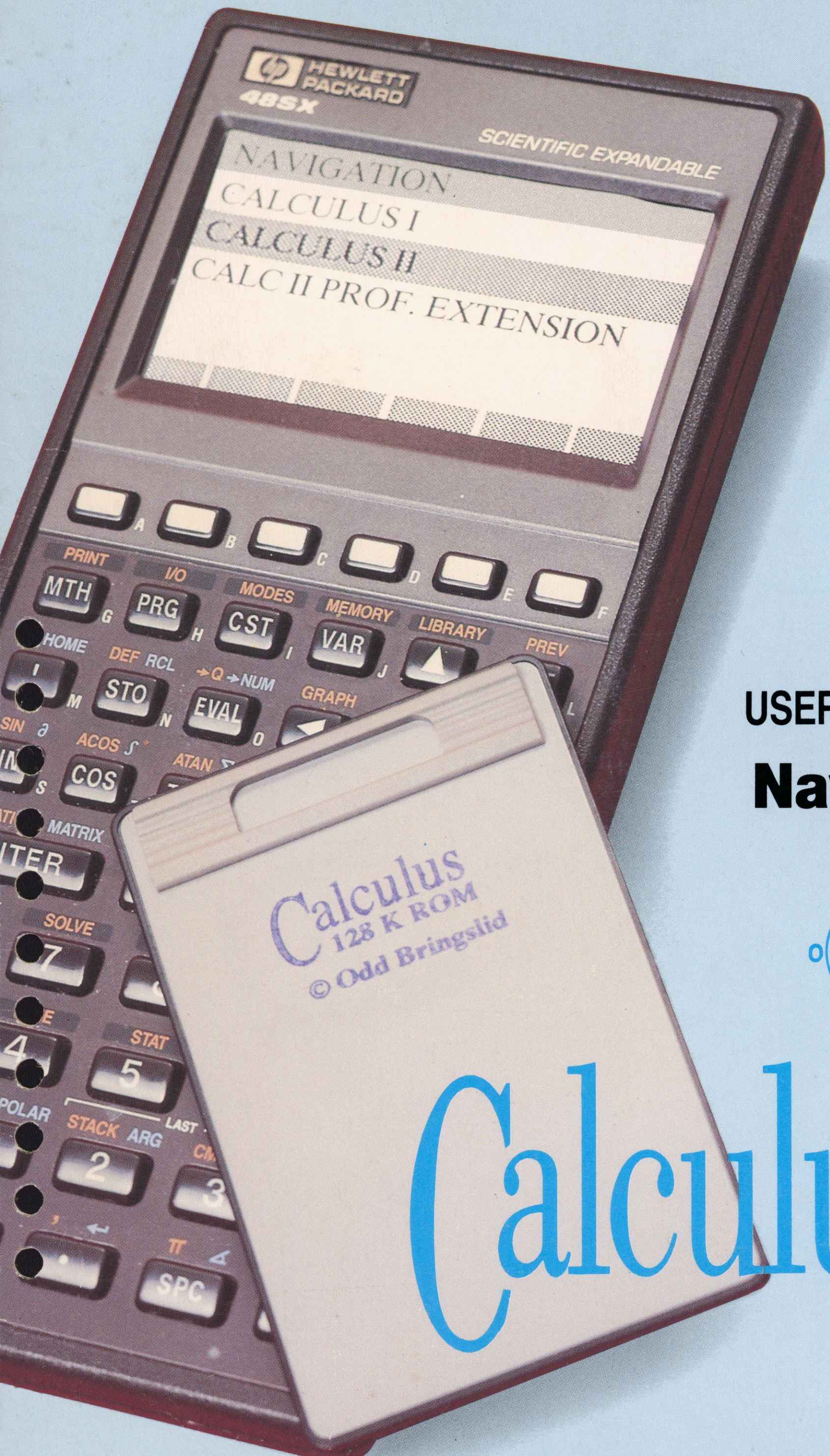


USER MANUAL

Navigare

Calculus

CALCULUS
NAVIGARE

VER 1.0

A navigation pac for calculators

Odd Bringslid
ISV
Postbox 1014   3601 Kongsberg
NORWAY

First edition august 1992

Contents

General 4

1

General

CALCULUS Navigare is a program intended for use in navigation training and by practising seamen. In the case of a few routines, other types of user such as aeronautical navigators have been taken into account. These are concerned with determining course and distance from a known end position or end position from a known course and distance for a constant course (rhumb line passage).

Required hardware

CALCULUS Navigare runs on the HP 48SX calculator. The program card can be put either in port 1 or port 2 of the calculator (open the back cover).

Start-up

Enter the calculator LIBRARY. Press NAV to display the main menu. You can then either press START to enter the menu-driven part of the program or else use a number of routines directly by putting data onto the stack.

If you, for example, enter the numbers 20 30 -10 45 and press NXT NXT SSkD then the program will calculate the great circle distance, course and vertex position for start position 20 N 30 E and end position 10 S 45 E.

These user routines are listed alphabetically and described in greater detail later in this chapter. You can see more such routines by pressing NXT on the calculator.

Signed longitude, latitude and declination values

Western longitudes and southern latitudes are negative. A latitude of -10 therefore means 10 S. Southerly declinations are also negative.

Abbreviations

The various units used have fixed abbreviations. There follows a list of these abbreviations.

• a	Earth's ellipse, major axis
• b	Earth's ellipse, minor axis
• s.lat	Start position latitude
• e.lat	End position latitude
• v.lat	Vertex latitude
• D	distance in nautical miles
• decl.	Declination (heavenly bodies)
• f	Earth's flattening (a-b)/a
• GMT	Greenwich Mean Time
• c(.s e)	True course (start pos. end pos.)
• RL	Rhumb line
• s.lon	Start position longitude
• e.lon	End position longitude
• v.lon	Vertex longitude
• LHA	Local hour angle

• LHAeast	Local eastern hour angle
• LP	Last position
• nmi	Nautical mile
• Oe	Observed, corrected elevation of heavenly body
• TB	True bearing
• PP	Probable position
• GC	Great circle
• GCD	Great circle distance
• ^{o}.m	Degrees.minutes
• ^{o}.ms	Degrees.minutes seconds
• Δe	Elevation error (signed)

Last position (LP)

Some routines store the last end position as LP. It is formatted as a complex number (latitude, longitude). This can be changed to two numbers (latitude and longitude) using C→R in the PRG/OBJ menu on the keyboard.

These routines are:

- 24 hour sailing
- DR dead reckoning
- Rhumb line reckoning
- PFLY(Subroutine)
- PMB (Subroutine)
- PMC (Subroutine)

Some routines require LP to be stored before they are run. These are:

- DR dead reckoning
- PFLY (Subroutine)
- PMB (Subroutine)
- PMC (Subroutine)

LP is stored for the first time by entering latitude and longitude, selecting R→C from the PRG/OBJ menu, entering LP and pressing STO.

User routines

A number of routines can be used independently of CALCULUS Navigare, either directly or as subroutines called from user programs. You then need to know the input data required by the routine (see manual). Direct use may be convenient for routines which are frequently used and after you have become familiar with CALCULUS Navigare (it saves having to hunt through the menus).

> *Rem. When you use routines directly, decimal degrees must be used for input. Degrees and minutes must therefore be converted first using IN→D.*

- DR — Dead Reckoning
  Input: t1 t2 v c
  t1: start time t2: stop time v: speed c: course
  LP must have been stored
- D→M — Convert decimal to °.'
  Input — Decimal number
- D→MS — Convert decimal to °.'"
  Input — Decimal number
- IN→D — Convert °.'(°.'") to decimal
  Input — Number with decimals as minutes or minutes and seconds.

- kDFLY Rocket/aircraft course/distance
  Input: s.lat s.lon e.lat e.lon
  Earth's flattening f must have been stored
  Very accurate extended latitude
  f is stored by, for example, entering 0.0036 f and pressing STO
- kDMC Mercator course/distance
  Input: s.lat s.lon e.lat e.lon
  Earth's flattening f = 1/(298.26)
- LBSTK Rhumb line reckoning
  Input: s.lat s.lon Total time
  {lat1 lon1 lat2 lon2}
  Find course/distance for various rhumb line passages from known points {lat lon...}.
- MCbm Mercator intermediate point
  Input: s.lat s.lon e.lat e.lon i.lat
  Find longitude of intermediate point i.lon
- MClm Mercator intermediate point
  Input: s.lat s.lon e.lat e.lon i.lat
  Find latitude of intermediate point i.lat
- PFLY End position, rocket/aircraft
  Input: s.lat s.lon D c
  f and LP must have been stored, very accurate extended latitude.

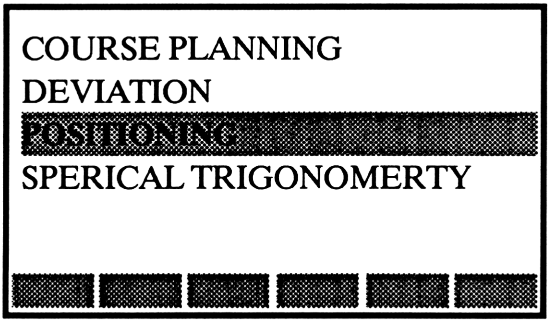
- PMB End position, middle latitude
  Input: s.lat s.lon D c
  Middle latitude sailing (Rhumb line D approx. 500) LP must have been stored
- PMC End position, Mercator
  Input: s.lat s.lon D c
  Rhumb line with fixed flattening f = 1/(298.26) LP must have been stored
- PSS End position great circle
  Input: s.lat s.lon SD c
- SSkDV Great circle SD/c/v.lat/v.lon
  Input: s.lat s.lon e.lat e.lon
- SSlm Great circle, intermediate point
  Input: s.lat s.lon e.lat e.lon i.lat
  Finds the longitude of the intermediate point.
- SSbm Great circle, intermediate point
  Input: s.lat s.lon e.lat e.lon i.lat
  Finds the latitude of the intermediate point.

User interface

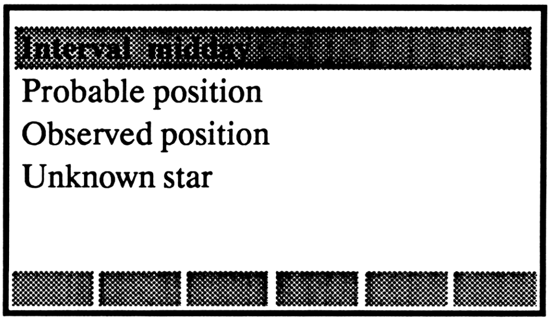
A menu system is available, which can be called by pressing START. There are, in addition, the routines which can be called directly.

The menu system is easy to use. The dark highlight can be moved up and down using the arrow keys. Press ENTER to select the highlighted field.

In the following example, POSITIONING is selected:

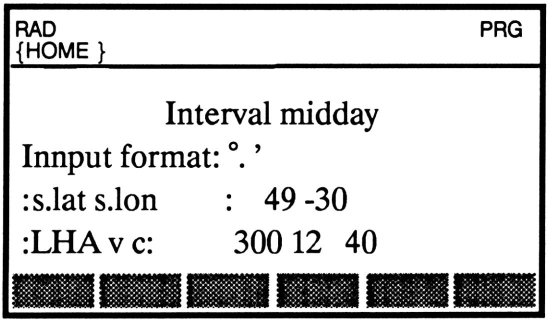


The following submenu is then displayed:



If you now press ENTER, the input form for interval to apparent midday is displayed, which calculates time and current position at midday.

Interface, interval midday



The input format specifies whether you should use degrees, minutes and decimal minutes (°.') or degrees, minutes and seconds (°.'"). This can be selected using OMR or →STK at the bottom of the window (°.'or °.'"). You can see these options in the currently displayed form, but so long as you are in the input form (EDITOR) you cannot select them.

You can now enter the numbers, typing a space between numbers entered on the same line. When you have entered all the numbers, press ENTER.

> *Note: Remember to put a space (SPC) between numbers/objects entered on the same line.*

You have now calculated the interval to the meridian from start position 49 N 30 W with a solar hour angle of 300°, a speed of 12 knots and course of 40°.

> *Note: In order to see the result you have to look on the stack. To do this, press →STK at the bottom of the window. To continue, press CONT.*

EDITOR (input form)

You can use the input form to enter data and to change data which was incorrectly entered.

If you enter the wrong number, for example, you can move the cursor to the number you want to change. Backspace (the arrow key next to CLR) can be used to delete backwards. The DEL key deletes in the current position (under the cursor).

If a number is incorrectly entered according to the syntax required by the calculator, then you will not be able to proceed until this is corrected (syntax error).

The cursor is initially positioned immediately after : s.lat s.lon : and you enter 49 here, which means start latitude 49 N. If you realize you have made a mistake, and that you should have entered 48°50' N, you can change this by using the Backspace key to delete the 9 and enter 8.50 so that 48.50 is now displayed. If you had already moved further down the form, you could have moved back using the arrow keys.

> *Note: Minutes are entered after the point, and this must not be confused with a normal decimal number. On conversion from decimal to degrees/minutes, the unit is specified.*

If you enter data which cannot be used in the calculation but which is syntactically correct, a message is displayed indicating an error in the calculation, and you can select again.

Recalling data from the stack

You may sometimes wish to import data from the stack to EDITOR. If you have made a calculation and want to use the result in subsequent calculations, you can press EDIT/→STK/ECHO/ENTER and the number will then appear in the EDITOR for use in a further calculation.

Moving around in the menus

When you are in a submenu, you can move up in the menu system using UPDIR at the bottom of the menu. You can move to the bottom of a menu by pressing shift right (blue) down arrow and move to the top of a menu by pressing shift right up arrow.

Displaying output

Output data can be displayed either as single figures or in a list. Some routines automatically output data to the matrix editor (MATRIX WRITER). In this case a fixed format is used: the first column contains course, the second distance, the third current latitude and the fourth current longitude.

The last row of output data has a special format. This row contains first accumulated distance, then the total distance along a perfect great circle followed by total distance on a rhumb line and finally deviation from the perfect great circle.

Table:

292	92	-34	18
......			
394	384	384	0.01

Note: There are a few variations from this in the cases of observed position and deviation table. For observed position, elevation error replaces perfect great circle distance and bearing replaces course in the last row. See also deviation table in Chapter 4.

The matrix editor is useful for looking at multiple results. You can move around in it using the arrow keys and go to the bottom using shift right (blue key), arrow down. If you want to put a single figure on the stack, you can use NXT →STK. You can go to the stack by selecting ATTN (ON key) and then →STK from the Navigare menu. As you can see, you get back to the Navigare menu system by pressing ATTN. For further information, see the calculator manual.

Other routines

In the menu field at the bottom of the window is the option OMR. This option gives you access to a range of conversion routines.

• °’	Degrees.minutes format
• °"	Degrees.minutes.seconds format
• D→°’	From decimal to degrees.minutes
• D→°"	From decimal to degrees.minutes.seconds
• °’→D	From degrees.minutes to decimal

- °"→D — From degrees.minutes.seconds to decimal

Note: In order to simplify the conversion of output data, some of these routines are available via →STK.

Degrees, minutes and seconds

Input data can be entered in degrees, minutes and decimal minutes or in degrees, minutes and seconds. Only decimal degrees are used in the calculations. Minutes and seconds are entered after the decimal point. You can select the mode you want to use. 34.234 can either mean 34°23.4' or 34°23'40" depending upon which mode you have selected.

Note: You must ensure that times are entered in the correct mode. If minutes and seconds are to be entered then minutes and seconds mode must have been selected. Otherwise, the time must be entered using tenths of minutes.

Ending Navigare

To end Navigare, press EXIT on the far right. Your original flags and CST menu set-up will then be restored.

2

Great circle sailing

Great circle sailing allows you to plan great circle passages by selecting various methods and conditions. In this way you can select the best route for a particular passage. There are also some independent routines for distance, course, vertex and intermediate points.

In many cases a direct great circle route will pass over land or through frozen sea, and for such cases composite routes or interruptions to the great circle route at specified latitudes are possible.

For great circle sailing, the Earth is assumed to be completely spherical.

Course, distance, vertex

This option calculates course, distance and vertex for a given start position and end position. Vertex point is output as a southern latitude in cases where the start position is entered as a southern latitude.

Interface:

```
RAD                                PRG
{HOME }
        Great circle course/dist.
Input format: °. '
:s.lat s.lon:   -35.05   20.01
:e.lat e.lon: 31.57 -80.41
```

The above example calculates distance, start course, end course and vertex point for the great circle route from 35.05 S 20.01 E (Cape Agulhas) to 31.57 N 80.41 W (Savannah) using degrees and minutes input format. The answers are gathered in a list and can be separated using PRG/OBJ→ DROP on the calculator keyboard.

24 hour sailing $\Delta c = 1^{o}$

This routine uses chord sailing (see later) with 1^{o} course changes. The alternative is to use tangential sailing with the formula for 1^{o} course changes, but this is less accurate. The deviation from an exact great circle route will be negligible as long as the passage does not pass near the equator or follow a direct northerly or southerly course.

The distance sailed is the distance for 24 hours at a given speed in knots.

Interface:

```
RAD                                   PRG
{HOME }
            24 hour sailing
Input format: °.'
:s.lat s.lon:       -35.05   20.01
:e.lat e.lon v:31.57  -80.41  16
```

The above example calculates course, distance and end position for every one degree change of course at a speed of 16 knots along the great circle from Cape Agulhas to Savannah. The 24 hour passage starts in Cape Agulhas. For further passages the last end position can be used as the start position for the next 24 hours.

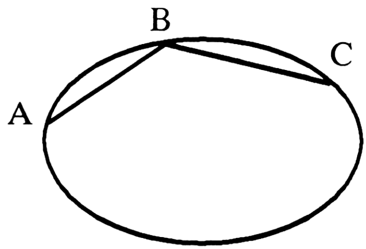
The output data is presented in the matrix editor as follows:

292	92	-34	18
384	384	384	0.01

The first row in the table (matrix) shows course, distance and end position. The second row shows total distance, the exact great circle equivalent, the rhumb line equivalent and the difference between exact great circle and actual route. End position after 24 hours is stored as LP in the format (latitude,longitude), a single complex number (see General).

Chord sailing ΔD=x_nmi

Chord sailing involves following rhumb lines from one point on a great circle to another, as opposed to tangential sailing where you are outside the great circle all the time.



The route followed is A-B-C..., that is, always to intermediate points on the great circle. The formulae are approximate but give very good results.

In this chapter, a constant distance between the points on the great circle is selected, and the course changes will therefore vary from point to point. Near the equator, the change of course at each point will be small.

Interface:

```
RAD                                PRG
{HOME }
        Chord sailing ΔD=x_nmi
Input format: °.'
:s.lat s.lon:  51.50  -55.18
:e.lat e.lon ΔD:    58.4 -14 120
```

The above example calculates chord sailing with course, distance and intermediate points from 51.5 N 55.18 W (Belle

Isle) to 58.4 N 14.0 W (Rock All) with a constant sailing distance of 120 nmi per course. The deviation from the ideal great circle is about 9 nmi, which is less than 1%.

The output data is presented in the same way as for 24 hour sailing.

Chord sailing $\Delta k = x^{o}$

Instead of constant distance per course, we are now using constant course changes (x^{o}) and variable distance per course. This routine often gives better results than constant distance, but cannot be used near the equator or for direct northerly or southerly passages (meridian passages).

Interface:

```
RAD                                   PRG
{HOME }
          Chord sailing Δk=x°
Input format: °. '
:s.lat s.lon:  51.50  -55.18
:e.lat e.lon ΔD:  58.4  -14  2
```

The above example calculates chord sailing from Belle Isle to Rock All using 2° course changes. The deviation from the ideal great circle route is less than 0.1% (approximately 1 nmi in a total of 1445 nmi).

For direct northerly or southerly passages, constant distance sailing should be used, and near the equator a rhumb line should be inserted (combined sailing).

The output data is presented in the same way as for 24 hour sailing.

Intermediate position, i.lon given

This routine calculates the latitude of an intermediate position when i.lon (longitude intermediate position) is given.

Interface:

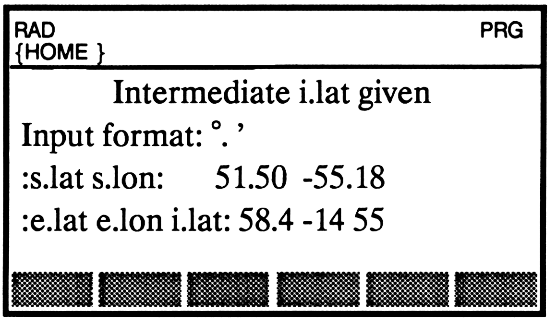
```
RAD                                   PRG
{HOME }
         Intermediate  i.lon given
Input format: °. '
:s.lat s.lon: 51.50  -55.18
:e.lat e.lon i.lon: 58.4 -14  -23
```

The above example calculates the latitude of the intermediate position with the longitude -23 (23 W) on a passage from Bell Isle to Rock All.

Intermediate position, i.lat given

This routine calculates the longitude of an intermediate point when the latitude is given. If the passage passes the vertex then there can be two answers. The answer which is nearest to the start position is chosen.

Interface:



The example calculates the longitude of the intermediate point on the passage from Bell Isle to Rockall when the latitude is 55 N.

The passage from Belle Isle to Rockall is a little unusual because it ends near to the vertex. This means that there are large changes in longitude for small changes in latitude close to Rock All.

Intermediate position, GCD given

This routine calculates the intermediate position at a known distance along the great circle. The great circle is defined in the usual way with two known positions.

Interface:

```
RAD                                PRG
{HOME }
      Intermediate GCD given
Input format: °. '
:s.lat s.lon:     51.50  -55.18
:e.lat e.lon GCD: 58.4 -14 800
```

The example calculates the intermediate position at a distance of 800 nmi from Belle Isle along the great circle to Rock All.

Distance $\Delta k = 1^{o}$

With tangential 24 hour sailing, a constant distance is sailed for every degree change of course. This routine calculates this distance. This type of sailing is usually not as accurate as chord sailing between points on the great circle. The advantage is that both the changes of course and the distances sailed are constant, apart from the last part of the day's sailing, where the distance must be adjusted to fit the total distance for 24 hours.

Interface:

```
RAD                                   PRG
{HOME }
           Distance per. °
Input format: °. '
:s.lat :     51.50
:v c:      20 57.3
```

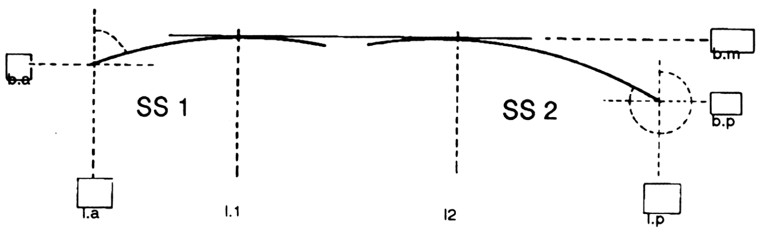
This example calculates distance per degree change of course for a passage from 51.5 N at a speed of 20 knots and a course of 57.3^{o} (Belle Isle/ Rock All).

Composite sailing

In cases where the direct great circle route passes over land or through frozen seas, composite sailing or interrupted great circle sailing can be used. It is a prerequisite, however, that the route passes the vertex.

This involves choosing a maximum latitude (between the start position latitude and the vertex latitude), and the program allows you to simulate different routes with different maximum latitude choices. The route follows a great circle as far as the maximum latitude and then follows the parallel until the longitude where it meets the end position's great circle (this longitude is calculated by Calculus Navigare).

The objective of composite sailing is to achieve a shorter distance than with pure rhumb line sailing and lower latitudes than with pure great circle sailing.



Interface:

```
RAD                                   PRG
{HOME }
          Composite sailing
Input format: °. '
:s.lat s.lon:        41.52  143.15
:e.lat e.lon m.lat:   34.24 -120.45 44
```

The example shows composite sailing from Korea to Los Angeles with 44 N as the maximum latitude. The output data shows course, distance and end position for the first great circle (GC 1), the parallel at 44 N and the second great circle (GC 2). The last row in the table shows the total distance, the direct great circle distance, the rhumb line distance and the deviation in distance from the direct great circle route.

3

Rhumb line sailing

Rhumb line sailing means following a fixed course all the time. This will always involve sailing a greater distance between two points than following a great circle, but is easier. It is usual to follow a rhumb line out of and into port, and a combination of great circle and rhumb line sailing is very common: combined sailing (not to be confused with composite sailing).

Rhumb line sailing over distances of less than 500 nmi is middle latitude sailing where no account is taken of increased latitude. Over greater distances, rhumb line sailing is either increasing sailing (assumes the world is a sphere) or Mercator sailing which takes account of the Earth's flattening.

For marine navigation a fixed value is used for the Earth's flattening, $f = 1/(298.26)$, but for high-speed navigation (aeroplanes, rockets) a user-defined value for f can be entered using 'f' STO. In this case a number of corrections which are otherwise omitted are applied.

Dead Reckoning (DR)

This routine is used for sailing distances of less than 500 nmi with a known last position (LP), course, speed and elapsed time. It is used in particular when updating the current position using sun and star observations (observed position).

LP must have been stored in the calculator. A number of CALCULUS Navigare routines store the last position automatically, but if LP is not stored then you must enter a value manually using 'LP' STO. LP is entered as a complex number (s.lat s.lon).

Interface:

```
RAD                              PRG
{HOME }
          Dead reckoning
Input format: °. '
: v  c:    20  125
:t1  t2:        8.15  12.30
```

This example calculates distance and end position for a speed of 20 knots, a course of 125°, a start time of 8^{15} and a stop time of 12^{30}. The position is stored as LP.

Rhumb Line Reckoning

This is a variant of DR, but distance is given instead of time and speed. The routine is used for a series of rhumb line passages with different courses and distances, and the end position is calculated each time. The average speed is calculated and the total elapsed time given.

Interface:

```
RAD                              PRG
{HOME }

Input format: °. '
:s.lat s.lon:     34  -16
:Time tot:         5.35
: {c  D}:  {125  20  200 35}
```

This example shows two rhumb line passages from start position 34 N 16 W. The first passage is 20 nmi on a course of 125° followed by a passage of 35 nmi on a course of 200°. The total time is 5 hours and 35 minutes. The output data is written to a table showing the current position at the end of each course and the total distance and average speed in the last row.

Combined sailing

Combined sailing means a combination of rhumb line and great circle sailing. A typical passage will follow a rhumb line leaving the start position and arriving at the destination, with a great circle route in between. Where great circles cross the equator, small changes of course will produce large distances, so it is common to follow a rhumb line when crossing the equator. Rhumb line input data (L) is distinguished from great circle (C) input data.

Interface:

```
RAD                                PRG
{HOME }

          Type L or C
Input format: °.'
:s.lat s.lon:      34  -16
:{Type lat lon}:  {L 23 12 C 12 40}
```

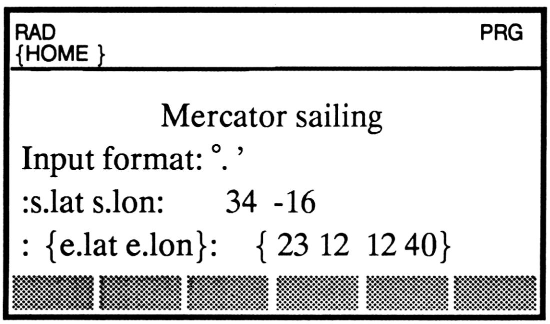
In this example a rhumb line is followed from 34 N 16 W to 23 N 12 E followed by a great circle route to 12 N 40 E. The output data is presented in a table and different passages are separated by 0 0 0 0.

Mercator sailing

Mercator sailing covers sailing which takes account of extended latitude, that is, passages over longer distances (more than 500 nmi). For educational purposes, increasing sailing is also included. Increasing sailing assumes that the Earth is spherical, whereas Mercator sailing takes into account the Earth's flattening.

Mercator sailing
This routine calculates extended latitude with a flattening of 1/(298.26).

Interface:



This example uses Mercator sailing to calculate two different courses and distances. First from 34 N 16 W to 23 N 12 E and then on to 12 N 40 E. Course and distance for each passage is given in the output data.

MC intermediate position, i.lat given

This routine calculates an intermediate position on a Mercator route where the position's latitude is known. The input data is the same as for a great circle intermediate point.

MC intermediate position, i.lon given

This routine calculates an intermediate position on a Mercator route where the position's longitude is known. The input data is the same as for a great circle intermediate point.

Increasing sailing

This routine is identical to Mercator sailing except that the Earth is seen as an unflattened sphere.

IN intermediate position, i.lat given

This routine calculates an intermediate position on an increasing sailing route where the position's latitude is known. The input data is the same as for a great circle intermediate point.

IN intermediate position, i.lon given

This routine calculates an intermediate position on an increasing sailing route where the position's longitude is known. The input data is the same as for a great circle intermediate point.

4

Deviation

Deviation means the error shown by a compass due to local magnetism (not to be confused with variation). Archibald Smith's deviation formula is used, with eight coefficients which are measured deviations for the eight points of the compass. In practice, residual deviation is calculated after various corrections have been applied.

The routine does not store coefficients, but calculates a table which shows the deviation for each 10. Deviation for other courses can be calculated using linear interpolation (for example d015 = (d010 + d020)/2.

Interface:

```
RAD                                PRG
{HOME }

          Deviation table
      Dev. 8 points of compass
        given as  d000  d045....
: d000 d045...: -9 -7.5 -4 1 6 4.5 -1 -6.5
```

The output data is a table showing the deviation for each 10^{o} course.

5

Positioning

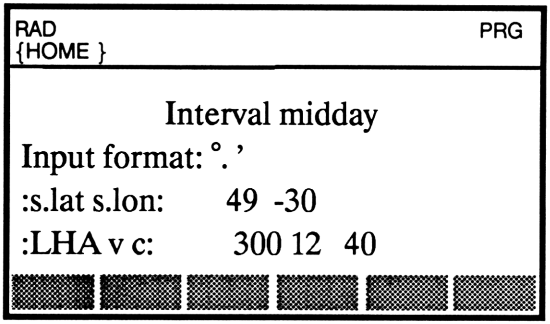
These routines determine probable position and observed position from sun and star observations. Interval to apparent midday for the sun is also included, as is the identification of unknown stars.

For determining observed position, intervening runs between two or more observations are used, even if the run only takes a few minutes ("simultaneous" observations).

Interval to local apparent midday

Interval to midday calculates current position at midday on a known course with known speed and gives the time interval. The local hour angle (LHA) is found using an almanac. The program converts this to LHA east.

Interface:



The example calculates the interval to midday in time and the current position sailing from 49 N 30 W on a course of 40 at a speed of 12 knots. The local hour angle is obtained from an almanac and is 300° in this case.

The output data shows the time interval in decimal form and the current position. The current position is stored as LP.

Most probable position

This routine calculates probable position from both sun and star observations. The estimated position is corrected by means of an elevation correction to the observation's bearing. The elevation correction is the difference between the observed, corrected elevation and the calculated elevation. It is applied towards the estimated position for positive values and away from the observed position for negative values.

The directly observed elevation is adjusted using an almanac, which takes account of the height of the bridge above sea level, refraction etc.

Interface:

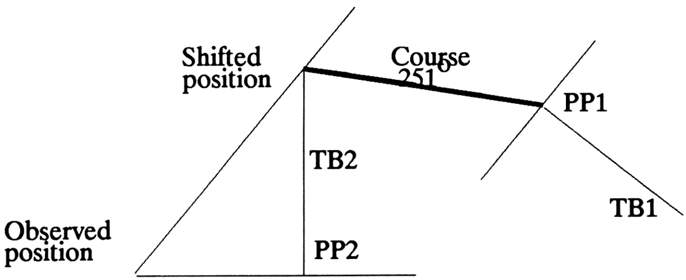
```
RAD                                  PRG
{HOME }
         Probable position
Input format: °. '
:  lat lon:        49  -30
:Oe decl LHA:    32.2 15.3  300
```

This example calculates probable position from an estimated position of 49 N 30 W, an observed, corrected elevation of 32.2°, a declination of 15.3 N and LHA of 300°. A southerly declination is negative.

The output data consists of true bearing, elevation error and probable position.

Observed position

Observed position is the intersection of the position lines through the probable positions for two observations. The position line for the first observation is shifted parallel along the course to the position of the second observation (sun-run-sun).



In the illustration PP stands for probable position, TB for true bearing.

Interface:

```
RAD                                   PRG
{HOME }
          Observed position
Input format: °. '
: lat lon:        50.42 -26.19
:LHA1 LHA2:          307  360
```

```
RAD                                   PRG
{HOME }
: Oe1 Oe2:   31.12 50.25
:dec1 dec2: 10.48  10.45
:      v  c:        18  251
:GMT1 GMT2:    10.15 13.54
```

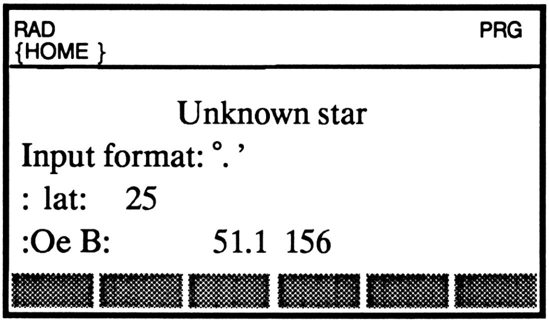
This example calculates observed position for two observations made at 10.15 GMT and 13.54 GMT. The estimated position is 50.42 N 26.19 W, and declination, corrected elevations and hour angles are given.

Output data is a table where the first row is bearing, elevation error and PP for first obervation, next course and distance to shifted position and finaly bearing and elevation error for last PP and the observed position.

Unknown star

Identification of an unknown star is performed using calculated declination which is compared with the declination in the almanac.

Interface:



This example calculates the declination of a star with a corrected elevation of 51.1° and a gyro bearing of 156°. The latitude is 25 N. The output is negative, which indicates a southerly declination.

6

Spherical trigonometry

The spherical trigonometry routines calculate the sides and angles of spherical triangles. They require the user to simply enter the known sides/angles and specify the ones to be found. The program will then automatically select the correct calculation.

A standard notation for sides and angles must be followed: A, B and C for the angles and a, b and c for their opposing sides.

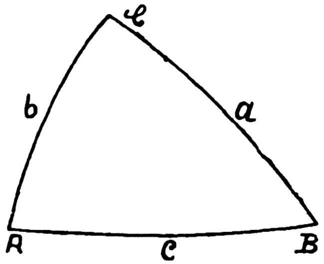


Fig.6.1

If the triangle is impossible, the message "Impossible" is output.

> *Note: In some cases two solutions are possible, byt only one is given.*

Two angles given

For this routine you must specify which unknown side or angle is to be found and which side and which two angles are known. You must use the symbols A, B and C for angles and a, b and c for sides.

Interface:

```
RAD                                PRG
{HOME }
Input format: °. '
:Unknown:        A
:     Side:    'a=80.12'
: Angles:    'B=28'  'C=157'
```

This example calculates angle A for a spherical triangle with known side a = 80.12 ($80^{o}12'$) and known angles B = 28^{o} and C = 157^{o}.

Two sides given

For this routine you must specify which unknown side or angle is to be found and which angle and which two sides are known. You must use the symbols A, B and C for angles and a, b and c for sides.

Interface:

```
RAD                         PRG
{HOME }

Input format: °.'
: Unknown:      B
:   Angle:    'A=32.41'
: Sides:'a=80.12' 'c=135.45'
```

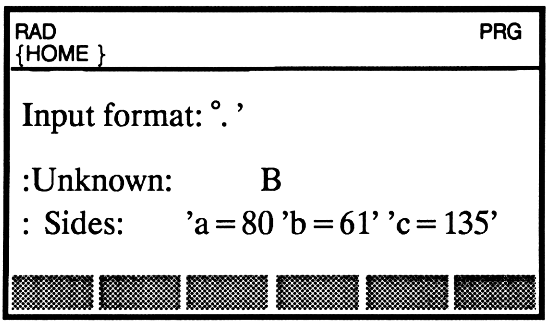
This example calculates angle B when angle A is $32^{o}41'$, side a = $80^{o}12'$ and side c = $135^{o}45'$.

> *Note:In this case there can be two solutions. This happens when a side's opposing angle is known and this side is the smallest known side. This is not implemented in Navigare.*

Three sides given

This routine calculates the specified angle when 3 sides are known.

Interface:



This example calculates angle B when side $a=80^{o}$, side $b=61^{o}$ and side $c=135^{o}$.

Three angles given

Interface:

```
RAD                                  PRG
{HOME }

Input format: °. ’

:Unknown:        c
: Angles:  ’A=32 ’B=28’ ’C=157’
```

This example calculates side c when angle $A=32^{o}$, angle $B=28^{o}$ and angle $C=157^{o}$.

Mathematical rules in spherical geometry

The spherical geometry programs use several formulae. The symbols shown in Fig. 6.1 are used.

Sine rule:

$$\mathrm{Sin}(a)/\mathrm{Sin}(A)=\mathrm{Sin}(b)/\mathrm{Sin}(B)$$

Cosine rule:

$$\mathrm{Cos}(a) = \mathrm{Cos}(b) * \mathrm{Cos}(c) + \mathrm{Sin}(b) * \mathrm{Sin}(c) * \mathrm{Cos}(A)$$
$$\mathrm{Cos}(A) = -\mathrm{Cos}(B) * \mathrm{Cos}(C) + \mathrm{Sin}(B) * \mathrm{Sin}(C) * \mathrm{Cos}(a)$$

Cotangent rule:

$$\mathrm{Cot}(a) * \mathrm{Sin}(b) - \mathrm{Sin}(C) * \mathrm{Cot}(A) = \mathrm{Cos}(C) * \mathrm{Cos}(b)$$

Nepers formula:

$$\tan(A + B) = \mathrm{Cot}(C/2) * (\mathrm{Cos}((a-b)/2)/\mathrm{Cos}((a + b)/2))$$

Changes in version 3.0

Version 3.0 has got the possibility to use a GPS receiver and a change is made in the output of positions. The format is the same as the input data i.e usualy in degrees, minutes and parts of minutes.

7

Extension of Navigare

The extension includes almanac, altitude corrections and identification of unknown star (a star is proposed). The almanac contents the sun and 38 stars.

Almanac

This may be found in the user menu and the accuracy is about 0.5’.

Sun

The LHA and declination for given date and time are calculated.

REGULUS,RIGEL, RIGIL KENT, SCHEDAR, SIRIUS, SPICA, SUHAIL, VEGA.

Extension of POSITIONING (Navigare)

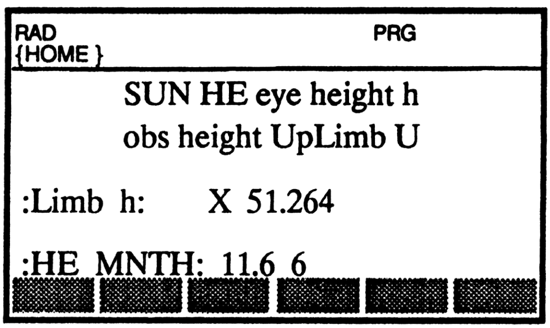
Unknown star

The input is as before but a proposal of which star is added. This is a proposal because the calculated declination will not match exactly with tabulated values.

Altitude correction SUN

Sextant altitude corrected for index error is put in together with the height of the eye i metres.

Interface:



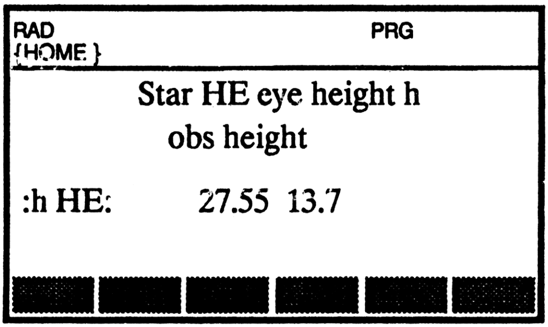
Here observed corrected altitude is calculated for sextant altitude adjusted for index $51^{o}26.4'$, height of eye 11.6 metres and month june (6).

Rem. Upper limb sun is U, lower limb is X or whatever

Altitude correction star

Sextant altitude corrected for index is put in.

Interface:



Observed corrected altitude is calculated for sextant altitude $27^{o}55'$ and height of eye 13.7 metres.

GPS

Navigare may be connected to GPS receivers and position, speed, course, distance and UTC (GMT) may be calculated.

GPS-NMEA-0183-GGA protocol

The most frequently used protocol for GPS receivers is the NMEA-0183. Standard parameters are:

Baud rate:	4800 bits/sek
Data bits:	8 (d7=0)
Parity:	0
Stop bits:	En eller flere

Navigare may be connected to all GPS receivers using this protocol and a RS232 serial interface.

The GPS menu may be found in the lower part of the screen and you will get into a submenu with the folowing possibilities:

SAIL

This routine is used in connection with practical sailing from one Way-point to another. Way-points are given in the format (lat, long) and you have to use degrees. minutes/1/100 minutes. Navigare will then calculate speed, true course, correct course, distance and total distance.

The speed is given in knots. Total distance (Tot.dist) is the distance left to the final Way-point. Distance is distance to the next Way-point, true course is the course you are currently sailing and correct course is the course that leads directly to the next Way-point.

During the sailing you have to adjust the course so that true course and correct course will be the same. The Way-points are put in a list {WP1 WP2...}. A distance in metres indicating the arrival to a Way-point has to be given in (TOL). If this is set to 100 metres, it means that you have arrived when you have come inside a circel around the point with radius 100 metres.

GPS receivers are not very accurate in non-differential mode and an accuracy of 100 metres (0.06 minutes) is normal. The first line in the output form indicates the next Way-point.

Interface (directly on the stack):

```
RAD                          PRG
{HOME }

2:  100
1: {(59.3456, 10.4523) }
```

In the example TOL is the distance around the Way-point indicating the arrival (100 m) and the sailing goes to the Way-point 59.3456 N 10.4528 E. If several points are used they are put into a list in the same way {(59.2345,10.2567) (59.5648,10.3789)}.

POS

Here your current position and the time (UTC) are found.

SATL

GPS receivers may in a given moment "find" a specified number of satelites. This routine will show the number of satelites being found at a given moment. This number should not be less than 4. Before using the GPS to navigate you should be sure of that the number is 4 or greater.

Index

CALCULUS

LOAD/UNLOAD

VER 1.0

A load/unload program for calculator

Odd Bringslid
ISV
Postbox 1014 3601 Kongsberg
NORWAY

First edition october 1992

1

In general

The load/unload program is written as an addition to CALCULUS Navigare and some of the general qualities of the program are described in Navigare.

Abbreviations

- KG Distance from centre of mass to the keel.
- KY Moment arm keel/line of buoyancy
- Θ Heeling angle
- GZ Arm of heeling
- DP Tonnage
- KGL Distance from center of mass to midships (L/2)

- I_ρ Effect of free surface in tanks
- VM Vertical moment
- TM Moment of trim
- Δ?? Change in ??, positiv for increase
- DWT Deadweight
- ρ Density
- BM Distance from centre of buoyancy to metacentre
- GM Distance from centre of mass to metacentre
- W Width (of ship)
- w weight

User menu

At the bottom of the screen there are two user menus. Theese are BRGN and OMR. BRGN gives access to some often used routines and OMR is conversion between units (MASS,LENG (length) and VOL (volume)).

BRGN menu

Accessible routines are listed:

- C→DV Calculation of volume-displacement with known block-coefficient, (Cb), Length (L) Width (W), draught (d)
- DV→C Calculation of block-coefficient with known volume displacement (DV), L, W and d.
- DV→D Calculation of mass displacement (tonnage) (DP) with known DV.
- DP→D Calculation of DV with known DP
- CW→A Calculation of Cw coefficient with known area of water line, L and W.
- AREA Calculation of area with known ordinates and interval of abscissa (Simpsons rule)
- DWT Calculation of dead weight for known DP.
- KGKY Calculation of KG for given KY and Θ (heeling angle) for wanted DZ.
- BM Calculation of BM

- GMK Calculation of GM, inclining experiment
- GMR Calculation of GM with known period of rolling.
- KG Look at current KG value

OMR menu

MASS:

- →LB Kg to Pound (lb)
- →TON Kg to UK ton
- →SHT Kg to US short ton
- LB→ Pound to Kg
- TON→ UK ton to Kg
- SHTN US short ton to Kg

LENG(TH):

- M→IN Metre to inch (in)
- M→FT Metre to feet (ft)
- M→YD Metre to yard (yd)
- M→N Metre to nautic mile
- IN→M Inch to metre
- FT→M Feet to metre

- YD→M Yard to metre
- NM→ Nautic mile to metre

VOL(ume):

- →IN3 M^3 to in^3
- IN3→ In^3 to m^3
- →FT3 M^3 to ft^3
- FT3→ ft^3 to m^3
- →YD3 M^3 to yd^3
- YD3→ yd^3 to m^3
- →UKG M^3 to UK gallon
- UKG→ UK gallon to m^3
- →USG M^3 to US gallon
- USG→ US gallon to m^3

2

Table of moments

In connection with loading and unloading of ship, some conditions will be unchanged. This includes volume and other characteristics of tanks and rooms (holds), and some of the load may be unchanged from time to time. It is therefore important that only changes are put into the program from time to time and not all of the information every time.

The routine for table of moments calculates vertical moments and trim moments for a given loading situation. The use of the table is divided into two parts: Define the table and the use of the table for calculating changes.

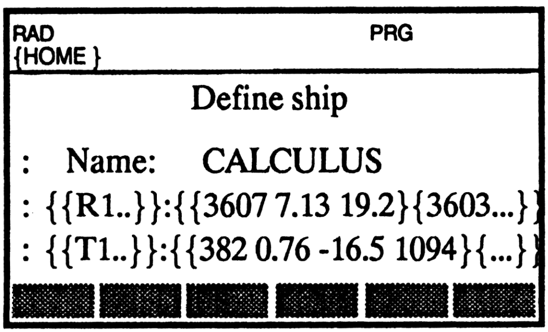
Define ship

Data that remain unchanged for a given ship are stored once for all. This includes volumes, center of mass, vertical (KG) and horizontal (KGL) for rooms and tanks, moment of inertia (free surface tanks) and informations for empty ship.

Initial values for displacement, deck cargo, load in tanks and rooms (holds) etc. have to be put into the table.

The table for this specific ship will then be constructed and stored under the name of the ship. The data to be unchanged are store under the name 'NameD'. Another ship will have another table (different number of tanks etc.) stored under another name.

Interface:



Next picture is the matrix writer where the table will be constructed. The data given so far is the name of the ship, data for rooms (R) and tanks (T) in double lists {{}}. For rooms the volume, KG and KGL are put into the list. For new rooms the corresponding data are put into a new list {}. For tanks the same data are given in addition to moment of inertia for calculating the effect of free surface. For two rooms and two tanks the double lists may look like this:

Rooms: {{3607 7.13 19.2}{3603 7.13 -15.9}}
Tanks: {{382 0.76 -16.5 1094}{349 0.76 -33.9 785}}
Further:{{DP KG KGL}} (empty ship)

DP	0	0	0	0	0	0	0	0
0	Δm	m	ρ	KG	KGL	I_ρ	ΔVM	ΔTM
DK	0	0	0	0	0	0	0	0
R1	0	0	0	0	19.2	0	0	0
R2	0	0	0	0	-15.9	0	0	0
T1	0	0	0	0	-16.5	0	0	0
T2	0	0	0	0	-33.9	0	0	0
Othr	0	0	0	0	0	0	0	0
Sum	0	0	0	0	0	0	0	0
ΔKG	0	ΔKGL		0	0	0	0	0

Rem. In the program the symbols are not used (numeric matrix writer) and the second row is filled with zeroes. DK stands for deck cargo, Othr stands for other kinds of load (stores etc)

Values for KGL for tanks and rooms are put in by the program (based on stored data).

The first row deals with displacement (tonnage) DP. Here data for empty ship may be put in or another known "loadsituation". KG is the "total" KG and KGL "total" KGL (ΣMoments/DP)

Next row is the symbols for the columns (zeroes in the program). Δm is change in mass (weight), m for mass (weight) after the loading/unloading, ρ for density of the cargo/load, I_ρ the effect of free surface in tanks, ΔVM change in vertical moment and ΔTM change in trim moment.

The third row is deck cargo and the weight, KG and KGL must be put in. R1 R2 is room1 and room2, further T1 T2 is tank1 and tank2.

Othr may be stores, the bag net for fishing boats or whatever.

The table may then look like this before its stored:

DP	0	9000	x	5.8	-1.6	x	x	x
x	Δm	m	ρ	KG	KGL	I_ρ	ΔVM	ΔTM
DK	0	0	0	0	0	0	0	0
R1	0	400	0.6	0	19.2	0	0	0
R2	0	300	0.8	0	-15.9	0	0	0
T1	0	0	0	0	-16.5	0	0	0
T2	0	145	0.95	0	-33.9	0	0	0
Othr	0	0	0	0	0	0	0	0
Sum	0	0	0	0	0	0	0	0
ΔKG	0	ΔKGL		0	0	0	0	0

Before the final storage the KG for tanks and rooms are calculated and the eventual effect of free surface in tanks.

After the table is stored it looks like this:

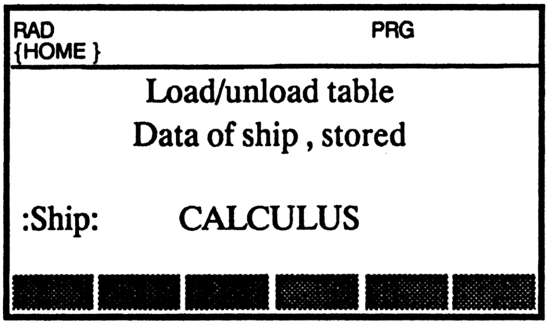
1	0	9000	0	5.8	-1.6	0	52...	-14..
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
1	0	400	0.6	1.32	19.2	0	0	0
2	0	300	0.8	0.74	-15.9	0	0	0
1	0	0	0	0	-16.5	x	0	0
2	0	145	0.95	0.34	-33.9	745..	0	0
1	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
1	0	2	0	0	0	0	0	

Compared to what was given in we see that VM and TM for the ship are calculated (first row) and KG for rooms and tanks and I_p for tanks are also calculated before stored.

Table of moments

Changes in a stored table are put in here and the table is put into the matrix writer by writing the name of the ship.

Interface:



1	0	9000	0	5.8	-1.6	0	52...	-14..
0	0	0	0	0	0	0	0	0
1	1000	0	0	14	1.1	0	0	0
1	0	400	0.6	1.32	19.2	0	0	0
2	-300	300	0.8	0.74	-15.9	0	0	0
1	0	0	0	0	-16.5	0	0	0
2	0	145	0.95	0.34	-33.9	745..	0	0
1	0	0	x	0	0	0	0	0
10	0	0	0	0	0	0	0	0
1	0	2	0	0	0	0	0	0

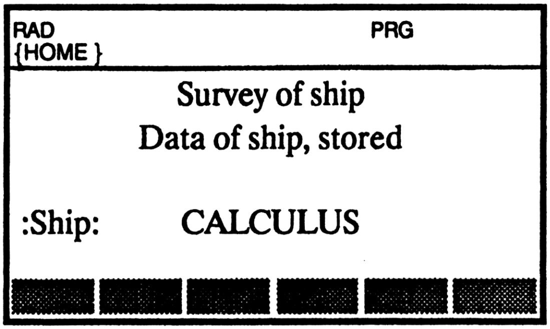
The changes put in are 1000 ton deck cargo, KG = 14 metres and KGL = 1.1 metres and unloading of room 2. This gives new displacement 9700 ton, new KG = 6.88 metres and new KGL = -0.88 metres (astern trim). Sums of changes in weight, vertical moment, trim moment and I_ρ are calculated and given in row 9. Below changes in KG and KGL are given.

Push ENTER to reset columns 2, 8 and 9 and to store the new table.

Look ship

Here some data stored for a specific ship may be looked at.

Interface:



3

GZ curve

GZ curves give the "heeling arm" GZ as function of the heeling angle Θ. The curves are based on crosscurves (KY curves) or a given GZ curve for a given KG. Standard values for Θ are $\Theta = 0, 7.5^o, 15^o, 30^o, 45^o, 60^o$ and 75^o.

By using an interpolating routine (cubic spline) values for every 2.5^o, $0\text{-}60^o$ may be calculated. Θ for max GZ are also calculated and put in the last row of the GZ table (Statistic matrix ΣDAT).

GZ-KY table

The equation for GZ and KY is:

$$GZ = KY - KG * \sin\Theta$$

KY at angles [0 7.5 15 30 45 60 75] are put into the program.

Interface:

```
RAD                         PRG
{HOME }
        GZ-KY table Θ =
      [0 7.5 15 30 45 60 75]
: KG: 7.6
:[KY]: [0 1.1 2.2 4.4 6.3 7.4 7.75]
```

Here the GZ table (GZ curve) for KG = 7.6 metres and KY values as given is calculated. The table is calculated in the interval [0 75], for every 2.5^{o} between 0 and 60^{o}. The last row gives the max GZ with corresponding angle. The value of KG is stored for future reference (KG option in user menu).

Define GZ table

Known values for GZ and the corresponding KG are put directly into the input form and the whole table is calculated using interpolation. GZ is put in like a vector [0 0.11...]. Stored as ΣDAT and the KG value as KG for future reference.

GZ Plot

Here the GZ table is plotted using SCATTER PLOT in the HP 48SX. The table is stored in ΣDAT and no further input is necessary.

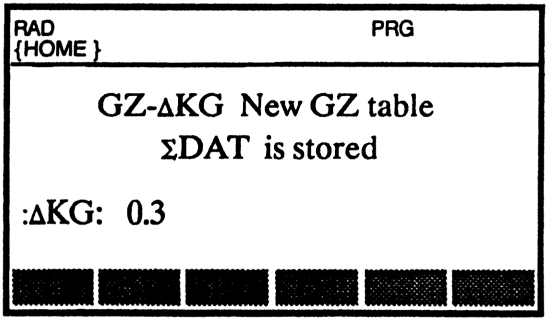
The coordinates of a point on the curve may be found by moving the cursor with the arrow keys and then push COORD.

To se the axes you may move the cursor slightly downwards and push CENT. Push the minus key to see. ATTN (ON) will get you back to the main menu.

GZ-ΔKG table

By changes in KG a new curve is calculated. The new value of KG is calculated and stored.

Interface:



Look GZ table

The GZ table is put into the matrix writer.

4

Stability

In this chapter some calculations about stability are done. This includes dynamic stability (control against norwegian claims), moving weights and heavy lifts (mast boom).

Dynamic stability

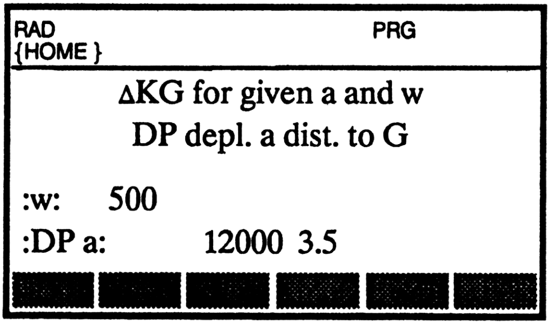
The calculations are entirely based on the GZ curve (ΣDAT table). The result is put on the stack and can be seen by pushing →STK. OK means that the claims are fulfilled and actual values are: {GZ at 30°, Θ for max GZ, area 0-30°, area 30-40°, area 0-40°}.

Moving weights

Move ΔKG?

Here the change in KG is calculated for a given weight.

Interface:



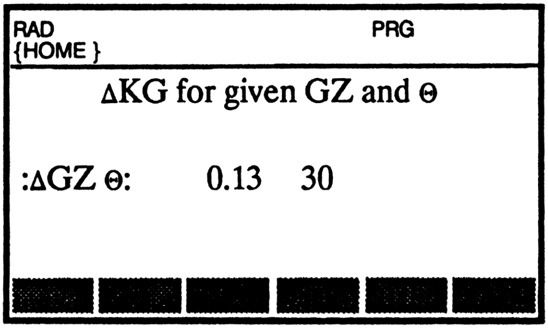
The routine calculates ΔKG by moving a weight of 500 tons a distance of 3.5 metres.

Rem. Negativ a is used by moving downwards

Move ΔKGΘ?

The change in KG for a certain change in GZ for given angle is calculated.

Interface:

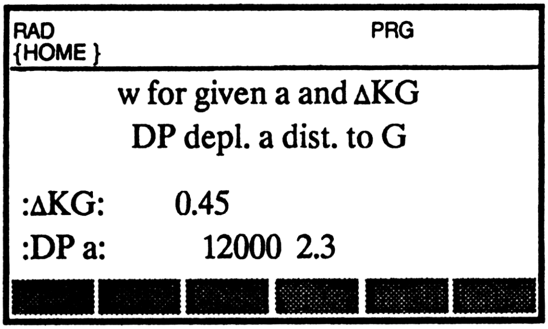


Here the change in KG if the change in GZ for 30^{o} is 0.13 metres is calculated. The problem might be to calculate the weight to be moved when the GZ at given angle is given.

> *Rem. If the GZ is increasing the centre of mass has to be lowered and ΔKG is negativ.*

Move w?
The weight to be moved for a given change in KG is calculated.

Interface:



Here the weight to be moved 2.3 metres to heighten the KG with 0.45 metres is calculated. Displacement 15000 tons.

Load ΔKG?
Loading a weight means to take on board a weight and not only move it. The change in KG is calculated.

Interface:

```
RAD                              PRG
{HOME }
        ΔKG for given a and w
        DP depl. a dist. to G
:w:    500
:DP a:        12000  3.5
```

The example calculates the change in KG by loading a weight of 500 tons and placing it 3.5 metres above the centre of mass.

Load w?

This routine calculates the weight to be loaded to change the KG by given amount.

Interface:

```
RAD                              PRG
{HOME }
        w for given a and ΔKG
        DP depl. a dist. to G
:ΔKG:      0.35
:DP a:        12000  3.5
```

The example calculates the weight that heightens the KG by 0.35 metres, placed 3.5 metres above G.

Unload ΔKG?

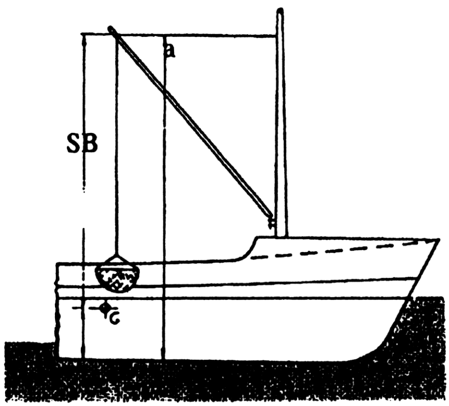
Same procedure as loading, but here the weight is unloaded. Change in KG is calculated by unloading a given weight.

Unload w?

Same procedure as loading. The weight to be unloaded is calculated.

Heavy lift

Heavy lift is heaving with a boom. The resulting GZ is calculated together with the corresponding change in KG. There is a possibility to use ballast tanks.



Interface:

```
RAD                                  PRG
{HOME }
            Heavy lift , weight
            See figure, manual
: SB a w: 12.2  15  200
:DP mT lT:    15000 331  6
```

The example calculates the change in KG and GZ arm when the distance to the top of the boom from the centre of mass is 12.2 metres. The distance from the mast to the top of the boom is 15 metres and the weight is 200 tons. Displacment 15000 tons and ballast 331 tons with centre of mass 6 metres form the centre line. New GZ curve may be calculated (with new KG) and the heeling angle may be found for the calculated GZ arm.

The following subjects are covered by CALCULUS:

- Great Circle sailing
- Spherical Trigonometry
- Rhumb Line sailing
- Composite sailing
- Combined sailing
- 24 hours sailing
- Marcator sailing with observations
- Sun observations
- Star observations

Odd Bringslid International Software Vendor
Box 1014 KNP
3601 Kongsberg
NORWAY
Tel. + 47-3-73 88 40. Fax. + 47-3-73 86 33

Distributor:
BB Marketing ANS. Ensjøveien 12 B, 0655 Oslo, Norway
Tel. + 47-2-67 11 57. Fax. + 47-2-67 11 37