LIST OF REFERENCES

The following abbreviations have been used in the references :

RINA	Royal Institution of Naval Architects (INA prior 1960)
NA	The Naval Architect Journal, RINA
SNAME	The Society of Naval Architects and Marine Engineers
MT	The Marine Technology, SNAME
NECIS	North East Coast Institution of Shipbuilders

1. Assereto, G., Shipbuilding and Shiprepairing Industries in Thailand., Office of the Board of Investment of Thailand, Bangkok, 1979.
2. Amy, J.R., Johnson, R.E. and Miller, E.R. Jr. "Development of Intact Stability Criteria for Towing and Fishing Vessels," SNAME, Vol. 84, 1976.
3. Abkowitz, M.A., Stability and Motion Control of Ocean Vehicles. The MIT Press, Cambridge, 1969.
4. Attwood, B.l. and Pengelly, H.S., Theoretical Naval Architecture. New Impressions Edition, Longmans, Green and Co.,Ltd., 1967.
5. Aertssen, G., Limiting Speed in Rough Seas. Proc. Van Lammeren Jubilee Memorials, 1970.

6. Aertssen, G. et.al, "Service Performance and Seakeeping Trials on a Stern Trawler" NECIS., 1966-67.

7. Bowden, K.F., "The Marine Environment-Some Features of Concern to Naval Architecture," RINA., Vol. 112, 1970.

8. Bales, S.L., Cummins, W.E. and Comstock, E.N., "Potential Impact of Twenty Year Hindcast Wind and Wave Climatology on Ship Design," MT., Vol.19, 1982.

9. Bhattacharyya, R., Dynamics of Marine Vehicles. John Wiley & Sons, New York, 1982.

10. Blagoveshchensky, S.N., Theory of Ship Motions., Vol. 1 & 2, New York, 1962.

11. Bales, N.K. and W.E. Cummins, "The Influence of Hull form on Seakeeping," SNAME., 1970.

12. Cunningham, D.B., "Notes on Trawl Fishing," Shipbuilding and Shipping Record., Oct. 1949.

13. Cleary, W.A. Jr., "Subdivision, Stability, Liability" MT., Vol. 19, 1982.

14. Comstock, J.P.(Editor), "Principles of Naval Architecture" SNAME., 1967.

15. Cleary, W. A. Jr., "Loadlines - the lever of safety" SNAME., 1975.

16. Cummins, W.E., Ship Motions or Design for Seakeeping Report Dept. of Naval Architecture and Marine Engineering U. of Michigan, 1972.

17. Dexter, Stephen C., Handbook of Oceanographic Engineering Materials., John Wiley & Sons, New York, 1982.

18. De Herre, R.F.S. and Bakker, A.R., Buoyancy and Stability of Ships., H. Stam, Holland, 1969.

19. Ewing, J.A., "The Effect of Speed, Forebody Shape and Weight Distribution on Ship Motions," RINA., 1967.

20. Weing, J.A. and Goodrich, J.G., "The Influence on Ship Motions of Different Wave Spectra and of Ship Length," RINA., 1967.

21. Fridman, A.L., Calculations for Fishing Gear Designs., FAO Fishing Manuals, Rome, 1976.

22. Fyson, J.F., Second Report to the Government of Thailand on Fishing Boats., FAO, Rome, 1970.

23. Ferdinande, V., "Model Tests in Regular and Irregular Waves at the Davidson Laboratory" RINA., 1963.

24. Henrickson, W.A., "Assessing Intact Stability," MT., Vol. 17, 1980.

25. Hooft, J.P., Advanced Dynamics of Marine Structures., John Wiley & Sons, New York, 1982.

26. Hamlin, N.A. and Compton, R.H., "Assessment of Seakeepability," MT., 1966.

27. IMCO Recommendation on Intact Stability for Passenger and Cargo Ships under 100 Metres in Length., Resolution A. 167, November 1968.

28. Jeager, H.E., "A Note on Large Trawlers," INA., Vol. 96, 1954.

29. Kim, C.H., Chou, F.S. and Tieu, D., "Motions and Hydrodynamic Loads of a Ship Advancing in Oblique Waves," SNAME., Vol. 88, 1980.

30. Koutitas, C.G., Mathematical Models in Coastal Engineering., 1st Edition, Pentech Press London, 1988.

31. Kinsman, B., Wind Waves., Prentice-Hall, Inc. Englewood Cliffs, N.J., 1965.

32. Lundgren, G.A. and Storch, R.L., "Small Fish Boat Stability : A Case Study," MT., Vol. 21, 1984.

33. Lloyd, A.R.J.M., SEAKEEPING : Ship Behavior in Rough Weather., 1st Edition, Ellis Horwood Ltd. England, 1989.

34. Martin, L.L., "Ship Maneuvering and Control in Wind," SNAME., Vol. 88, 1980.

35. McGowan, J.F. and Meyer, R.B., "Has Stability Delayed the Delivery of Your Tug?," MT., Vol. 17, 1980.

36. Muckle, W., "A Note on the Buoyancy of a Ship Amongst Waves," RINA., Vol. 107, 1965.

37. McCormick, M.E., Ocean Engineering Wave Machanics., John Wiley & Sons, New York, 1982.

38. Minikin, R.R., Winds, Waves and Maritime Structures., 2nd Edition, Charles Griffin and Company Ltd., London, 1963.

39. Tong Nadgratok, Boatbuilding in Thailand Labour-Intensive Construction Methods for Fishing Vessels., Technical Report No. 4, Fishing Vessel Development Section, Marine Fisheries Division, Dept. of Fisheries, Bangkok, 1981.

40. Tong Nadgratok, Fishing Boat : 1 General Information., Technical Paper No. 4, Fishing Vessel Development Section, Marine Fisheries Division, Dept. of Fisheries, Bangkok, Aug. 1982.

41. Nickum, G.C., The Stability of Fishing Vessels., Inter. Fishing Boat Congress, Paris 12-16 Oct. 1953.

42. Nickum. G.C., "An Evaluation of Intact Stability Criteria," MT., Vol. 15, 1978.

43. Newton, R.N., "Wetness Related to Freeboard and Flare," RINA., Vol. 102, 1960.

44. Norbby, R., "The Stability of Coastal Vessels," RINA., Vol. 104, 1962.

45. Nadeinski, V. and Jens, J., "The Stability of Fishing Vessels," RINA., Vol. 110, 1968.

46. Newman, J.N., Marine Hydrodynamics., The MIT Press, Cambridge, 1977.

47. Office of the Board of Investment, The Basic Study Report on Development of Shipbuilding and Repairing industry in the Kingdom of Thailand., Bangkok, July 1980.

48. Ochi, M.K., "Extreme Behavior of a Ship in Rough Sea Slamming and Shipping of Green Water," SNAME., Vol. 72, 1964.

49. OSCC, Fishing Boats., 1^{st} Edition, Japan International Cooperation Agency, 1977.

50. Prohaska, C.W., "Influence of Ship form on Transverse Stability," INA., 1951.

51. Penney, M.D., "Small Junks Built in Hong Kong," RINA., Vol. 104, 1962.

52. Pattullo, R.N.M., "The B.S.R.A. Trawlwer Series (Part 3) Beam-Drought and Length-Displacement Ratio Series Resistance and Propulsion Tests," RINA., Vol. 107, 1965.

53. Prohaska, C., "Residuary Stability," INA., Vol. 89, 1947.

54. Price, W.G. and Bishop, R.E.D., Probabilistic Theory of Ship Dynamics., 1^{st} Edition, Chapman and Hall, London, 1974.

55. Roorda, A., Small Sea-Going Craft & Vessels for Inland Navigation., N.V. De Techische Vitgeresij H. Stam-Haarleur, Holland, 1955.

56. Radwan, A.M. "A Different Method to Evaluate the Intact Stability of Floating Structures," MT., Vol. 20, 1981.

57. Robertson, J.B. et. al., "The New Equivalent International Regulations on Suddivision and Stability of Passenger Ships," SNAME., Vol. 82, 1974.

58. Rahola, J., The Judging of the Stability of Ship and the Determination of the Ninimum Amount of Stability., Helsingfors, 1939.

59. Robb, A.M., Theory of Naval Architecture., 1st Edition, Charles Griffin & Company Ltd., 1952.

60. Spanner, W.F., "Some Notes on the Design of Trawlers and Drifters with Particular Reference to Seaworthiness and Stability," INA., Vol. 89, 1947.

61. Simpson, D.S., "Small Craft, Construction and Design," SNAME., Vol. 59, 1951.

62. Storch, R.L., "Stability of Offshore Tugboats," MT., Vol. 9, 1972.

63. Steel, H.E., "The Practical Approach to Stability of Ships," RINA.,, Vol. 98, 1956.

64. Sarchin, T.H., and Goldberg, L.L., "Stability and Buoyancy Criteria for U.S. Naval Surface Ships," SNAME., Vol. 70, 1962.

65. Seuyonov, V., Static and Dynamics of the Ship., Peace Publishers, Moscow, 1960.

66. Silvester, R., Coastal Engineering., Vol. 1 2nd Impression Edition, Elsevier Scientific Publishing Company Oxford, 1979.

67. SEAFDEC, Fishing Techniques., 1[st] Edition, Japan International Cooperation Agency, 1975.

68. The Society of Naval Architects of Japan, 60[th] Anniversary Series Volume 6. Advances in Research on Stability and Rolling of Ships," 1960.

69. Sawnders, H.E. (Editor), Hydrodynamics in Ship Design., Vol. 1-3, SNAME., 1957.

70. Taylor, A.R., "Fishing Vessel Design," INA., Vol. 85, 1943.

71. McTaggart, R.G. and Gunderson, R.H., "On Damaged Stability of Drilling Vessels," SNAME., Vol. 81, 1973.

72. Thomson, G., "International Considerations of Intact Stability Standards," RINA., Vol. 112, 1970.

73. Vossers, G., "Behavior of Ships in Waves," International Shipbuilding Progress., Vol. 61, 1961.

74. Weinblum, G. and St. Denis, M., "On the Motion of Ships at Sea," SNAME., 1950.

75. Yamagata, M., "Standard of Stability Adopted in Japan," INA., Vol. 101, 1959.

76. ศูนย์พัฒนาการประมงแห่งเอเชียตะวันออกเฉียงใต้, "อนาคตประมงไทย" รายงานผลการสัมมนาร่วมภาครัฐบาลและภาคเอกชน., ตุลาคม 2530.

77. ไพฑูรย์ ขาวมาลา, การต่อเรือไม้ในประเทศไทย., วิทยาลัยการต่อเรือพระนครศรีอยุธยา, พ.ศ. 2523.

78. ทอง นัดกระโทก, สภาวะอุตสาหกรรมการต่อเรือในประเทศไทย ปี 2529., รายงานวิชาการที่ กพน. 01/2529, กองประมงทะเล, กรมประมง, กระทรวงเกษตรและสหกรณ์, กันยายน 2530.

APPENDICES

APPENDIX (A)

METHOD OF FULL-SCALE STABILITY SURVEYS

General

To find KG and displacement (D) of the vessel as built, an inclining test is normally performed. The designer is often responsible for its execution and must know the procedure. The steps during the exercise are :

Measure draughts fore, aft and amidships, check readings by plotting them on a lines plan or general arrangement (GA) of vessel. Note the difference between operating draught, including keel, and draught (moulded). Use hydrostatic data to find D. If the vessel trims, as it often does, the trim takes place around an axis through the centroid of the waterplane. The draught used when reading off D from the hydrostatic curve must be corrected to :

$$T = (T_F + T_A) \div 2 + (T_A - T_F - S) \times (X \div L) \; .... \; (A\text{-}1)$$

X = distance between midship station and the centroid of the waterplane (positive when aft of midships)

S = design trim (positive)

T_F, T_A = draught fore (F) and aft (A)

As dealt with in some detail later, the outmost case must be taken when determining KM for a vessel with substantial trim.

- Expose the vessel to a known external moment by several movements of weights changed from side to side and back to amidships. Measure inclination for every movement. Check relationship of inclination to moment, and calculate the metacentric height, GM.

- If free surfaces of liquids are presented in tanks (free surface should be avoided by topping up tanks if possible), decrease GM (M is lowered) with the correction δBM = δGM found according to chapter (2).

- Find the D and KG for the vessel as inclined by using the relation that GM = dMe/(Δ x dϕ) .... (A-2)

- By resolving moments, deduct weight and moments of items that do not belong on board for the light vessel condition, ie, inclining weights, personnel, liquid in tanks, other items, and add items that have not yet been brought on board (ie, fishing gear, etc). Arrive at the final KG and D for the light condition, filling out the form shown and task is completed. The rolling test, which is an approximate way of finding GM, will be dealt with later.

<u>The practical execution of the inclining experiment</u>

As the intention of the experiment is to find KG and D of the completed vessel, fully equipped but without crew provisions, liquid in tanks and cargo, it should be as close to the light condition as possible. This being not always possible, all alien weights must be known as regard weight and position. This implies sounding of all tanks. Correspondingly, missing weights must be noted. This being done, the vessel must be positioned at its berth so that heeling will not be restrained in any way. Then, draught readings forward, amidships and aft must be taken. If no draught marks are on the vessel, the heights from the top of the working deck at the same positions are measured. Then, these measurements are plotted on the GA or the lines plan to find the draughts, and to see if they agree.

Then, weights are moved from amidship to one side-back amidships - then to the other side. The movement of two separate weights is recommended - one of them first, then both.

The heeling angle is found by using a pendulum with length "1", suspended, preferably into the hold, and free to move. the heeling angle (in radians) is found as seen from Fig. (A-1)

Then

$\emptyset$ = a/f (radians)

= 57.3 x a/f (degrees)

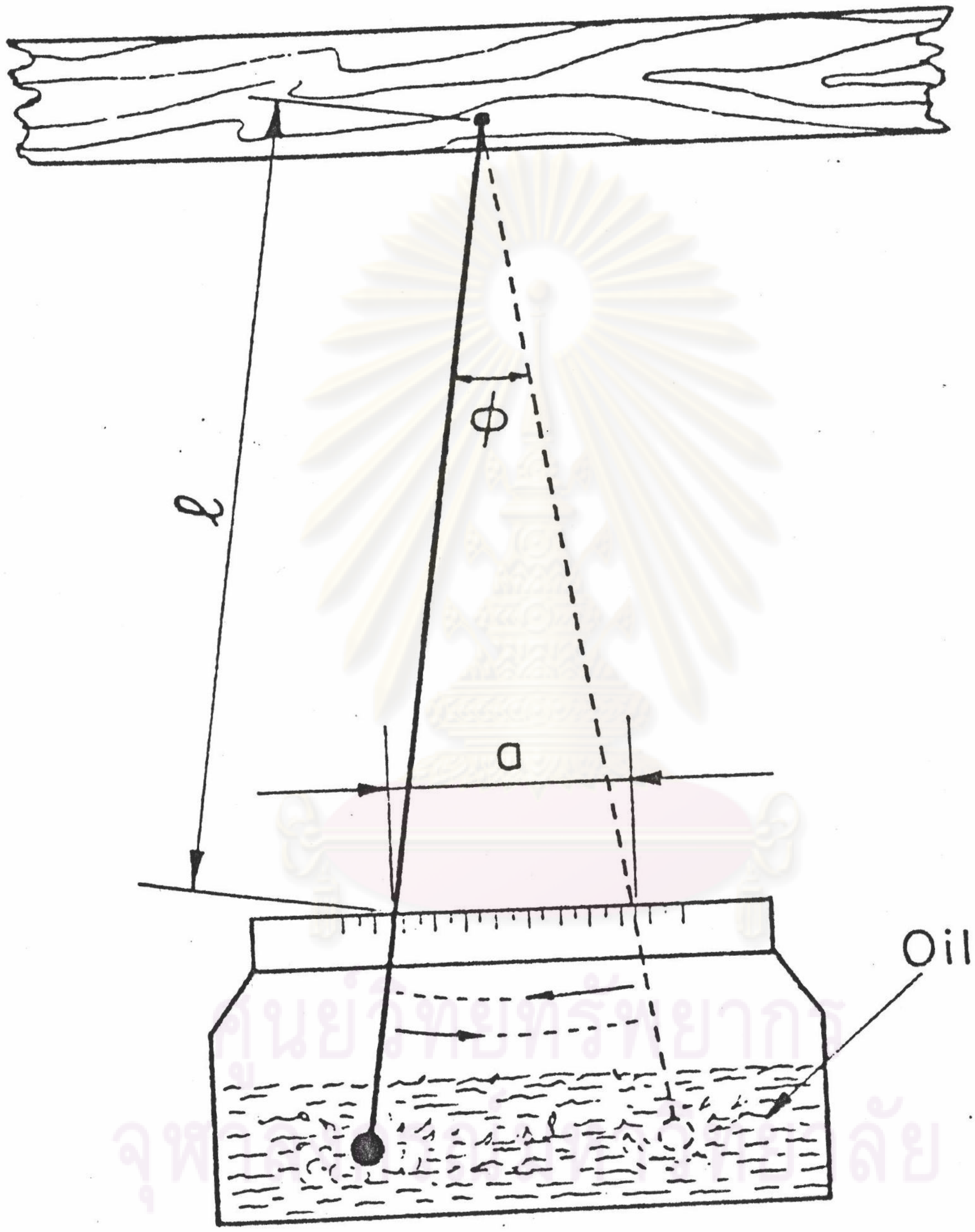


Fig (A-1) Suspended pendulum

It is advisable to suspend the small weight at the end of the pendulum into oil or some other kind of liquid to dampen oscillations.

The weight to be used must produce a reasonable heeling angle $d\phi max$. They can be determined by estimating D and GM.

If the moment arm is B/2 and $d\phi max$ = 10 degree, we obtain

$$dMe_{max} = D \times B/2 = GM \times D \times d\phi_{max}$$

$$W_{max} = (2 \times GM \times D \times 10^\circ) / (B \times 57.3^\circ)$$

$$= 0.35 \times (GM \times D) / B$$

The estimation of GM can be by preliminary weight and KG calculation as dealt with previously, or a rolling test can be performed prior to the inclining experiment. Assuming GM 0.05 m, B = 4.0 m and D = 30 t, the inclining weight should be about 1.3 tons.

This could be any well-defined weight. If nothing else is at hand, a barrel can be used. It can be filled with water until a heeling angle of, say, 10 degree is reached, then emptied and moved to the opposite side of the vessel, and then refilled, both half-full and full.

It is important that one person is in charge of the experiment. The plots of heeling angle versus heeling moments, must be checked for each weight movement by the person in charge.

and decreased in the .pa fore body. The consequence is physical increase in I_L, and thus in GM. If we use the KM for the vessel without trim for the evaluation of KG, ie.

$$KG = KM \text{ (from hydrostatic curve without trim)} - GM \text{ (including experiment)} \quad \ldots\ldots\ldots \quad (A-3)$$

Then KG will be too low or too optimistic. Vessels often trim by the stern at the inclining experiment. If the trim L/100, it is advisable to calculate I_L, for the waterline, and to correct KG.

Again, it is important that the person in charge understands the physical background of the inclining represented by equation (A-2)

The form shown is a good example of systematic collection of all data needed from the experiment. Also, on the last pages of the form, the calculations preceding the experiment can be performed in an orderly manner.

Then form is otherwise self-explanatory.

Accuracy

The accuracy of the inclining test and of the stability calculation is dealt with in an IMO document, Reference [27].

Apart from the features already mentioned, the following points are made :

- All measurements should be taken with care, and be checked as the experiment proceeds
- Special emphasis should be placed on draught readings and control of heeling weights
- The natural period for the pendulum (for reading heeling angle) and the roll period of the vessel should not be close to each other
- At least 4 readings of moment versus heeling angle should be taken
- The maximum heeling angle should be in the range 5 - 10 degree.

The possibility of negative initial stability should not be ruled out. This manifests itself through an angle of loll to one side or the other.

An inclining test cannot be meaningful in such a case because a false GM will be measured.

The reason for taking trim seriously at the inclining experiment is evident from the studying. With trim be the stern, the waterline alert. Normally, the breadth of the waterline is increased in the after body

Report form on inclining experiment

1. General information

(a) Shipyard : ____________

Yard No. and/or vessel name : ____________

Signal letters : ____________

(b) Date and place of experiment : ____________

Time commenced : ________ Time completed : ________

Berth and mooring : ____________

Weather : ________ Sea : ________ Wind : ________

Specific gravity of water : ____________

(c) Person in charge : ____________

(d) Attending surveyor : ____________

2. Principal dimensions

(a) Overall length L_{OA} : ________ m

Standard waterline length L : ________ m

Breadth moulded B : ________ m

Depth moulded D : ________ m

Designed trim on L : ________ m

(b) Draught during experiment (operational):

Draught aft da : ________ m

Draught fore df : ________ m

Draught mid dm : ________ m

(c) Trim (T) = (da - df) - designed trim on L : ________ m

(d) Displacement in metric tons on even keel from displacement curve Δ : ________ t

Corrected for specific gravity of water : ________ t

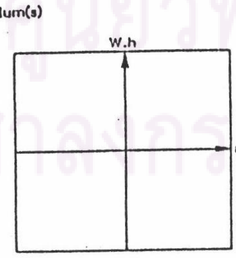
Corrected for trim : ________ t

3. Execution of the inclining experiment and calculations

Shifting weights consist of : ________

Length of pendulum(s) ℓ : ________ m

Position of pendulum(s) : ________



Pendulum swings (a) plotted against shifting moments (W.h)

Weights shifted from starboard to port side

Shifting weight No.	Weight W (metric tons)	Shifting distance h (m)	Shifting moment W.h (mt)	Pendulum swing a (cm)
1				a_1 =
2				a_2 =
3				a_3 =
4				a_4 =
Σ	W_I =		M_I =	a_I =

3. (Continued)

Weights shifted from port to starboard side

Shifting weight No.	Weight W (metric tons)	Shifting distance h (m)	Shifting moment W.h (mt)	Pendulum swing a (cm)
1				a_1 =
2				a_2 =
3				a_3 =
4				a_4 =
Σ	W_{II} =		M_{II} =	a_{II} =

Transverse metacentric height (GM) of ship as inclined :

$$GM = \frac{(M_I + M_{II}) \cdot 100 \cdot \ell}{\Delta (a_I + a_{II})}$$ = ________ m

Transverse metacentre above reference line (KM) at equivalent draught = ________ m

KM corrected for trim = ________ m

Correction for the free surface effect in tanks (gm)

Tank	Volume (m^3)	Transverse moment of inertia of the liquid surface (I) (m^4)	$\gamma (t/m^3)$	$\gamma \cdot i$
Σ				

$gm = \frac{\Sigma(\gamma \cdot i)}{\Delta}$ = ________ m

Centre of gravity above K :

KG = KM - (GM + gm) = ________ m

4. Trim calculations

Moment to change trim one cm (MT 1) at equivalent draught = ________ m

Centre of buoyancy fore/aft of amidships (LCB) = ________ m

Horizontal distance between the centre of gravity and the centre of buoyancy : T · MT 1 · 100 ÷ Δ = ________ m

Centre of gravity fore/aft of amidships (LCG) = ________ m

5. Calculation of lightweight of vessel

Item	Weight (metric tons)	KG (m)	Mv	LCG (m)	MH
Vessel at experiment					
- Surplus load					
+ Missing loads					
Lightweight					

6. Additional comments

____

APPENDIX B

COMPUTER PROGRAM TO DETERMINE HYDROSTATIC/DYNAMICAL STABILITY PARTICULARS

```
CONST NumStation = 35;
     NumWl      = 30;
     NumCal     = 2;
     NoOfM      = 8;
     degsp      = 5;
     Roll       = 1.0252;
TYPE NUMS       = 0..Numstation;
     Numw       = 0..NumWl;
     Numc       = 0...NucCal;
     VIAR       =array]Nums] of integer;
     XIAR       =array]Nums] of real;
     YIJAR      =array]Nums,Numw] of real;
VAR       L, Bmld, H,Wl,Dis,Draft,
     BG                                        :real;
     B,C,BB,CBB,DBB                            :char;
     i,j,Ma,Mn,Mf,Ml,Ns,Nw,m                   :integer;
     X,Z,UU,WW,Cur,Awp,Mmid,Imid,Icl,LCF,
     Ina,VolV,MVk,KB,BMt,GBMl,KMt,KMl,VolL,
     Mvmid,LCB,DISPT,DISPTst,TPC,MCT1m,A1,
     A2,AT,AWV1,AWV2,GZ,LDelOH,LDelOHcos,
     Mcl1,Mcl2,MclT,CRP,Del1,Del2,DelDif,BR,
     DelOH,W,Work                              :XIAR;
     Y,Ast,R1,R2,MR1,MR2,LDelR                 :YIJAR;
     kk,ll                                     :VIAR;
PROCEDURE Principal_Dimension;
```

```
BEGIN Clrscr;
    Writeln('   LBP ?');
    readln (L);
    Writeln('  BMLD ?') ;
    readln(Bmld);
    Writeln('  DRAFT ?') ;
    readln(H) ;
    Writeln('  Number of Stations ?');
    readln(Ns) ;
    Writeln(Nw) ;
    Writeln('  Water line Apart ?');
readln(Wl)
END;


PROCEDURE Position_To_Input_Half_Breadths;
LABEL   10,20,40,50;
BEGIN   Clrscr;
i : = 0;
Writeln('Position to Input Half Breadths' :50);
Writeln;
10:     Writeln('   Distance from AP (Aft) in m ?');
        Readln(S]i]);
        If X]i] = 0  then
              BEGIN
Ma  : = i;
i  :  = i+1;
```

```
goto 20
              END
           else
              BEGIN
                    i := i+1;
goto 10
              END ;
20:     Writeln('     Station ?);
        readln (X]i]) ;      0
If X]i] = (  ns/2 ) then
BEGIN
    Mm  := i;
    i   := i+1;
    goto 20
END
    else if X]i] = Ns then
    BEGIN
Mf:= i;
i := i+1;
goto 40
    END
    else
    BEGIN
i := i+1;
goto 20
    END ;
40:     Writeln('   Distance from FP (Forward) in m ?');
        readln (X]i]);
```

```
        Writeln('    More Station ]Y/N]') ;
repeat B := readkey;until B in ]'y','n'];
        If B = 'n'  then
                BEGIN
Ml := i ;
goto  5o
                END
              else
                BEGIN
                       i :=i+1;
goto 40
END;
50:     END;

PROCEDURE Distance_from_Midship;
BEGIN
FOR i := 0 to Ma do Z]i] := -(X]i]+L/2);
For i := Ma+1 to Mf do Z]i] := -(Ns/2 - X]i]*L/Ns;
For i := Mf+1 to Ml do Z]i] := L/2 + X]i]
END;


PRODUCEDURE Input_Half_Breadths;
BEGIN   Clrscr;
FOR j = 0 to Nw do
BEGIN
    Clrscr;
    FOR i := 0 to Ma-1 do
```

```
   BEGIN
Writeln('    Input Half Breadths of WATERLINE ',
   WL*j:7:2,' M.');
Writeln;
Writeln('Half Breadth of Waterline ;,Wl*j:7:2,' m.',
   'at STATION ',X]i]:5:2,' m. aft AP');
Read ln (Y]i,j])
   END;
   Writeln;
   FOR i := Ma to Mf do
   BEGIN
Writeln ('Half Breadth of waterline ',Wl*j:7:2,
   'm. at STATION ', X]i]:5:2);
Readln (Y]i,j])
   END;
   Writeln;
   For i := Mf+1 to Ml do
   BEGIN
Writeln ('Half Breadth of Waterline ',Wl*j:7:2 ,
,m. at STATION ',X]i]:5:2 ,' m. fwd FP');
Readln (Y]i,j])
    END
END
END;
```

```
PROCEDURE Input_Station_of_Zero_Half_breadth;
BEGIN
    Clrscr;
    FOR j:= 1 to Nw do
    BEGIN
Writeln ('Input DISTANCE from midship to position of zero',
'half breadth in m. of WATERLINE', Wl*j:7:2,' m.');
Writeln ('Station of zero half breadth at AFT BODY');
Readln (WW]j]);
Writeln ('Station of zero half breadth at FORE BODY');
Readln (UU]j]);
WRITENLN;
    END;
END;
```

```
PROCEDURE Find_Initial_Station_to_be_calculated;
BEGIN
    FOR j:= 1 to Nw do
    BEGIN
    FOR i:= 0 to Mm do
    BEGIN
If (( Abs(Z]i] - WW]j]) > 0 ) And
(( Abs(Z]i+1]) - WW]j] < 0) Or
```

```
(( Abs(Z]i]) - WW]j]) = 0) then kk]j] := i ;
END;
FOR i:= M1 downto Mm do
BEGIN
If (( Abs(Z]i] - UU]j]) > 0) And
(( Abs(Z]i-1]) - UU]j]) < 0) Or
(( Abs(Z]i]) - UU]j] = 0) then ll]j] := i ;
END;
END;
END;




PROCEDURE LonglInt (V : YIJAR; m,n :VIAR;VAR Cur:XIAR);
VAR dCur                                              :XIAR;
o,p,q                                          : integer;
s,axi,axo,axp,codet,ac,be,cc,de,za,zv,zc : real;
LABEL 8;
BEGIN
    If m]j] = n]j] then
    Begin
    Cur]j] := 0;
    goto 8;
    End ;
I   : = m]j];
```

```
Repeat dCur]I] := 0; inc (I) Until I n]j];
I := m]j];
WHILE ( I - Mm ) < 0 do
BEGIN
     o := I+1;
     p := I+2;
     q := I+3;

     CASE (Mm - I) of
1 : begin
s := Abs (Z]i] - X]0]);
dCur]I] := s/2* (V]I,  j] + V]o,j]);
I := I + 1
     end;
2 : begin
(axi := 0;
axo  := Abs(Z]I] - Z]o]);
axp  := Abs(Z]I] - Z]p]);
codet := axo*axp* (axp-axo);
ac  := (V]I,j}*axo*axp*(axp-axo))/codet;
bc  := ( (sqr(axp)*(V]o,j]-V]I,j])) +
(Sqr(axo)*(V]I,j] - V]p,j])) )/ codet;
cc := ( axo*(V]p,j] - V]I,j])
+ axp* (V]I,j] - V]o,j]) )/codet ;
dc :+ axp/2;
v]o,j] := ac + bc*dc + cc*Sqr(dc);
s := dc;
```

```
dCur]I] := s/3*(V]I,j] + 4*V]o,j] + V]p,j]);
I := i + 2
axi := 0;
axo := Abs(Z]I] - Z]o[);
axp := Abs(Z]I] - A]p]);
codet := axo*axp*(axp-axo);
be := ( (V]o,j]-V]I,j]*Sqr (axp)
- (V]p,j] - V]I,j]*Sqr (axo))/codet;
cc := ( - (axp*V]o,j] - V]I, j]))
- (axo*(V[I,j] - V]p, j])) )/codet;
ac := V]I, j];
dc := axp/2;
v]o,j] := ac + bc*dc + cc*Squ(dc);
s := dc;
dCur]I] := s/3*( V]I,j] + 4*V]o,j] + V]p,j]);
I := i + 2;

end
ELSE
     za := (Z]i] - Z]o]);
     zb := (Z]o] - Z]p]);
     zc := (Z]p] - Z]q]);
     If za <> zb then
begin
(axi := 0;
axo  := Abs(Z]I] - Z]o];
axp  := Abs(Z]I] - z]p]);
```

```
elseif zb <>  zc then
begin
s := Abs (z]I] - Z]o]);
dCur]I] := s/3*( V]I,j] + 4*V]o,j] + V]p,j] );
I := I + 2
end
else
begin
      s := Abs (Z]I] - Z]o);
      dCur ]I] := s*3/8* ( V]I,  j] + 3*V]o,j]
      + 3*V]p,j] + V]q,j] );
      I := I + 3
end
end (CASE)
END; {WHILE}
      I : = n]j]; { Integration of Fwd Body }
      WHILE ( I = Mm ) > 0 do
BEGIN
      o := I-1;
      p := I-2;
      q := I-3;
      CASE (I - Mm ) of
1 : begin
s := Abs(Z]I] - Z]O]);
dCur]I] := s/2* (V]I,j] + V]o,  j]) ;
I := I - 1
end;
```

```
2 : begin
{
axi := 0:
axo := Abs (Z]I] - Z]o]);
axp := Abs (Z]I] - Z]p]);
codet := axo*axp*(axp-axo);
ac := ( V]I,j]*axo*axp*(axp-axo))/codet;
bc := ( (Squ(axp)*(V]o,j] - V]I,j]) +
(Sqr (axo)*(V]I,j] - V]p,j])) )/codet;
cc := ( axo*(V]p,j] - v]I,j})
+ axp*(V]I,j] - V]o,j]) 0/codet;
dc := axp/2;
v]o,j] := ac + bc*dc + cs*Sqr(dc);
s := dc;
dCur]I] := s/3*(V]I,j] + 4*V]o,j] + V]p,j]);
I := i - 2
)
                        axi ;= 0;
axo := Abs(Z]I] - Z]o]);
axp := Abs(Z]I] - Z]p]);
codet := zso*axp*(axp-axo);
bc := ( (V]o,j] - v]I,j])*Sqr(axp)
-(V]p,j] - V]I,j])*Sqr(axo))/codet;
cc := ( -(axp*(V]o,j] - V]I,j]))
-(axo*(V]I,j] - v]p,j])) )/codet;
ac := V]I,j] ;
dc := axp/2;
v]o,j] := ac + bc*dc + cc*Sqr(dc);
```

```
s := dc;
dCur]I] := s/3*( V]I,j] + 4*V]o,j] + V]p,j]);
I := i - 2;

end;
ELSE
        za := (Z]I]-Z]o]);
        zb := (Z]o]-Z]p]);
        zc := (Z]p]-Z]q]);
        If za <> zb tjem
begin
{
axi := 0;
axo := Abs(Z]I] - Z]o]);
axp := Abs(Z]I] - Z]p]);
codet := axo*axp*(axp-axo);
ac := ( V]I,j]*axo*axp*(axp-axo))/codet;
bc := ( (Sqr(axp)*(V]o,j]-V]I,j])) +
(Sqr(axo)*(V]I,j]-V]p,j]) ))/ coddt;
cc:= ( axo*(V]p,j]-V]I,j]) )
+ axp*(V]I,j]-V]o,j])/codet;
dc := axp/2;
v]o,j] := ac + bc*dc + cc*Sqr(dc);
s := dc;
dCur]I] := s/3* ( V]I,j] + 4*V]o,j] + V]p,j]);
I := I - 2
}
                                  axi := 0;
```

```
axo := Abs(Z]I] - Z]o]);
axp := Abs(Z]I] - Z]p]);
codet := axo*axp*(axp-axo);
bc := ( (V]o, j]-V]I, j]*Sqr(axp)
-( V]p, j]-V]I, j])*Sqr(axo))/codet;
cc := ( - (axp*(V]I, j]-V]I, j]))
-(axo*(V]I, j]-V]p, j])) )/codet;
ac := V]I, j} ;
dc := axp/2;
v]o, j] := ac+ bc*dc + cc*Sqr(dc);
s := dc;
dCur]I] := s/3*( V]I, j] + 4*V]o, j] + V]p, j]);
I := i - 2;

end
elseif zb <> zc when
        begin
        s := Abs (Z]I-Z]o);
dCur]I] := s/3* ( V]I, j] + 4*V]o, j] + V]p, j] );
        I := I - 2
        end
else
begin
        s := Abs(Z]I] - Z]o]);
        dCur]I] := s*3/8* ( V]I, j] + 3*V]o, j]
        + 3*V]p, j] + v]q, j] );
        I : = I - 3
end
```

```
end {CASE}
END;  {WHILE}
Cur]j] := 0;
FOR I := m]j] to n]j] do
Cur]j] := Cur]j] + dCur]I];

8 : end;  {PROCEDURE}



PROCEDURE VerlInt ( Vv : XIAR ; VAR CCur : XIAR );
VAR  m,n,o,p,q  : interger;
     dCCur      : XIAR;
BEGIN
CCur]j] := 9;
CASE j of
     0 : CCur]j] := 0;
     1 : CCur]j] := W1/2*(Vv]j] + Vv]0]);
     ELSE
     m : = 0;
     Repeat dCCur]m] := 0: Inc (m) until m =j;
     Case (j mod 2) of
     0 : begin
     m := 0;
     While m<j do
     begin
     o := m+1;
     p := m+2;
```

```
dCCur]m] := W1/3*( Vv]m] + 4*Vv]o] + Vv]p]);
CCur]j] :+ CCur]j] + dCCur]m];
m := m + 2
     end     {While}
        end;  {Case j mode 2 of 0}
     Else
     m := 0;
     o := 1;
     p := 2;
     q := 3;
     dCCur]m] := 3/8*W1*( Vv]m] + 3*Vv]o] + 3*Vv]p] + Vv]q]);
     m := m + 3;
     While m<j do
     Begin
 o := m + 1;
 p := m + 2;
                dCCur]m] := W1/3*( Vv]m + 4*Vv]o] + Vv]p])]
 m : m + 2
      End;  {While}
      for m := 0 to j do CCur]j] := CCur]j] + dCCur]m]
 End  {Case}
 END  {CASE}
 END;  {PROCEDURE}
```

```
PROCEDURE Calculation;
VAR  YM,YMI,YMIcl,YIcl,LMVolL                   : YIJAR;
     m,ii,iii                                   : integer;
     YV,HAjst,HAwp,HMmid,Himid,HIcl,lMVk        : SIAR;
LABEL 4,5,9;
BEGIN
     FOR j := o to Nw do
     BEGIN
longlInt (Y,kk,ll, HAwp);
Awp]j] := 2*HAwp]j];
For i := kk]j] to ll]j] do
YM]i,j] : = Z]i]Iy]i,j];
LonglInt (YM,kk,ll, HMmid);
Mmid]j] := 2*HMmid]j] ;
For i := kk]j] to ll]j] do
    YMI]i,j] := Z]i]*YM]i,j];
longlInt 9YMI,kk,ll HImid);
Imid]j] := 2*HImid]j];
for i := kk]j] to ll]j] do
YIcl]j,j] := Sqr (Y]i,j])*Y]i,j]/3;
LonglInt (YIcl,kk,ll,HIcl);
Icl]j] := 2*HIcl]j];
If Awp]j] = 0 then
Begin
LCF;j; := 0;
goto 9;
End;
```

```
LCF]j] := Mmid]j]/awp]j];
9 : Ina]j] := Imid]j] - Awp]j]*Sqr(LCF]j]);
VerlInt (Awp, VolV );
lMVk]j] := j*W1*Awp]j];
VerlInt ( lMVk, MVk);
     If VolV]j = 0 then
Begin
KB]j] := 0;
BMt]j] := 0;
KMt]j] := 0;
BMl]j] := 0;
KMl]j] := 0;
goto 4
End;
     KB]j] := MVk]j]/VolV]j];
BMt]j] := Icl]j]/VolV]j];
KMt]j] := KB]j] + BMt]j];
BMl]j] := Ina]j]/VolV]j];
KMl]j] := KB]j] + Bml]j];
4 : For i := 0 to Ml do Ast]ji,j] := 0;
For i := kk]j] to ll]j] do
Begin
     for m := 0 to j do
     begin
Yv]m] := Y]i,m];
VerlInt (Yv, HAjst);
Ast]i]j] := 2*HAjst]j]
end
```

```
     End;
     LonglInt (Ast,kkll, VolL);
     for i := kk]j] to ll]j] do
     LMVolL]i,j] := z]i]*Ast]i,j];
     LonglInt (LMVolL,kk,ll, Mvmid);
     If VolL]j] = 0 then
Begin
     LCB]j] := 0;
     goto 5
End;
     LCB]j] := Mvmid]j]/VolL]j];
5 : DISPT]j] := VolV]j]*Roll;
     DISPTst]j] := volL]j]*Roll;
     TPC]j] := Roll*Awp]j]/100;
     MCT1M]J] := DISPT]j]*BMl]j]/L
     End
          End;

PROCEDURE ResultOfHydrostaticData;
VAR BBo  :char;
BEGIN
ClrScr;
     Writeln (' Hydrostatic Data' :46);
     Writeln;
     Writeln;
```

```
Writeln('____________________________',
'__________________________________________________');
Write ('DRAFT' :10);
Write ('DISPT' :7);
Write ('KB' :5);
Write ('BMt' :6);
Write ('BMl' :7);
Write ('LCB' :7);
Write ('LCF' :7);
Write ('MCT1m' :8);
Writeln ('WParea' :10);
Write ('fwd mid' :8);
Write ('m.' :8);
Write ('Tons' :8);
Write ('m.' :6);
Write ('m.' :6);
Write ('m.' :7);
Write ('m.' :7);
          Write ('m.' :7);
Write ('Ton/cm' :9);
Write ('Ton.m/m' :8);
Writeln ('sqr m' :9);
          Writeln  ('______________________________',
'______________________________________');
FOR j := 0 to Nw do
     Begin
```

```
Write (W1*j:9:2);
Write (DISPT]j]:8:2);
Write (KB]j]:6:2);
Write (BMt]j]:6:2);
Write (BM1]j]:7:2:);
Write (LCB]j]:7:2);
Write (LCF]j]:7:2);
Write (TPC]j]:8:3);
Write (MCT1m]j]:8:2);
Writeln (Awp]j:9:2);
     End;  {Fro}
          Writeln ('____________________________________________________',
'_____________________________________________________________________');

Writeln ('Do you want to print the result ?');
repeat BB :+ readkey;until BB in ]'y','n'];
     If BB = 'y' then
     BEGIN
Writeln (LST,' Hydrostatic Data' :46);
Writeln (LST);
Writeln (LST);
Writeln (LST,'________________________________________',
'___________________________________________________________________________');
Writeln (LST);
 'DRAFT' :10);
Write (LST, 'DISPT' :7);
Write (LST, 'KB' :5);
```

```
Write (LST, 'BMt' :6);
Write (LST, 'BMl' :7);
Write (LST, 'LCB' :7);
Write (LST, 'LCF' :7);
Write (LST, 'TPC' :8);
Write (LST, 'MCT1m' :8);
Writeln (LST, 'WParea' :10);
Write (LST, 'fwd mid' :43);
Writeln (LST, 'fwd mid' :8);
Write (LST, 'm.' :8);
Write (LST, 'Tons' :8);
Write (LST, 'm.' :6);
Write (LST, 'm.' :6);
Write (LST, 'm.' :7);
Write (LST, 'm.' :7);
        Write (LST, 'm.' :7);
Write (LST, 'Ton/cm' :9);
Write (LST, 'Ton.m/m' :8);
Writeln (LST, 'sqr m' :9);
        Writeln (LST, '_____________________________________________',
'_______________________________________');
Writeln (LST);
FOR j := 0 to Nw do
    Begin
Write (LST, Wl*j:9:2);
Write (LST, DISPT]j]:8:2);
Write (LST, KB]j]:6:2);
Write (LST, BMt]j]:6:2);
```

```
Write (LST, BM1]j]:7:2);
Write (LST, LCB]j]:7:2);
Write (LST, LCF]j]:7:2);
Write (LST, TPC]j]:8:3);
Write (LST, MCT1m]j]:8:2);
Writeln (LST, Awp]j]:9:2);
     End;  {For}
                  Writeln (LST,'_______________________________',
'__________________________________________________',
End;   {If}
END;



PROCEDURE InputData;
BEGIN
     Clrscr;
     Writeln ( 'STABILITY CHECKING' :47);
     Writeln ( ' Displacement ?');
     Readln  (Dis ) ;
     Writeln ( ' Draft ?');
     Readln ( Draft );
     Writeln  (' BG ?');
     Readln ( BG );
```

```
      FOR i := 0 to (Ma-1) do
 Begin
      Clrscr;
      Writeln ( 'Input Radius' :42);
      For m := 0 to NoOfM do
 begin

      Writeln ( 'EMMERGE SIDE at Station ', X]i]:5:2,
'm. aft AP ', m*degsp,' degree');
Readln ( R1]i,m] );
Writeln ( 'IMMERGE SIDE at Station ', X]i]:5:2,
'm. aft AP ', m*degsp,' degree');
Readln ( R2]i,m] );
end;
      End;
      FOR i := Ma to Mf do
      Begin
           Clrscr;
      Writeln ( 'Input Radius' :42);
      For m := 0 to NoOfM do
begin
      Writeln ( ' EMMERGE SIDE at Station ', X]i]:5:2,
' and ', m*degsp, ' degree');
      Readln ( R1]i,m] );
      Writeln ( ' IMMERGE SIDE at Station ', X']i]:5:2,
'and ', m*degsp, ' degree');
      Readln ( R2]i,m] );
end;
```

```
End;
FOR i := (Mf + 1) to Ml do
Begin
     Clrscr;
     Writeln ( 'Input Radius' :42);
For m := 0 to NoOfM do
begin
     Writeln ( 'EMMERGE SIDE at Station ', X'i':5:2,
'm. fwd FP ', m*degsp,' degree');
     Readln ( R1]i,m] );
     Writeln ( 'IMMERGE SIDE at Station ', X]i]:5:2,
'm. fwd FP ', m*degsp, ' degree');
     Readln (R2]i,m] );
end;
     End;
END;
```



```
PROCEDURE RadialInt ( Vv :XIAR ; VAR CCur : XIAR );
VAR k,n,o,p,q  : interger;
     dCCur     : XIAR;
BEGIN
CCur]m] := 0;
CASE m of
     1 : CCur]m] := degsp/2* ( Vv]m] + Vv]0] 0;
```

```
ELSE
k := 0;
Repeat dCCur]k] := 0; Inc(k) until k]j;
Case (m mod 2) of
0 : begin
K := 0;
While k<m do
begin
O := k+1;
p := k+2;
dCCur]k] := degsp/3*( Vv]k] + Vv]k] + 4*Vv]o] + Vv]p]);
CCur]m] := CCur]m] + dCCur]k];
k := K +2;
    end ;      {While}
      end;       {Case m mod 2 of 0}
Else
    k := 0;
    o := 1;
    p := 2;
    q := 3;
dCCur]k] := 3/8*degsp*( Vv]k] + 3*Vv]o] + 3*Vv]o] + 3*Vv]p + Vv]q]
k := k + 3;
While k<m do
Begin
o := k + 1;
p := k + 2;
          dCCur]k] := degsp/3* ( Vv]k] +4*Vv]o] + Vv]p];
k := k + 2;
```

```
LDelOH]0] : LDelOH]j];
LDelOHcs]0] := LDelOH]0]*cos(degsp*NoOfM*pi/180);
FOR m := 1 to NoOfM do
Begin
     LonglInt (R1, kk, ll, A1);
A1]m] := A1]j];
LonglInt (R2,kk,ll,A2);
A2]m] := A2]j];
AT]m] := A1]m]+A2]m];
For i := kk]j] to ll]j] do
Begin
     MR1]i,m] := Sqr(R1]i,m])/2;
     MR2]i,m] := Sqr(R2]i,m])/2;
     LDelR]i,m] := (sqr(R1]i,m])*R1]i,m] + Sqr(R2]i,0]*R2]i,m])/3;
End;
     LonglInt(MR1, kk, ll, AWV1);
AWV1m] := AWV1]j];
LonglInt (MR2, kk, ll, AWV2);
AWV2]m] := AWV2]j];
Longlint (LDelR, kk, ll, LDelOH);
LDelOH]m] := LDelOH]j];
LDelOHcos]m] := LDelOH]m]*cos(degsp*(NoOfM-m)*pi/180);
Mcl1]m] := AWV1]m];
Mcl2]m] := AWV2]m];
MclT]m] := Mcl2]m]-Mcl1]m];
CRP]m] := MclT]m]/AT]m];

RadialInt (AWV1,Del1);
```

```
 RadialInt (AWV2, Del2);
 DelDif]m] := Del2]m] - Del1]m];
 RadialInt(LdelOHcos,DelOH);
 BR]m] := ( DelOH]m] - DelDif]m]*CRP]m])/(Dis/Roll);
 GZ]m] := BR]m] - BG*Sqrt(1-SQr(cos(degsp*m)*pi/180));
      END;  {FOR}
 END;         {PROCEDURE}


 PROCEDURE StaticalStabResult;
 BEGIN
      clrscr;
      Writeln ( 'Statical Stability Result' :46);
      Writeln;
      Writeln;
      Writeln ('______________________________' :46);
      Writeln;
      Write ('Angle of Roll' :34)'
      Writeln ('GZ' :11);
      Write ('degree' :30);
      Writeln ('m.' :15);
      Writeln ('______________________________' :46);
      Writeln;
      FOR m := 0 to NoOfM do
Begin
Write (degsp*m :29);
Writeln (GZ]m] :16:2);
```

```
End;
     Writeln ('_____________________________' :46);
     Writeln;
     Writeln ('Do you want to print the result ?');
     repeat CBB := readkey;until CBB in ]'y','n'];
     If CBB = 'y' then
     BEGIN
clrscr;
Writeln(LST,'Statical Stability Result' :46);
Writeln(LST);
Writeln(LST);
Writeln(LST, '___________________________' : 46);
Writeln(LST);
     Write(LST, 'Angle of Roll' :34);
     Writeln(LST, 'GZ' :11);
     Write(LST, 'degree' :30);
     Writeln(LST, 'm.' :15);
Writeln (LST,'____________________________' :46);
Writeln(LST);
     FOR m :- 0 to NoOfM do
     Begin
Write(LST,degsp*m :29);
Writeln(LST,GZ]m] :16:2);
End;
     Writeln(LST, '________________________________' :46);
END;      {If}
END;
```

```
PROCEDURE DynamicalStab;
BEGIN
     Work]0] := 0;
     FOR m:= 1 to NoOfM do
     Begin
RadialInt (GZ,W);
Work]m] := W]m]*pi/180;
     End;

END;




PROCEDURE DynamicalStabResult;
BEGIN
     clrscr;
     Writeln ('Dynamical Stability Result' :50);
Writeln;
Writeln;
Writeln ('_________________________________' :55);
Writeln;
     Write ('Angle of Roll' :34);
     Writeln ('Area of GZ Curve' :20);
     Write ('degree' :30);
     Writeln ('m.rad' :21);
     Writeln ('_____________________________________' :55);
```

```
      Writeln);
      FOR m := 0 to NoOfM do
 Begin
 Write (degsp*m:30);
 Writeln (Work]m] :20:3);
      End;
      Writeln ('_________________________________' :55);
      Writeln;
      Writeln (Do you want to print the result ?');
      repeat DBB := readkey;until DBB in ]'y','n'];
      If DBB = 'y' then
 BEGIN
 clrscr;
      Writeln(LST, 'Dynamical Stability Result' :50);
 Writeln(LST);
 Writeln(LST);
      Writeln(LST, '___________________________________' :55);
      Writeln(LST);
      Write(LST, 'Angle of Roll' :34);
      Writeln(LST, 'Area of GZ Curve' :20);
      Write (LST, 'degree' :30);
      Writeln(LST, 'm.rad' :21);
      Writeln(LST, '___________________________________' :55);
      Writeln(LST);
      FOR m := 0 to NoOfM do
 Begin
 Write(LST,degsp*m:30);
 Writeln(LST,Work]m]:20:3);
```

```
End;
     Writeln(LST, '____________________________________' :55);
     Writeln(LST);
     END;      {If}
END;


PROCEDURE STABILITY_CHECKING;
VAR BBb,BBc          : CHAR;
BEGIN
     Writeln ('Do you want to check stability of the ship ?');
     repeat BBb := readkey;until BBb in ]'y','n'];
     If BBb = 'y' then
     Begin
     Writeln ('Please read DRAFT,DISPT,BG.');
     Writeln (' Are you ready ? ');
     repeat BBc := readkey;until BBc in ] 'y', 'n'];
     If BBc = 'y' then
begin
InputData;
StatisticalStabCal;
StaticalStabCal;
StaticalStabResult;
DynamicalStab;
DynamicalStabResult;
     end;
     End;
     END;
```

```
PROCEDURE Sample;
BEGIN
     L := 10;
     H := 3;
     Ns := 10;
     Nw := 6;
     Wl := 0.5;



    Ma : 1;
    X]0] : 0.2;
    X]1] : 0;
    X]2] : 1;
    X]3] : 2;
    X]4] : 3;
    X]5] : 4;
    X]6] : 5;
    X]7] : 6;
    X]8] : 7;
    X]9] : 8;
    X]10] : 9;
    X]11] : 10;
    X]12] : 0.2;
    Mm := 6;
    Mf := 11;
    Ml := 12;
```

```
FOR j := 1 to Nw do Y]0,j] := 0;
FOR j := 1 to Nw do Y]12,j] := 0;
FOR i := 1 to 11 do
Begin
For j := 1 to 6 do
begin
     Y]i,j] := 2;
end
     End;



     FOR j := 1 to 6 do
     Begin
     WW]j] := 5.1;
     UU]j] := 5.1;
     End;




     Dis : = 110 ;
     Draft := 3;
     BG := 0.5;
     FOR m := 0 to 8 do
     Begin
For i := 0 to Ml do
begin
     R1]j,m] := 2;
```

```
      R2]i,m] := 2;
end
      End
           END;



PROCEDURE HBcorrection;
VAR abc, cba : char;
 Wc      : real;

BEGIN
     clrscr;
     Writeln ('Do you want to correct any Half Breadths ?');
     repeat abc := readkey;until abc in] 'y', 'n'];
     If abc = 'y' then
     Begin
     cba := 'y''
     While cba = 'y' do
     begin
     clrscr;
     Writeln ('Waterline (m. ab Bl) ?');
     Readln (Wc);
     j := Trunc (Wc/Wl);
     Writeln (Station NO. (begin with zero in the aft msot) ?' );
     Readln (i);
     Writeln( 'Original HB is',Y] i, j]:7:2, 'm.');
```

```
Writeln ('Input corrected Half Breadth');
Readln (Y]i,j]);
Writeln ('More correction ? ] y/n]');
repeat cba := readkey;until cba in ]'y','n'];
end;
     End;
END;



Procedure Checking;
VAR Ddd     : Char;
BEGIN
     clrscr;
     For j := 0 to Nw do
     Begin
     Writeln ('kk]' :20,j,'] =',kk]j]:5,'ll]':10,j,'] =',ll]j]:5)
     End;
     Writeln ('Do you want to print the Checking result ?');
     repeat Ddd := readkey;until Ddd in ]'y','n'];
     If Ddd = 'y' then
BEGIN
clrscr;
For j := 0 to Nw do
Begin
Writeln (LST, 'kk]' :20,j,'] =',kk]j]:5,'ll]':10,j,] = ',ll]j]:5)
     End
     END
```

```
END;


Procedure Offset;
VAR ooo,oooo  : char;
LABEL 6;
BEGIN
     Writeln ('Do you want to print the Offset ?');
     repeat ooo := readkey;until ooo in] 'y','n'];
6 : If ooo = 'y' then
     BEGIN
     clrscr;
     Writeln (LST);
     Writeln (LST,'Offset Table' :45);
     Writeln (LST,'__________________________________');
'____________________________________________');
Writeln (LST);
Writeln (LST,'WL':5);
write (LST,' ' :4);
For j := 0 to Nw do
Begin
Write (LST,Wl*j:7:2);
     End;
     Writeln (LST);
     Writeln (LST,'STA.' :5);
     Writeln (LST,'________________________________',
'__________________________________________________');
```

```
     For i := 0 to Ml do
Begin
     Write (LST,X]i]:5:2);
     for j := 0 to Nw do
     begin
     Write (LST,Y]i,j]:7:3);
     end;
     Writeln (LST);
End;
     Writeln (LST,'______________________________',
'_____________________________________________________');
     END;
     Writeln ('Do you want to print the Offset ?');
     repeat oooo := readkey;until oooo in ]'y','n'];
     If oooo = 'y' then goto 6;
     END;


PROCEDURE FindInt;
LABLE 2,3;
BEGIN
     FOR j := 0 to Nw do
     Begin
     i := 0;
     If Y]i, j] <> then kk]j := i
     Else
```

```
begin
2: i := i + 1;
     if Y]i,j] <> 0 then kk]j] := i - 1
     else if i = Mm then kk]j] := i
else goto 2;
     end ;
i := Ml;
If Y'i,j] <> 0 then ll]j] := i
Else
     begin
     3: i := i - 1;
     if Y]i,j] <> 0 then ll]j] := i+1
     else if i ] Mm then ll]j] := i
else goto 3;
end
     End
END;
```



```
VAR Dddd : Char;
LABEL 1;
BEGIN
{Sample;}
1 : PRINCIPAL_DIMENSION;
POSITION_TO_INPUT_HALF_BREADTHS;
DISANCE_FROM_MIDSHIP;
INPUT_HALF_BREADTHS;
```

```
Offset;
HBcorrection;
Offset;
{INPUT_STATION_OF_ZERO_HALF_BREADTH;
FIND_INITIAL_STATION_TO_BE_CALCULATED:}
FindInt;
Calculation;
ResultOfHydrostaticData;
Checking;
STABILITY_CHECKING;
write ('continue ?');
repeat Dddd := readkey;until Dddd in ]'y','n'];
     If Dddd = 'y' then goto 1;
END.
```

About the Author

Mr. Annop Palawatvichai obtained his first degree in Mechanical Engineering from The Prince of Songkla University in 1977. He also earned B.Sc. in Naval Architecture and Shipbuilding and M.Sc. in Marine Technology from the University of Newcastle - upon - Type (U.K.) in 1979 and 1981 respectively. He spent six years in Italthai Marine Ltd. before entering the teaching profession at Chulalongkorn University.