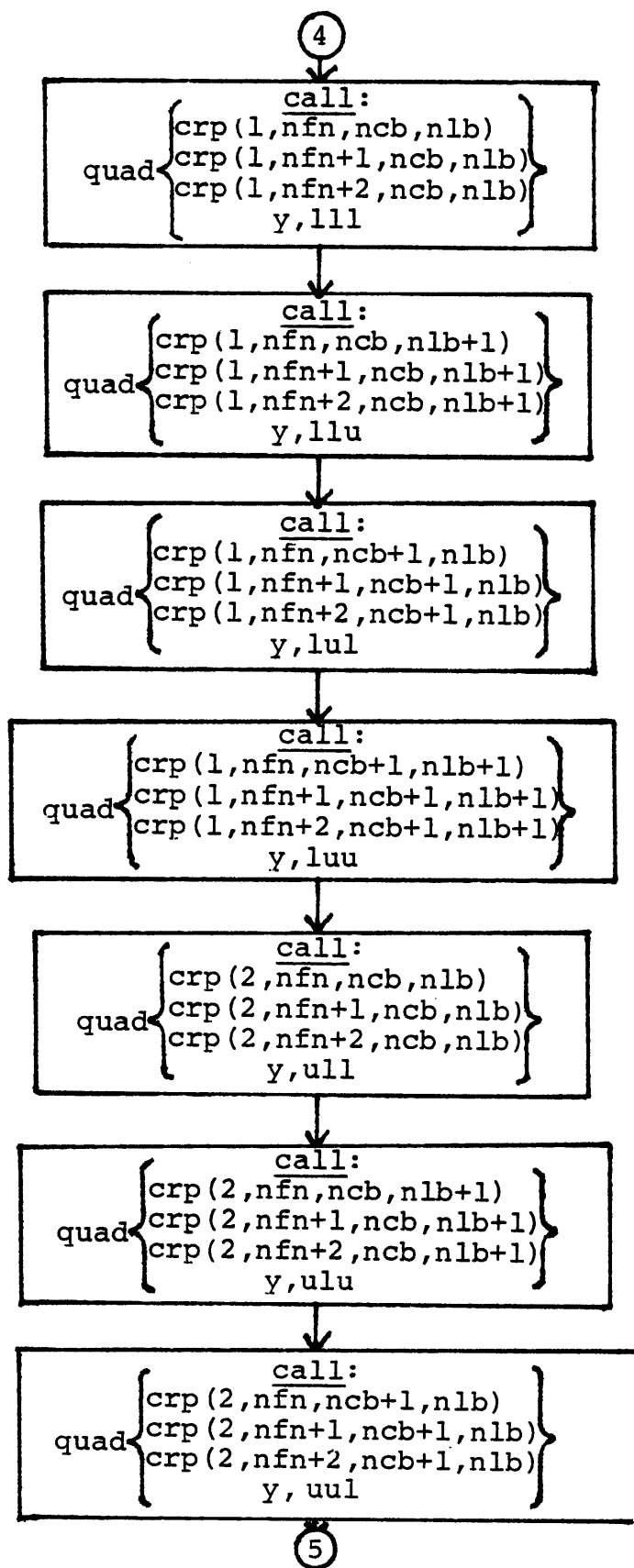
- f1: value of ordinate associated with an abscissa assumed to be 0.
- f2: value of ordinate associated with an abscissa assumed to be 1.
- f3: value of ordinate associated with an abscissa assumed to be 2.
- x: value of abscissa for which quadratic interpolation/extrapolation is desired.
- y: value of ordinate associated with abscissa x.

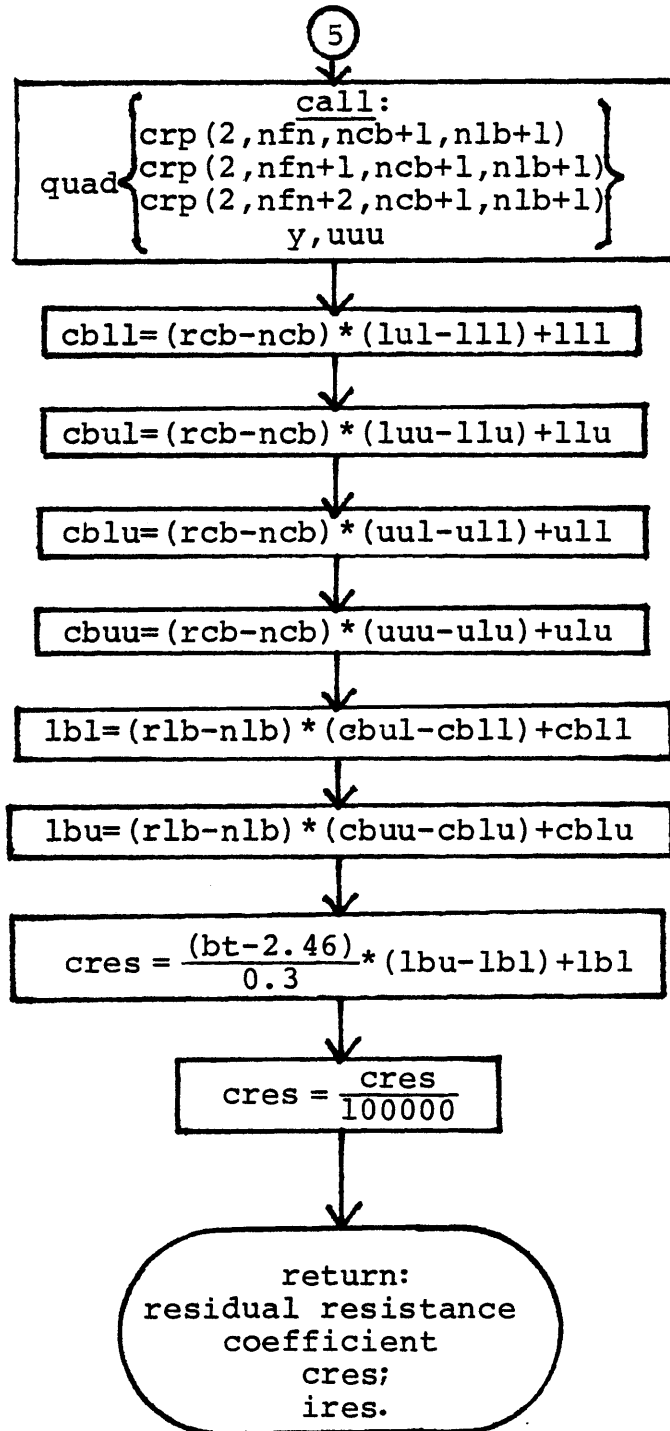
Subroutine: resist











```

1 power: procedure(l,b,dr,cb,v,wa,th,hr,propef,ehp,dhp,index,c,d,dt,dq,rc,t,w,eta);
2 dcl (c(39),d(47),dt(9),dq(13),rc(2,8,7,8),t(15,15),w(15,15),eta(15,15)) bin float;
3 dcl (l,b,dr,cb,v,wa,th,hr,propef,ehp,dhp,rpm) bin float;
4 dcl (cf,cr,ct,temp,r,lb,bt) bin float;
5 dcl (g,ro,gnu,fn,vs,wsa,rn,exarea,zb,pdiam) bin float;
6 dcl (i,iprop,ires,index) fixed bin;
7 dcl sysprint file stream output;
8 index = 0;
9 bt = b/dr;
10 bt = round(bt,2);
11 if bt<2.0 ; bt>3.25 then
12     put skip list("BT value out of reasonable extrapolable range--processing will continue");
13     lb = 1/b;
14 lb = round(lb,2);
15 vs = v*1.68731;
16 g = 32.1740;
17 ro = 1.9905;
18 gnu = 1.2791e-5;
19 fn = vs/sqrt(g*1);
20 fn = round(fn,3);
21 call propfactors(cb,lb,wa,th,hr,iprop);
22 if fn>0.20 then wa = (wa-0.02)/0.98;
23 if fn>0.16 & fn<0.20 then wa = wa + (fn-0.16)*(wa-1.0)/1.96;
24 if iprop = 1 then do;
25     index = 1;
26     return;
27 end;
28 call resist(cb,lb,bt,fn,cr,ires);
29 if ires = 1 then do;
30     index = 1;
31     return;
32 end;
33 pdiam=0.636*dr;
34 zb = 4.0;
35 wsa = 1.81*1*dr + cb*1*b;
36 rn = vs*1/gnu;
37 cf = 0.075/((log10(rn)-2.0e0)**2);
38 do i = 1 to 50;
39     temp = cf;
40     cf = (4.132e0*log10(rn*cf))**2;
41     if abs(temp - cf) < 5.0e-7 then go to next;
42 end;
43 put skip list("CF algorithm does not converge");
44 index = 1;
45 next:
46 ct = cr + cf + 0.0004;
47 r = 0.5*ro*wsa*vs**2*ct;
48 ehp = r*vs/550.0;
49 exarea=(1.3 + 0.3*zb)*(550.*ehp/vs)/((2080.8 + 64.0423*(pdiam/2 + 3.0))*pdiam**2) + 0.2;
50 call prop(zb,exarea,pdiam,ehp,v,th,wa,hr,dhp,rpm,propef);
51 return;
52 propfactors: procedure(cb,lb,wa,th,hr,iprop);
53 dcl (cb,lb,wa,th,hr,wp,tp,etar,rcb,rlb,rcbp,rlbp,lcb,ucb,llb,ulb) bin float;
54 dcl (iprop,icb,ilb,ncb,nlb) bin fixed;
55 iprop=0;
56 if cb=0.84 then rcb=14.999999; else rcb=(cb-0.77)*200.0 + 1.0;
57 if lb=7.6 then rlb=14.999999; else rlb=(lb-6.2)*10.0 + 1.0;
58 icb=floor(rcb);
59 ilb=floor(rlb);
60 if cb<0.75 ; cb>0.85 then do;

```

P


```

61      put skip list("CB value out of interpolable range");
62      put data(cb);
63      iprop=1;
64      return;
65      end;
66 if lb<6.0 ; lb>8.0 then do;
67     put skip list("LB value out of interpolable range");
68     put data(lb);
69     iprop=1;
70     return;
71     end;
72 if (icb<2 ; icb>14) & (ilb<1 ; ilb>14) then do;
73 /*      put skip list("LB and CB both out table ranges");      */
74     iprop=1;
75 /*      put skip data(lb,cb);      */
76     return;
77     end;
78 if (icb>0 & icb<15) & (ilb>0 & ilb<15) then do;
79     lcb=(rlb-ilb)*(w(icb,ilb+1)-w(icb,ilb)) + w(icb,ilb);
80     ucb=(rlb-ilb)*(w(icb+1,ilb+1)-w(icb+1,ilb)) + w(icb+1,ilb);
81     wp=(rcb-icb)*(ucb-lcb) + lcb;
82     lcb=(rlb-ilb)*(t(icb,ilb+1)-t(icb,ilb)) + t(icb,ilb);
83     ucb=(rlb-ilb)*(t(icb+1,ilb+1)-t(icb+1,ilb)) + t(icb+1,ilb);
84     tp=(rcb-icb)*(ucb-lcb) + lcb;
85     lcb=(rlb-ilb)*(eta(icb,ilb+1)-eta(icb,ilb)) + eta(icb,ilb);
86     ucb=(rlb-ilb)*(eta(icb+1,ilb+1)-eta(icb+1,ilb)) + eta(icb+1,ilb);
87     etar=(rcb-icb)*(ucb-lcb) + lcb;
88     go to compl;
89     end;
90 if icb<1 then do;
91     rcbp=rcb-1.0;
92     ncb=1;
93     go to quad1;
94     end;
95 if icb>14 then do;
96     rcbp=rcb-13.0;
97     ncb=13;
98     go to quad1;
99     end;
100 if ilb<1 then do;
101     rlbp=rlb-1.0;
102     nlb=1;
103     go to quad2;
104     end;
105 if ilb>14 then do;
106     rlbp=rlb-13.0;
107     nlb=13;
108     go to quad2;
109     end;
110 quad2:
111 call quad(w(icb,nlb),w(icb,nlb+1),w(icb,nlb+2),rlbp,lcb);
112 call quad(w(icb+1,nlb),w(icb+1,nlb+1),w(icb+1,nlb+2),rlbp,ucb);
113 wp=(ucb-lcb)*(rcb-icb) + lcb;
114 call quad(t(icb,nlb),t(icb,nlb+1),t(icb,nlb+2),rlbp,lcb);
115 call quad(t(icb+1,nlb),t(icb+1,nlb+1),t(icb+1,nlb+2),rlbp,ucb);
116 tp=(ucb-lcb)*(rcb-icb) + lcb;
117 call quad(eta(icb,nlb),eta(icb,nlb+1),eta(icb,nlb+2),rlbp,lcb);
118 call quad(eta(icb+1,nlb),eta(icb+1,nlb+1),eta(icb+1,nlb+2),rlbp,ucb);
119 etar=(ucb-lcb)*(rcb-icb) + lcb;
120 go to compl;

```

```

121 quad1:
122 call quad(w(ncb,ilb),w(ncb+1,ilb),w(ncb+2,ilb),rcbp,ilb);
123 call quad(w(ncb,ilb+1),w(ncb+1,ilb+1),w(ncb+2,ilb+1),rcbp,ulb);
124 wp=(ulb-ilb)*(rlb-ilb) + llb;
125 call quad(t(ncb,ilb),t(ncb+1,ilb),t(ncb+2,ilb),rcbp,ilb);
126 call quad(t(ncb,ilb+1),t(ncb+1,ilb+1),t(ncb+2,ilb+1),rcbp,ulb);
127 tp=(ulb-ilb)*(rlb-ilb) + llb;
128 call quad(eta(ncb,ilb),eta(ncb+1,ilb),eta(ncb+2,ilb),rcbp,ilb);
129 call quad(eta(ncb,ilb+1),eta(ncb+1,ilb+1),eta(ncb+2,ilb+1),rcbp,ulb);
130 etar=(ulb-ilb)*(rlb-ilb) + llb;
131 compl:
132 wa=1.0-wp/1000.0;
133 th=1.0-tp/1000.0;
134 hr=etar/1000.0;
135 return;
136 end propfactors;
137 quad: procedure(f1,f2,f3,x,y);
138 dcl (f1,f2,f3,y,x,x1,x2,x3) bin float;
139 x1=1.0 + (x-3.0)*x/2.0;
140 x2=(x-2.0)*x;
141 x3=(x-1.0)*x/2.0;
142 y=f1*x1 - f2*x2 + f3*x3;
143 return;
144 end quad;
145 resist: procedure(cb,lb,bt,fn,cres,ires);
146 dcl (cb,lb,bt,fn,crp(2,9,7,8),cres,lll,llu,lul,luu,ull,uul,uuu,ulu,y) float bin;
147 dcl (rcb,rlb,rfn,cbll,cbul,cbuu,lbll,lbll) float bin;
148 dcl (ires,icb,ilb,ifn,nlb,ncb,nfn,i,j,k,m) fixed bin;
149 ires = 0;
150 if cb=0.835 then rcb=6.999999; else rcb=(cb-0.775)*100.0 + 1.0;
151 if lb=7.6 then rlb=7.999999; else rlb=(lb-6.2)*5.0 + 1.0;
152 if fn=0.22 then rfn=8.999999; else rfn=(fn-0.14)*100.0 + 1.0;
153 icb=floor(rcb);
154 ilb=floor(rlb);
155 ifn=floor(rfn);
156 do i=1 to 2;
157 do k=1 to 7;
158 do m=1 to 8;
159 do j=3 to 9;
160 crp(i,j,k,m)=rc(i,j-1,k,m);
161 end;
162 crp(i,1,k,m)=rc(i,1,k,m);
163 crp(i,2,k,m)=0.5*(crp(i,1,k,m) + crp(i,3,k,m));
164 end; end; end;
165 if ifn>8 then do;
166 put skip list("FN value out of range");
167 put skip data(fn,v);
168 ires=1;
169 return;
170 end;
171 if ifn>0 then do;
172 if ((rfn-ifn)<0.5 & (ifn>1)) ; (ifn=8) then nfn=ifn-1;
173 else nfn=ifn;
174 y=rfn-nfn;
175 end;
176 else do;
177 nfn=1;
178 y=rfn-1.0;
179 end;
180 if(icb>0 & icb<7) & (ilb>0 & ilb<8) then do;

```

```

181     ncb=icb;
182     nlb=ilb;
183 end;
184 if (icb<1) then do;
185     ncb=1;
186     nlb=ilb;
187 end;
188 if (icb>6) then do;
189     ncb=6;
190     nlb=ilb;
191 end;
192 if (ilb<1) then do;
193     nlb=1;
194     if ncb^=1 ; ncb^=6 then ncb = icb;
195 end;
196 if (ilb>7) then do;
197     nlb=7;
198     if ncb^=1 ; ncb^=6 then ncb = icb;
199 end;
200 call quad(crp(1,nfn,ncb,nlb),crp(1,nfn+1,ncb,nlb),crp(1,nfn+2,ncb,nlb),y,lll);
201 call quad(crp(1,nfn,ncb,nlb+1),crp(1,nfn+1,ncb,nlb+1),crp(1,nfn+2,ncb,nlb+1),y,llu);
202 call quad(crp(1,nfn,ncb+1,nlb),crp(1,nfn+1,ncb+1,nlb),crp(1,nfn+2,ncb+1,nlb),y,lul);
203 call quad(crp(1,nfn,ncb+1,nlb+1),crp(1,nfn+1,ncb+1,nlb+1),crp(1,nfn+2,ncb+1,nlb+1),y,luu);
204 call quad(crp(2,nfn,ncb,nlb),crp(2,nfn+1,ncb,nlb),crp(2,nfn+2,ncb,nlb),y,u1l);
205 call quad(crp(2,nfn,ncb,nlb+1),crp(2,nfn+1,ncb,nlb+1),crp(2,nfn+2,ncb,nlb+1),y,u1u);
206 call quad(crp(2,nfn,ncb+1,nlb),crp(2,nfn+1,ncb+1,nlb),crp(2,nfn+2,ncb+1,nlb),y,u1l);
207 call quad(crp(2,nfn,ncb+1,nlb+1),crp(2,nfn+1,ncb+1,nlb+1),crp(2,nfn+2,ncb+1,nlb+1),y,u1u);
208 cbl1=(rcb-ncb)*(lul-llu) + ll1;
209 cbul=(rcb-ncb)*(luu-llu) + ll1;
210 cblu=(rcb-ncb)*(u1l-u1l) + u1l;
211 cbuu=(rcb-ncb)*(u1u-u1u) + u1u;
212 lbi=(rlb-nlb)*(cbul-cbl1) + cbl1;
213 lou=(rlb-nlb)*(cbuu-cblu) + cblu;
214 cres=((bt-2.46)/0.3)*(lbu-lb1) + lb1;
215 cres=cres*1.e-5;
216 return;
217 end resist;
218 prop: procedure(zb,x,om,hp,vr,th,wa,hr,dhp,rpm,propef);
219     del (f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12,f13,f14,f15,g1,g2,g3,
220         g4,g5,g6,g7,g8,g9,g10,g11,g12,g13,g14,g15,g16,g17,g18,g19,tb1,tb2,
221         tb3,tb4,tb5,qb1,qb2,qb3,qb4,qb5,qb6,qb7,tg1,tg2,tg3,tg4,qg1,qg2,qg3,
222         qg4,qg5,x,y,fff,om,hp,vr,th,wa,hr,rk,tq1,tq2,tq3,tq4,tq5,tq6,tq7,eps,
223         fr1,fr2,fr3,fr4,fr5,fr6,fr7,fr8,gr1,gr2,gr3,y1,y2,ym,
224         gr4,gr5,gr6,gr7,gr8,th1,th2,th3,th4,th5,qh1,qh2,qh3,qh4,qh5,dhp,rpm,
225         zb,z,z1,z2,zm,t,q,td,qd,tf,qf,propef,aa,bb,cc,for,rn,con,re) bin float;
226 del f(100) bin float;
227 del (k,ll) fixed bin;
228 eps=0.0001;
229 re=1.217*10.**7;
230 f1=c(1)+c(8)*zb+c(12)*x*zb+c(18)*x**2+c(29)*zb**2+c(36)*x*zb**2;
231 f3=c(7)*x+c(4)+c(28)*x**2*zb;
232 f4=c(23)*zb+c(14)+c(26)*x*zb+c(37)*x*zb**2+c(39)*x**2*zb**2;
233 f2=c(3)+c(10)*zb;
234 f5=c(15);
235 f6=c(2)+c(13)*x*zb+c(27)*x**2*zb+c(30)*zb**2;
236 f7=c(6)*x+c(11)*zb;
237 f8=c(33)*zb**2;
238 f9=c(24)*zb;
239 f10=c(21)*x**2+c(34)*zb**2;
240 f11=c(5)*x+c(9)*zb+c(19)*x**2+c(31)*zb**2;

```

```

241 f12=c(16)+c(22)*x**2+c(35)*zb**2;
242 f13=c(17)*x+c(20)*x**2+c(32)*zb**2;
243 f14=c(25)*zb;
244 f15=c(38)*zb**2*x;
245 g1=d(1)+d(19)*x**2+d(43)*x**2*zb**2;
246 g2=d(5)*x+d(10)*zb+d(22)*x**2;
247 g3=d(8)*x+d(4)+d(15)*x*zb+d(33)*x**2*zb+d(41)*x*zb**2;
248 g4=d(17)*x+d(25)*x**2+d(44)*x**2*zb**2;
249 g5=d(26)*x**2+d(29)*zb+d(31)*x*zb+d(35)*x**2*zb+d(42)*x*zb**2+d(46)
250 *x**2*zb**2;
251 g6=d(9)*zb+d(20)*x**2+d(32)*x**2*zb+d(39)*x*zb**2;
252 g7=d(3)+d(6)*x+d(11)*zb+d(14)*x*zb+d(36)*zb**2;
253 g9=d(18)*x;
254 g10=d(47)*x**2*zb**2;
255 g11=d(2)+d(13)*x*zb+d(40)*x*zb**2;
256 g12=d(7)*x+d(12)*zb;
257 g13=d(24)*x**2;
258 g14=d(34)*x**2*zb;
259 g15=d(16)*x+d(21)*x**2+d(27)*zb+d(30)*x*zb;
260 g16=d(23)*x**2;
261 g17=d(37)*zb**2;
262 g18=d(28)*zb+d(45)*zb**2*x**2;
263 g19=d(38)*zb**2;
264 tb1=dt(3)*x;
265 tb2=dt(8)*zb*x;
266 tb3=dt(9)*zb**2*x;
267 tb4=dt(2)*x;
268 tb5=dt(7)*zb*x+dt(4)*x;
269 qb1=dq(1)+dq(4)*x**2;
270 qb2=dq(12)*x**2;
271 qb3=dq(13)*x**2;
272 qb4=dq(10)*zb;
273 qb5=dq(11)*zb;
274 qb6=dq(9)*zb*x;
275 qb7=dq(8)*zb*x;
276 fff=log10(re)-.301;
277 tg1=tb1+tb2*fff;
278 tg2=tb3 *fff;
279 tg3=tb4+tb5*fff**2;
280 tg4=dt(5)*fff +dt(6)*fff**2;
281 qg1=qb1+qb2*fff +qb3*fff**2;
282 qg2=dq(2)+dq(5)*fff;
283 qg3=dq(6)*fff +dq(7)*fff**2;
284 qg4= qb4*fff +qb5 *fff**2 +dq(3)*zb;
285 qg5=qb6*fff**2;
286 qh3=qb7*fff;
287 th1=dt(1);
288 rk=5.2449*hp/((1.-wa)**2*(1.-th)*(om*.3048)**2*vr**3);
289 y1=.60;
290 y2=.40;
291 y=y1;
292 k=1;
293 N811: call grend(for,z);
294 f(k)=for;
295 k=k+1;
296 if k<3 then go to N809;
297 if k>3 then go to N840;
298 else go to N810;
299 N809: ym=(y1+y2)*.5;
300 if abs(y1-y2)<=eps*ym then do;

```

```

301      y=ym;
302      go to zeld;
303      end;
304      y=ym;
305      go to N811;
306      N810: bb=f(1);
307      N840: ll=k-1;
308      cc=f(ll);
309      if bb*cc<0.e0 then y2=ym;
310          else if bb*cc=0.e0 then do;
311              y = ym;
312              go to zeld;
313          end;
314          else do;
315              y1=ym;
316              bb=cc;
317              end;
318      go to N809;
319  zeld:  t=fr1+z*fr2+z**2*fr3+z**3*fr4;
320          q=gr1+z*gr2+z**2*gr3+z**3*gr4;
321          td=th1+z*th2+th3*z**2;
322          qd=qh1+z*qh2+z**2*qh3;
323          tf=t+td;
324          qf=q+qd;
325          propef=z*tf/(6.283185308*qf);
326          rpm=30.9*vr*(1.-wa)/(z*om*.3048);
327          dhp=hp*(1.-wa)/(propef*(1.-th)*hr);
328  grend: procedure(for,z);
329          dcl (for,z) bin float;
330          rn=2.;
331          fr1=f1+y*f2+y**2*f3+y**6*f5+y**3*f4;
332          fr2=f6+y*f7+y**2*f8+y**3*f9+f10*y**6;
333          fr3=f11+y**6*f12;
334          fr4=f13+f14*y**3+f15*y**6;
335          fr5=f2+2.*y*f3+6.*y**5*f5+3.*y**2*f4;
336          fr6=f7+2.*y*f8+3.*y**2*f9+6.*y**5*f10;
337          fr7=6.*y**5*f12;
338          fr8=6.*f15*y**5+3.*f14*y**2;
339          gr1=g1+y*g2+y**2*g3+y**3*g4+y**6*g5;
340          gr2=g6+y*g7          +g9*y**3+g10*y**6;
341          gr3=g11+g12*y+g13*y**2+g14*y**3;
342          gr4=g15+g16*y+g17*y**2+g19*y**6+y**3*g18;
343          gr5=g2+2.*y*g3+3.*y**2*g4+6.*y**5*g5;
344          gr6=g7          +3.*g9*y**2+6.*g10*y**5;
345          gr7=g12+2.*g13*y+3.*g14*y**2;
346          gr8=g16+2.*g17*y+3.*g18*y**2+6.*g19*y**5;
347          th2=y*tg1+y**3*tg2;
348          th3=tg3+          y**6*tg4;
349          th4=tg1+3.*y**2*tg2;
350          tn5=6.*y**5*tg4;
351          qh1=qg1+y*qg2+y**2*qg3+y**6*qg4;
352          qh2=qg5*y;
353          qh4=qg2+2.*y*qg3+6.*y**5*qg4;
354          qh5=qg5;
355          con=rk;
356          tq1=fr5*(rn+1.)*(gr1+qh1)-rn*(fr1+th1)*(gr5+qh4);
357          tq2=(rn+1.)*(gr1+qh1)*(fr6+th4)+(gr2+qh2)*fr5*rn+(gr5+qh4)*(1.-rn)
358          *(fr2+th2)-(gr6+qh5)*rn*(fr1+th1);
359          tq3=fr5*(rn-1.)*(gr3+qh3)+(fr6+th4)*(gr2+qh2)*rn+(fr7+th5)*(rn+1.)
360          *(gr1+qh1)-(gr5+qh4)*(2.-rn)*(fr3+th3)+(gr6+qh5)*(1.-rn)*(fr2+th2)

```

```

361      -gr7*rn*(fr1+th1);
362      tq4=fr5*(rn-2.)*gr4+(fr7+th5)*(gr2+qh2)*rn+(fr6+th4)*(rn-1.)*(gr3+
363      qh3)+fr8*(rn+1.)*(gr1+qh1)+(gr5+qh4)*(3.-rn)*fr4+(gr6+qh5)*(2.-rn)
364      *(fr3+th3)+gr7*(1.-rn)*(fr2+th2)-gr8*rn*(fr1+th1);
365      tq5=(fr6+th4)*(rn-2.)*gr4+(fr7+th5)*(rn-1.)*(gr3+qh3)+fr8*(gr2+qh2)
366      *rn+(gr6+qh5)*(3.-rn)*fr4+gr7*(2.-rn)*(fr3+th3)+gr8*(1.-rn)*(fr2+th2);
367      tq6=(fr7+th5)*(rn-2.)*gr4+fr8*(rn-1.)*(gr3+qh3)+gr7*(3.-rn)*fr4+gr8
368      *(2.-rn)*(fr3+th3);
369      tq7=fr8*(rn-2.)*gr4+gr8*(3.-rn)*fr4;
370      z1=.01;
371      z2=y+.1;
372      aa=fr1+th1+(fr2+th2)*z1+(fr3+th3)*z1**2+fr4*
373      z1**3-con*z1**2;
374      N801: zm=(z1+z2)*.5;
375      cc=fr1+th1+(fr2+th2)*zm+(fr3+th3)*zm**2+fr4*
376      zm**3-con*zm**2;
377      if abs(z1-z2)<=eps*zm then go to N805;
378      if aa*cc=0.e0 then go to N805;
379      if aa*cc<0.e0 then z2=zm;
380      else do;
381          z1=zm;
382          aa=cc;
383      end;
384      go to N801;
385      N805: z=zm;
386      for      =(tq1+z*tq2+z**2*tq3+z**3*tq4+z**4*tq5+z**5*tq6+z**6*tq7)
387      *con*z**2/((gr1+qh1+(gr2+qh2)*z+(qh3+gr3)*z**2+gr4*z**3)**2*(fr2
388      +th2+2.*(fr3+th3)*z+3.*fr4*z**2 -rn*con*z));
389      return;
390      end grend;
391  end prop;
392  end power;

```

\\314 SOURCE FILES USED IN THIS COMPILATION.

LINE	NUMBER	DATE MODIFIED	NAME	PATHNAME
\\314	0	04/14/79	2316.7 power.pl1	>udd>Barges>Kaskin>power.pl1

IDENTIFIER	OFFSET	LOC	STORAGE CLASS	DATA TYPE	ATTRIBUTES AND REFERENCES
NAMES DECLARED BY DECLARE STATEMENT.					
aa	000364		automatic	float bin(27)	dcl 219 set ref 372 378 379 382
b			parameter	float bin(27)	dcl 3 ref 1 9 13 35
bb	000365		automatic	float bin(27)	dcl 219 set ref 306 309 310 316
bt			parameter	float bin(27)	dcl 146 ref 145 214
bt	000107		automatic	float bin(27)	dcl 4 set ref 9 10 10 11 11 28
c			parameter	float bin(27)	array dcl 2 ref 1 230 230 230 230 230 230 231 231 231 232 232 232 232 232 233 233 234 235 235 235 235 236 236 237 238 239 239 240 240 240 241 241 241 242 242 242 243 244 dcl 146 ref 145 150 150 dcl 53 ref 52 56 56 60 60 62 dcl 3 set ref 1 21 28 35 dcl 147 set ref 208 212 212 dcl 147 set ref 210 213 213 dcl 147 set ref 209 212 dcl 147 set ref 211 213 dcl 219 set ref 308 309 310 316 375 378 379 382 dcl 4 set ref 37 39 40 40 41 45 dcl 219 set ref 355 372 375 386 386 dcl 4 set ref 28 45 dcl 146 set ref 145 214 215 215 array dcl 146 set ref 160 162 163 163 163 200 200 200 201 201 201 202 202 202 203 203 203 204 204 204 205 205 205 206 206 206 207 207 207 dcl 4 set ref 45 47 array dcl 2 ref 1 245 245 245 246 246 246 247 247 247 247 247 248 248 248 249 249 249 249 249 249 251 251 251 251 252 252 252 252 252 253 254 255 255 255 256 256 257 258 259 259 259 260 261 262 262 263 dcl 3 set ref 1 50 dcl 219 set ref 218 327 array dcl 2 ref 1 269 269 270 271 272 273 274 275 282 282 283 283 284 dcl 3 ref 1 9 33 35 array dcl 2 ref 1 264 265 266 267 268 268 280 280 287 dcl 3 set ref 1 48 49 50 dcl 219 set ref 228 300 377 array dcl 2 set ref 1 85 85 85 86 86 86 117 117 117 118 118 118 128 128 128 129 129 129 dcl 53 set ref 87 119 130 134 dcl 5 set ref 49 50 array dcl 226 set ref 294 306 308 dcl 219 set ref 230 331 dcl 138 ref 137 142 dcl 219 set ref 239 332 336 dcl 219 set ref 240 333 dcl 219 set ref 241 333 337 dcl 219 set ref 242 334 dcl 219 set ref 243 334 338
cb			parameter	float bin(27)	
cb			parameter	float bin(27)	
cb			parameter	float bin(27)	
cbll	002074		automatic	float bin(27)	
cblu	002076		automatic	float bin(27)	
cbul	002075		automatic	float bin(27)	
cbuu	002077		automatic	float bin(27)	
cc	000366		automatic	float bin(27)	
cf	000101		automatic	float bin(27)	
con	000371		automatic	float bin(27)	
cr	000102		automatic	float bin(27)	
cres			parameter	float bin(27)	
crp	000100		automatic	float bin(27)	
ct	000103		automatic	float bin(27)	
d			parameter	float bin(27)	
dnp			parameter	float bin(27)	
dnp			parameter	float bin(27)	
dq			parameter	float bin(27)	
dr			parameter	float bin(27)	
dt			parameter	float bin(27)	
enp			parameter	float bin(27)	
eps	000314		automatic	float bin(27)	
eta			parameter	float bin(27)	
etar	000102		automatic	float bin(27)	
exarea	000117		automatic	float bin(27)	
f	000373		automatic	float bin(27)	
f1	000214		automatic	float bin(27)	
f1			parameter	float bin(27)	
f10	000225		automatic	float bin(27)	
f11	000226		automatic	float bin(27)	
f12	000227		automatic	float bin(27)	
f13	000230		automatic	float bin(27)	
f14	000231		automatic	float bin(27)	

f15	000232	automatic	float bin(27)	dcl 219 set ref 244 334 338
f2		parameter	float bin(27)	dcl 138 ref 137 142
f2	000215	automatic	float bin(27)	dcl 219 set ref 233 331 335
f3	000216	automatic	float bin(27)	dcl 219 set ref 231 331 335
f3		parameter	float bin(27)	dcl 138 ref 137 142
f4	000217	automatic	float bin(27)	dcl 219 set ref 232 331 335
f5	000220	automatic	float bin(27)	dcl 219 set ref 234 331 335
f6	000221	automatic	float bin(27)	dcl 219 set ref 235 332
f7	000222	automatic	float bin(27)	dcl 219 set ref 236 332 336
f3	000223	automatic	float bin(27)	dcl 219 set ref 237 332 336
f3	000224	automatic	float bin(27)	dcl 219 set ref 238 332 336
fff	000303	automatic	float bin(27)	dcl 219 set ref 276 277 278 279 280 280 281 281 282 283 283 284 284 285 286
fn	000113	automatic	float bin(27)	dcl 5 set ref 19 20 20 22 23 23 28
fn		parameter	float bin(27)	dcl 146 ref 145 152 152 167
for		parameter	float bin(27)	dcl 329 set ref 328 386
for	000367	automatic	float bin(27)	dcl 219 set ref 293 294
fr1	000315	automatic	float bin(27)	dcl 219 set ref 319 331 356 357 359 362 372 375
fr2	000316	automatic	float bin(27)	dcl 219 set ref 319 332 357 359 362 365 372 375 386
fr3	000317	automatic	float bin(27)	dcl 219 set ref 319 333 359 362 365 367 372 375 386
fr4	000320	automatic	float bin(27)	dcl 219 set ref 319 334 362 365 367 369 372 375 386
fr5	000321	automatic	float bin(27)	dcl 219 set ref 335 356 357 359 362
fr6	000322	automatic	float bin(27)	dcl 219 set ref 336 357 359 362 365
fr7	000323	automatic	float bin(27)	dcl 219 set ref 337 359 362 365 367
fr8	000324	automatic	float bin(27)	dcl 219 set ref 338 362 365 367 369
e1	000110	automatic	float bin(27)	dcl 5 set ref 16 19
e10	000233	automatic	float bin(27)	dcl 219 set ref 245 339
e11	000243	automatic	float bin(27)	dcl 219 set ref 254 340 344
e12	000244	automatic	float bin(27)	dcl 219 set ref 255 341
e13	000245	automatic	float bin(27)	dcl 219 set ref 256 341 345
e14	000246	automatic	float bin(27)	dcl 219 set ref 257 341 345
e15	000247	automatic	float bin(27)	dcl 219 set ref 258 341 345
e16	000250	automatic	float bin(27)	dcl 219 set ref 259 342
e17	000251	automatic	float bin(27)	dcl 219 set ref 260 342 346
e18	000252	automatic	float bin(27)	dcl 219 set ref 261 342 346
e19	000253	automatic	float bin(27)	dcl 219 set ref 262 342 346
e2	000254	automatic	float bin(27)	dcl 219 set ref 263 342 346
e3	000234	automatic	float bin(27)	dcl 219 set ref 246 339 343
e4	000235	automatic	float bin(27)	dcl 219 set ref 247 339 343
e5	000236	automatic	float bin(27)	dcl 219 set ref 248 339 343
e6	000237	automatic	float bin(27)	dcl 219 set ref 249 339 343
e7	000240	automatic	float bin(27)	dcl 219 set ref 251 340
e8	000241	automatic	float bin(27)	dcl 219 set ref 252 340 344
eru	000242	automatic	float bin(27)	dcl 219 set ref 253 340 344
er1	000112	automatic	float bin(27)	dcl 5 set ref 18 36
er2	000325	automatic	float bin(27)	dcl 219 set ref 320 339 356 357 359 362 386
er3	000326	automatic	float bin(27)	dcl 219 set ref 320 340 357 359 362 365 386
er4	000327	automatic	float bin(27)	dcl 219 set ref 320 341 359 362 365 367 386
er5	000333	automatic	float bin(27)	dcl 219 set ref 320 342 362 365 367 369 386
er6	000334	automatic	float bin(27)	dcl 219 set ref 343 356 357 359 362
er7	000335	automatic	float bin(27)	dcl 219 set ref 344 357 359 362 365
er8	000336	automatic	float bin(27)	dcl 219 set ref 345 359 362 365 367
np	000337	automatic	float bin(27)	dcl 219 set ref 346 362 365 367 369
nr		parameter	float bin(27)	dcl 219 ref 218 288 327
nr		parameter	float bin(27)	dcl 219 ref 218 327
nr		parameter	float bin(27)	dcl 53 set ref 52 134
nr		parameter	float bin(27)	dcl 3 set ref 1 21 50

i	002110	automatic	fixed bin(17,0)	dcl 148 set ref 156 160 160 162 162 163 163 163
i	000122	automatic	fixed bin(17,0)	dcl 6 set ref 38
icb	002102	automatic	fixed bin(17,0)	dcl 148 set ref 153 180 180 181 184 188 194 198
icb	000113	automatic	fixed bin(17,0)	dcl 54 set ref 58 72 72 78 78 79 79 80 80 80
				81 82 82 82 83 83 83 84 85 85 85 86 86 86 87 90
				95 110 110 110 112 112 112 113 114 114 114 115
				115 115 116 117 117 117 118 118 118 119
ifn	002104	automatic	fixed bin(17,0)	dcl 148 set ref 155 165 171 172 172 172 172 173
ilb	002103	automatic	fixed bin(17,0)	dcl 148 set ref 154 180 180 182 186 190 192 196
ilb	000114	automatic	fixed bin(17,0)	dcl 54 set ref 59 72 72 78 78 79 79 79 80 80
				80 80 82 82 82 82 83 83 83 83 85 85 85 85 86 86
				86 86 100 105 121 121 121 123 123 123 124 125
				125 125 126 126 126 127 128 128 128 129 129 129
				130
index		parameter	fixed bin(17,0)	dcl 6 set ref 1 8 25 30 44
iprop	000123	automatic	fixed bin(17,0)	dcl 6 set ref 21 24
iprop		parameter	fixed bin(17,0)	dcl 54 set ref 52 55 63 69 74
ires		parameter	fixed bin(17,0)	dcl 148 set ref 145 149 168
ires	000124	automatic	fixed bin(17,0)	dcl 6 set ref 28 29
j	002111	automatic	fixed bin(17,0)	dcl 148 set ref 159 160 160
k	000537	automatic	fixed bin(17,0)	dcl 227 set ref 292 294 295 295 296 297 307
k	002112	automatic	fixed bin(17,0)	dcl 148 set ref 157 160 160 162 162 163 163 163
l		parameter	float bin(27)	dcl 3 ref 1 13 19 35 35 36
lb		parameter	float bin(27)	dcl 53 ref 52 57 57 66 66 68
lb		parameter	float bin(27)	dcl 146 ref 145 151 151
lo	000106	automatic	float bin(27)	dcl 4 set ref 13 14 14 21 28
lcl	002101	automatic	float bin(27)	dcl 147 set ref 212 214 214
ltu	002100	automatic	float bin(27)	dcl 147 set ref 213 214
lcb	000107	automatic	float bin(27)	dcl 53 set ref 79 81 81 82 84 84 85 87 87 110
				113 113 114 116 116 117 119 119
ll	000540	automatic	fixed bin(17,0)	dcl 227 set ref 307 308
llb	000111	automatic	float bin(27)	dcl 53 set ref 121 124 124 125 127 127 128 130
				130
lll	002060	automatic	float bin(27)	dcl 146 set ref 200 208 208
llu	002061	automatic	float bin(27)	dcl 146 set ref 201 209 209
lui	002062	automatic	float bin(27)	dcl 146 set ref 202 208
luu	002063	automatic	float bin(27)	dcl 146 set ref 203 209
m	002113	automatic	fixed bin(17,0)	dcl 148 set ref 158 160 160 162 162 163 163 163
nob	002106	automatic	fixed bin(17,0)	dcl 148 set ref 181 185 189 194 194 194 198 198
				198 200 200 200 201 201 201 202 202 202 203 203
				203 204 204 204 205 205 205 206 206 206 207 207
				207 208 209 210 211
nob	000115	automatic	fixed bin(17,0)	dcl 54 set ref 92 97 121 121 121 123 123 123 125
				125 125 126 126 126 128 128 128 129 129 129
nfn	002107	automatic	fixed bin(17,0)	dcl 148 set ref 172 173 174 177 200 200 200 201
				201 201 202 202 202 203 203 203 204 204 204 205
				205 205 206 206 206 207 207 207
nib	002105	automatic	fixed bin(17,0)	dcl 148 set ref 182 186 190 193 197 200 200 200
				201 201 201 202 202 202 203 203 203 204 204 204
				205 205 205 206 206 206 207 207 207 212 213
nib	000116	automatic	fixed bin(17,0)	dcl 54 set ref 102 107 110 110 110 112 112 112
				114 114 114 115 115 115 117 117 117 118 118 118
on		parameter	float bin(27)	dcl 219 ref 218 288 326
ptiam	000121	automatic	float bin(27)	dcl 5 set ref 33 49 49 50
propef		parameter	float bin(27)	dcl 3 set ref 1 50
propef		parameter	float bin(27)	dcl 219 set ref 218 325 327
q	000357	automatic	float bin(27)	dcl 219 set ref 320 324
qb1	000262	automatic	float bin(27)	dcl 219 set ref 269 281
qb2	000263	automatic	float bin(27)	dcl 219 set ref 270 281
qb3	000264	automatic	float bin(27)	dcl 219 set ref 271 281

qb4	000265	automatic	float bin(27)	dcl 219 set ref 272 284
qb5	000266	automatic	float bin(27)	dcl 219 set ref 273 284
qb6	000267	automatic	float bin(27)	dcl 219 set ref 274 285
qb7	000270	automatic	float bin(27)	dcl 219 set ref 275 286
qd	000361	automatic	float bin(27)	dcl 219 set ref 322 324
qf	000363	automatic	float bin(27)	dcl 219 set ref 324 325
qg1	000275	automatic	float bin(27)	dcl 219 set ref 281 351
qg2	000276	automatic	float bin(27)	dcl 219 set ref 282 351 353
qg3	000277	automatic	float bin(27)	dcl 219 set ref 283 351 353
qg4	000300	automatic	float bin(27)	dcl 219 set ref 284 351 353
qg5	000301	automatic	float bin(27)	dcl 219 set ref 285 352 354
qh1	000345	automatic	float bin(27)	dcl 219 set ref 322 351 356 357 359 362 386
qh2	000346	automatic	float bin(27)	dcl 219 set ref 322 352 357 359 362 365 386
qh3	000347	automatic	float bin(27)	dcl 219 set ref 286 322 359 362 365 367 386
qh4	000350	automatic	float bin(27)	dcl 219 set ref 353 356 357 359 362
qh5	000351	automatic	float bin(27)	dcl 219 set ref 354 357 359 362 365
r	000105	automatic	float bin(27)	dcl 4 set ref 47 48
rc		parameter	float bin(27)	array dcl 2 ref 1 160 162
rcb	000103	automatic	float bin(27)	dcl 53 set ref 56 56 58 81 84 87 91 96 113 116 119
rb	002071	automatic	float bin(27)	dcl 147 set ref 150 150 153 208 209 210 211
rcop	000105	automatic	float bin(27)	dcl 53 set ref 91 96 121 123 125 126 128 129
re	000372	automatic	float bin(27)	dcl 219 set ref 229 276
rfn	002073	automatic	float bin(27)	dcl 147 set ref 152 152 155 172 174 178
rk	000304	automatic	float bin(27)	dcl 219 set ref 288 355
rlb	000104	automatic	float bin(27)	dcl 53 set ref 57 57 59 79 80 82 83 85 86 101 106 124 127 130
rib	002072	automatic	float bin(27)	dcl 147 set ref 151 151 154 212 213
rlbp	000106	automatic	float bin(27)	dcl 53 set ref 101 106 110 112 114 115 117 118
rn	000370	automatic	float bin(27)	dcl 219 set ref 330 356 356 357 357 357 357 359 359 359 359 362 362 362 362 362 362 362 365 365 365 365 365 365 367 367 367 367 367 369 386
rn	000116	automatic	float bin(27)	dcl 5 set ref 36 37 40
ro	000111	automatic	float bin(27)	dcl 5 set ref 17 47
rpa		parameter	float bin(27)	dcl 219 set ref 218 320
rpa	000100	automatic	float bin(27)	dcl 3 set ref 50
sysprint	000016	constant	file	output stream dcl 7 set ref 7 11 43 61 62 67 68 166 167
t		parameter	float bin(27)	array dcl 2 set ref 1 82 82 82 83 83 83 114 114 114 115 115 115 115 125 125 125 125 126 126 126
t	000356	automatic	float bin(27)	dcl 219 set ref 319 323
tb1	000255	automatic	float bin(27)	dcl 219 set ref 264 277
tb2	000256	automatic	float bin(27)	dcl 219 set ref 265 277
tb3	000257	automatic	float bin(27)	dcl 219 set ref 266 278
tb4	000260	automatic	float bin(27)	dcl 219 set ref 267 279
tb5	000261	automatic	float bin(27)	dcl 219 set ref 268 279
ti	000360	automatic	float bin(27)	dcl 219 set ref 321 323
temp	000104	automatic	float bin(27)	dcl 4 set ref 39 41
tf	000362	automatic	float bin(27)	dcl 219 set ref 323 325
tg1	000271	automatic	float bin(27)	dcl 219 set ref 277 347 349
tg2	000272	automatic	float bin(27)	dcl 219 set ref 278 347 349
tg3	000273	automatic	float bin(27)	dcl 219 set ref 279 348
tg4	000274	automatic	float bin(27)	dcl 219 set ref 280 348 350
tn		parameter	float bin(27)	dcl 3 set ref 1 21 50
th		parameter	float bin(27)	dcl 219 ref 218 288 327
tn		parameter	float bin(27)	dcl 53 set ref 52 133
th1	000340	automatic	float bin(27)	dcl 219 set ref 287 321 356 357 359 362 372 375
th2	000341	automatic	float bin(27)	dcl 219 set ref 321 347 357 359 362 365 372 375 386

th3	000342	automatic	float bin(27)	dcl 219 set ref 321 348 359 362 365 367 372 375 386
th4	000343	automatic	float bin(27)	dcl 219 set ref 349 357 359 362 365
th5	000344	automatic	float bin(27)	dcl 219 set ref 350 359 362 365 367
tp	000101	automatic	float bin(27)	dcl 53 set ref 84 116 127 133
tq1	000305	automatic	float bin(27)	dcl 219 set ref 356 386
tq2	000306	automatic	float bin(27)	dcl 219 set ref 357 386
tq3	000307	automatic	float bin(27)	dcl 219 set ref 359 386
tq4	000310	automatic	float bin(27)	dcl 219 set ref 362 386
tq5	000311	automatic	float bin(27)	dcl 219 set ref 365 386
tq6	000312	automatic	float bin(27)	dcl 219 set ref 367 386
tq7	000313	automatic	float bin(27)	dcl 219 set ref 369 386
ucb	000110	automatic	float bin(27)	dcl 53 set ref 80 81 83 84 86 87 112 113 115 116 118 119
ulb	000112	automatic	float bin(27)	dcl 53 set ref 123 124 126 127 129 130
ull	002064	automatic	float bin(27)	dcl 146 set ref 204 210 210
ulu	002067	automatic	float bin(27)	dcl 146 set ref 205 211 211
uul	002065	automatic	float bin(27)	dcl 146 set ref 206 210
uuu	002066	automatic	float bin(27)	dcl 146 set ref 207 211
v		parameter	float bin(27)	dcl 3 set ref 1 15 50 167
vr		parameter	float bin(27)	dcl 219 ref 218 288 326
vs	000114	automatic	float bin(27)	dcl 5 set ref 15 19 36 47 48 49
w		parameter	float bin(27)	array dcl 2 set ref 1 79 79 79 80 80 80 110 110 110 112 112 112 121 121 121 123 123 123
wa		parameter	float bin(27)	dcl 219 ref 218 288 326 327
wa		parameter	float bin(27)	dcl 3 set ref 1 21 22 22 23 23 23 50
wp	000100	automatic	float bin(27)	dcl 53 set ref 52 131
wsa	000115	automatic	float bin(27)	dcl 53 set ref 81 113 124 131
x		parameter	float bin(27)	dcl 5 set ref 35 47
x		parameter	float bin(27)	dcl 138 ref 137 139 139 140 140 141 141
				dcl 219 ref 218 230 230 230 231 231 232 232 232
				235 235 236 239 240 240 241 242 242 244 245 245
				246 246 247 247 247 247 248 248 248 249 249 249
				249 249 251 251 251 252 252 253 254 255 255 256
				257 258 259 259 259 260 262 264 265 266 267 268
				268 269 270 271 274 275
				dcl 138 set ref 139 142
				dcl 138 set ref 140 142
				dcl 138 set ref 141 142
				dcl 138 set ref 137 142
				dcl 219 set ref 291 301 304 311 331 331 331 331
				332 332 332 332 333 334 334 335 335 335 336 336
				336 337 338 338 339 339 339 339 340 340 340 341
				341 341 342 342 342 342 343 343 343 344 344 345
				345 346 346 346 347 347 348 349 350 351 351 351
				352 353 353 371
				dcl 146 set ref 174 178 200 201 202 203 204 205 206 207
y	002070	automatic	float bin(27)	dcl 219 set ref 289 291 299 300 315
y1	000330	automatic	float bin(27)	dcl 219 set ref 290 299 300 309
y2	000331	automatic	float bin(27)	dcl 219 set ref 299 300 301 304 309 311 315
ym	000332	automatic	float bin(27)	dcl 329 set ref 328 385 386 386 386 386 386 386
z		parameter	float bin(27)	386 386 386 386 386 386 386 386
z	000352	automatic	float bin(27)	dcl 219 set ref 293 319 319 319 320 320 320 321 321 322 322 325 326
z1	000353	automatic	float bin(27)	dcl 219 set ref 370 372 372 372 372 374 377 381
z2	000354	automatic	float bin(27)	dcl 219 set ref 371 374 377 379
z0		parameter	float bin(27)	dcl 219 ref 218 230 230 230 230 231 232 232 232 232 233 235 235 235 236 237 238 239 240 240 241 242 243 244 245 246 247 247 247 248 249 249 249

zb	000120	automatic	float bin(27)	249 249 251 251 251 252 252 252 254 255 255 256 258 259 259 261 262 262 263 265 266 268 272 273 274 275 284
zm	000355	automatic	float bin(27)	dcl 5 set ref 34 49 50 dcl 219 set ref 374 375 375 375 375 377 379 381 385

NAMES DECLARED BY EXPLICIT CONTEXT.

N201	006643	constant	label	dcl 374 ref 374 384
N205	006731	constant	label	dcl 385 ref 377 378 385
N209	005177	constant	label	dcl 299 ref 296 299 318
N210	005221	constant	label	dcl 306 ref 298 306
N211	005164	constant	label	dcl 293 ref 293 305
N240	005223	constant	label	dcl 307 ref 297 307
compl	002444	constant	label	dcl 131 ref 88 120 131
grend	005374	constant	entry	internal dcl 328 ref 293 328
next	000601	constant	label	dcl 45 ref 41 45
power	000251	constant	entry	external dcl 1 ref 1
prop	004072	constant	entry	internal dcl 218 ref 50 218
propfactors	000703	constant	entry	internal dcl 52 ref 21 52
quad	002464	constant	entry	internal dcl 137 ref 110 112 114 115 117 118 121 123 125 126 128 129 137 200 201 202 203 204 205 206 207
quad1	002062	constant	label	dcl 121 ref 93 98 121
quad2	001470	constant	label	dcl 110 ref 103 108 110
resist	002525	constant	entry	internal dcl 145 ref 28 145
zeld	005250	constant	label	dcl 319 ref 302 312 319

NAMES DECLARED BY CONTEXT OR IMPLICATION.

abs			builtin function	internal ref 41 300 377
floor			builtin function	internal ref 58 59 153 154 155
log10			builtin function	internal ref 37 40 276
round			builtin function	internal ref 10 14 20
sqrt			builtin function	internal ref 19

STORAGE REQUIREMENTS FOR THIS PROGRAM.

	Object	Text	Link	Symbol	Defs	Static
Start	0	0	7200	7222	7101	7210
Length	14450	7101	22	5212	76	2

BLOCK NAME	STACK SIZE	TYPE	WHY NONQUICK/WHO SHARES STACK FRAME
power	509	external procedure	is an external procedure.
propfactors	146	internal procedure	uses I/O statements.
quad	69	internal procedure	is called by several nonquick procedures.
resist	1167	internal procedure	uses I/O statements.
prop		internal procedure	shares stack frame of external procedure power.
grend		internal procedure	shares stack frame of external procedure power.

THE FOLLOWING EXTERNAL OPERATORS ARE USED BY THIS PROGRAM.

fx1_to_fx2	r_l_a	r_g_a	r_e_as	r_ne_as	call_ext_out
call_int_this	call_int_other	return	fx2_to_fx1	ext_entry	int_entry
put_end	stream_io	floor_f1	round_f1	put_data_eis	put_list_eis
real_to_real_rd	sqrt	log10	real_p_real		

THE FOLLOWING EXTERNAL ENTRIES ARE CALLED BY THIS PROGRAM.

decimal_exp_

THE FOLLOWING EXTERNAL VARIABLES ARE USED BY THIS PROGRAM.

sysprint

sysprint.fsb

C.3 DICTIONARY OF VARIABLES USED IN THE BARGE POWERING MODEL

DEFINITION OF THE VARIABLES USED IN SUBROUTINE "power"

<u>Variable</u>	<u>Definition</u> (units)
b:	Barge Molded Breadth.
bt:	B/T. Ratio of barge molded breadth to barge draft.
c:	An array of values used as coefficients in Subroutine "prop".
cb:	Cb. Barge block coefficient.
cf:	Frictional Resistance Coefficient.
cr:	Residual Resistance Coefficient.
ct:	Total Resistance Coefficient.
d:	An array of values used as coefficients in Subroutine "prop".
dhp:	Delivered Horsepower. Horsepower required to be delivered to the propeller to provide the effective horsepower needed to propel barge at the specified speed.
dq:	An array of values used as coefficients in Subroutine "prop".
dr:	Barge Summer Waterline Draft.
dt:	An array of values used as coefficients in Subroutine "prop".
ehp:	Effective horsepower required to be delivered by the propeller to drive the barge at the specified speed.
eta:	Array of values of relative rotative efficiency coefficients used as input into Subroutine "propfactors".

exarea: Propeller Expanded Area Ratio. (sq in)
 fn: Froude Number.
 g: Acceleration Due to Gravity. (ft/sec)
 gnu: Kinematic Velocity.
 hr: Relative rotative efficiency for hull of specified form. Output of Subroutine "propfactors" and input into Subroutine "prop".
 i: Index counter used in frictional resistance routine.
 index: Subprogram "power" Error Index. Index that indicates whether calls to subroutines "propfactors" and "resist" have not been successful or that the frictional resistance routine does not converge (index=0 indicates success of Subroutine "power", index=1 indicates some error in the execution).
 iprop: Subroutine "propfactors" Error Index. Index that indicates to Subroutine "power" that some error occurred during the execution of Subroutine "propfactors". (iprop=0 indicates no error; iprop=1 indicates error.)
 ires: Subroutine "resist" Error Index. Index that indicates to Subroutine "power" that some error occurred during the execution of Subroutine "resist". (ires=0 indicates no error; ires=1 indicates error.)
 l: Barge Length Between Perpindiculars.
 lb: L/B. Ratio of barge length to barge molded breadth.
 pdiam: Diameter of Propeller. (ft)
 propef: Propeller Open Water Propulsive Efficiency. Output of Subroutine "prop".
 rc: An array of values of residual resistance coefficients, used as input into Subroutine "resist".
 r: Barge Resistance. Total still water resistance developed by barge.

rn: Reynold's Number.
 ro: Density of Sea Water.
 rpm: RPM. Revolutions per minute of propeller at specified speed.
 sysprint: Output Switch Name.
 t: Array of values of thrust deduction fraction used as input into Subroutine "propfactors".
 temp: Temporary holding variable used in routine to calculate frictional resistance coefficient.
 th: Thrust deduction fraction for hull of specified form. Output of Subroutine "propfactors" and input into Subroutine "prop".
 v: Speed of Barge. (kts)
 vs: Speed of Barge. (ft/sec)
 w: Array of values of wake fraction used as input into Subroutine "propfactors".
 wa: Wake fraction for hull of specified form. Output of Subroutine "propfactors" and input into Subroutine "prop".
 wsa: Barge Wetted Surface Area.
 zb: Number of Blades on Propeller.

DEFINITION OF THE VARIABLES USED IN SUBROUTINE "propfactors"

<u>Variable</u>	<u>Definition</u>
cb:	Cb. Barge block coefficient.
etar:	Relative rotative efficiency for the specific barge form under consideration, multiplied by one thousand.
hr:	Relative rotative efficiency for the specific barge form under consideration.
icb:	Index used in the propulsion coefficient arrays indicating the first of two values with respect to cb that are used as inputs into the linear and quadratic interpolation routines for lb interpolation.
ilb:	Index used in the propulsion coefficient arrays indicating the first of two values with respect to lb that are used as inputs into the linear and quadratic interpolation routines for cb interpolation.
iprop:	Error Index. Index used to indicate whether lb or cb inputs are out of propulsion coefficient table ranges; (iprop=1 indicates inputs out of range; iprop=0 indicates inputs satisfactory).
lb:	L/B. Ratio of barge length to barge molded breadth.
lcb:	The linear or quadratic interpolated value with respect to lb for the lower of two cb values.
llb:	The quadratic interpolated value with respect to cb for the lower of two lb values.

ncb: Index used in propulsion coefficient arrays indicating the first of three cb table values that are inputted into the cb quadratic interpolation routine for a given lb value (ilb).

nlb: Index used in propulsion coefficient arrays indicating the first of three lb table values that are inputted into the lb quadratic interpolation routine for a given cb value (icb).

rcb: cb index value for which table interpolation is desired.

rcbp: Fractional part of cb index value for which table interpolation is desired and which is inputted into the cb quadratic interpolation routines.

rlb: lb index value for which table interpolation is desired.

rlbp: Fractional part of lb index value for which table interpolation is desired and which is inputted into the lb quadratic interpolation routines.

th: Thrust deduction fraction for the specific barge form under consideration.

tp: One thousand times one minus the thrust deduction fraction for the specific barge form under consideration.

ucb: The linear or quadratic interpolated value with respect to lb for the upper of two cb values.

ulb: The quadratic interpolated value with respect to cb for the upper of two lb values.

wa: The wake fraction for the specific barge form under consideration.

wp: One thousand times one minus the wake fraction for the specific barge form under consideration.

DEFINITION OF THE VARIABLES USED IN SUBROUTINE "resist"

<u>Variable</u>	<u>Definition</u>
bt:	B/T. Ratio of barge breadth to barge draft.
cb:	Cb. Barge block coefficient.
cbll:	Value of linear interpolation between two cb index values for the first of two lb index values, where bt = 2.46.
cblu:	Value of linear interpolation between two cb index values for the first of two lb index values, where bt = 2.76.
cbul:	Value of linear interpolation between two cb index values for the second of two lb index values, where bt = 2.46.
cbuu:	Value of linear interpolation between two cb index values for the second of two lb index values, where bt = 2.76.
cres:	Residual resistance coefficient found from the interpolation of the residual resistance table.
crp:	Array of residual resistance coefficients used for table interpolation. This array is identical to the input array rc except that 0.15 Froud number value is assumed to be linearly interpolated between the 0.14 and 0.16 Froude number values.
fn:	Froude Number.
i:	The bt index of the residual resistance array.
icb:	The integral part of the cb index value for which interpolation is desired.
ifn:	The integral part of the fn index value for which interpolation is desired.
ilb:	The integral part of the lb index value for which interpolation is desired.

ires: Index that indicates whether lb, cb, or fn inputs are out of table interpolation ranges (ires=1 indicates inputs are out of range; ires=0 indicates inputs satisfactory).

j: The Froude number (fn) index of the residual resistance coefficient array.

k: The cb index of the residual resistance coefficient array.

lb: L/B. Ratio of barge length to barge molded breadth.

lbl: Value of the linear interpolation between cbul and cbl1, the two lb values, where $bt = 2.46$.

lbu: Value of the linear interpolation between cbuu and cblu, the two lb values, where $bt = 2.76$.

lll: The quadratically interpolated value with respect to fn for the first of two cb values and the first of two lb values, where $bt = 2.46$.

llu: The quadratically interpolated value with respect to fn for the first of two cb values and the second of two lb values, where $bt = 2.46$.

lul: The quadratically interpolated value with respect to fn for the second of two cb values and the first of two lb values, where $bt = 2.46$.

luu: The quadratically interpolated value with respect to fn for the second of two cb values and the second of two lb values, where $bt = 2.46$.

m: The lb index of the residual resistance coefficient array.

ncb: The cb index for the first of two table index values that bracket the index of the desired cb value.

nf n: The fn index for the first of three table index values that bracket the index of the desired fn value.

nlb: The lb index for the first of two table index values that bracket the index of the desired lb value.

rcb: The cb index for the desired cb value.

rfn: The fn index for the desired fn value.

rlb: The lb index for the desired lb value.

ull: The quadratically interpolated value with respect
to fn for the first of two cb values and the
first of two lb values, where $bt = 2.76$.

ulu: The quadratically interpolated value with respect
to fn for the first of two cb values and the sec-
ond of two lb values, where $bt = 2.76$.

uul: The quadratically interpolated value with respect
to fn for the second of two cb values and the
first of two lb values, where $bt = 2.76$.

uuu: The quadratically interpolated value with respect
to fn for the second of two cb values and the
second of two lb values, where $bt = 2.76$.

y: The fractional part of the index used to repre-
sent the desired fn value, used as input into the
quadratic interpolation routine.

REFERENCES

American Bureau of Shipping (ABS)	1973	<u>Rules for Building and Classing Steel Barges for Offshore Service</u> , ABS, New York.
	1978	<u>Rules for Building and Classing Steel Vessels</u> , ABS, New York.
Comstock, J. P. (ed.)	1967	<u>Principles of Naval Architecture</u> , SNAME, New York.
Evans, J. H. (ed.)	1975	<u>Ship Structural Design Concepts</u> , Cornell Maritime Press, Cambridge, Md.
Fletcher, E. H.	1974	"ARTUBAR--Resume of Achievements and Developments," author's report, August 1974.
Horne, M. R.	1958	"The Full Plastic Moments of Sections Subjected to Shear Force and Axial Load," <u>British Welding Journal</u> (April).
Hukill, E. L.	1974	"The Development of an Integrated Tug-Barge Concept," SNAME Technical Report, Toronto, March 1974.
IMCO	1966	"International Conference on Load Lines, 1966," Inter-Governmental Maritime Consultative Organization, London.
Jenkins, W. M. de Jesus, G. C. Burns, A.	1977	"Optimum Design of Welded Plate Girders," <u>The Structural Engineer</u> , Vol. 55, (April).
Maritime Administration (MARAD)	1979	From the program notebook of the National Ocean-Going Tug-Barge Planning Conference held on 27-28 March at New Orleans.

Robinson, J. H.	1976	"Experimental Research Relative to Improving the Hydrodynamic Performance of Oceangoing Tug/Barge Systems. Summary Report," NSRDC Ship Performance Department Report No. SPD-518-05 (NTIS No. PB 263 539).
Seabulk Chemical Carriers	1976	"Technical and Operational Features of <u>Seabulk Magnachem/SCC 3902</u> ," company brochure.
Sharp, G. S.	1975	"Development of a Model for Economic and Operational Evaluation of Integrated Tug-Barge Systems," report for MARAD Contract No. 2-36285.
Triantafyllou, M. S.	1977	"Development of Analytical Methods in B-Series Propeller Design for Application in Computer Programs," Master Thesis M.I.T. Departments of Mechanical and Ocean Engineering.
Tsuchida, K. Koichi, Y. Yazaki, A. Shigeo, M. Seizo, O.	1969	"Systematic Resistance and Propulsion Tests With Models of Single-Screw Full-Bodied Oil Tankers," Japanese Ministry of Transport Ship Research Institute Report, Vol. 1, No. 6. Translated in Univ. of Michigan Depart. of N.A. and M.E. Report No. 039.
Waller, D. B. Filson, J. J.	1972	"Advanced Ocean Tug-Barge Systems: A Review of the State-of-the-Art of Ocean Barge Design and Operation," National Maritime Research Center Report No. NMRC-272-21000 RO (NTIS No. COM-74-10121).