

---

# HARMONIZED CALCULATION METHOD FOR MOBILE SERVICES (HCM-MS)

---

Developed by the Sub Working Group Program  
Version 7  
7<sup>th</sup> April 2016  
(Prague)



## FOREWORD

---

As chairmen of the Sub Working Groups Program (suspended after Plenary 2014, Budapest) and Mobile Service, sub-working groups of the HCM Agreement's Technical Working Group Harmonized Calculation Method, we have the honor to present you the Version 7 of the HCM-MS program and documentation, which is in line with the HCM Agreement 2014 (Budapest).

The last Version 7 of the HCM-MS program implements calculation methods applicable to:

- HCM Agreement 2014
- ERMES (\*)
- GSM 900 (\*)
- GSM 1800 (\*)
- Emergency- and Security Services (band 380-385/390-395 MHz)
- UMTS and IMT 2000 systems (\*)
- Coverage
- Wideband systems

*(\*) Please note that not all the technical parameters of these systems are fully integrated in the HCM Agreement.*

The program and its subroutines are written in FORTRAN-90.

In the documentation some pages are intentionally left blank to permit recto-verso printing. In the documentation you will find the flow charts on the left page, the text on the right page.

As chairmen of the -SWG-Mobile Service I would like to thank all active participants of the group for their support during the development of the new version, the program and its documentation.

Kind regards,

Philipp Birrer  
Chairman of the Sub Working Group Mobile Service  
(TWG-HCM SWG-MS)

---

## TABLE OF CONTENTS

---

<b>FOREWORD</b>	<b>3</b>
<b>GENERAL</b>	<b>6</b>
<b>Chapter 1: HCMMS_V7 subroutine</b>	<b>8</b>
Part 1	8
Part 2	11
Chapter 1.1: Read and test input values	13
<b>Chapter 2: Subroutine P_to_P_calculation</b>	<b>15</b>
Part 1	15
Part 2	18
Part 3	20
Part 4	22
Part 5	24
Part 6	26
Part 7	28
Chapter 2.1: Subroutine Position_of_mobile	30
Part 1	30
Part 2	32
Part 3	34
Part 4	36
Chapter 2.1.1: Subroutine TestCut	38
Chapter 2.1.2: Subroutine NearestLinePoint	40
Chapter 2.1.3: Subroutine Calc_Tx_pos	42
Chapter 2.1.4: Subroutine Calc_Rx_pos	44
<b>Chapter 3: Subroutine Line_calculation</b>	<b>46</b>
Part 1	46
Part 2	49
Part 3	51
Chapter 3.1: Subroutine CBR_Coordinates	53
Chapter 3.2: Subroutine Test_cut1	55
Chapter 3.3: Subroutine Manage_List	57
<b>Chapter 4: Subroutine Permissible_FS_calculation</b>	<b>59</b>
Part1	59
Part 2	61
<b>Chapter 5: Common subroutines</b>	<b>63</b>
Chapter 5.1: Subroutine Calc_distance	63

Chapter 5.2:	Subroutine Calc_direction.....	64
Chapter 5.3:	Subroutine Point_height.....	65
Chapter 5.4:	Subroutine Point_type.....	67
Chapter 5.5:	Subroutine Profile.....	69
	Part 1.....	69
	Part 2.....	71
Chapter 5.6:	Subroutine Antenna.....	73
Chapter 5.7:	Subroutine Antenna_Correction.....	75
Chapter 5.8:	Subroutine CooConv.....	77
Chapter 5.9:	Subroutine TCA_correction_calculation.....	79
Chapter 5.10:	Subroutine Dh_calculation.....	81
Chapter 5.11:	Subroutine Dh_Correction.....	83
Chapter 5.12:	Subroutine Get_FS_from_figures.....	85
	Chapter 5.12.1: Calculate field strengths according Heff for 10 m < Heff < 3000 m.....	87
	Chapter 5.12.2: Calculate L_E10d, S_E10d, L_E10dh10, S_E10dh10, L_Edh1, S_E10dh1 and Land_FS_1kW, Sea_FS_1kW.....	89
	Chapter 5.12.3: Calculate dhx, L_E10dhx, S_E10dhx and Land_FS_1kW, Sea_FS_1kW.....	91
Chapter 5.13:	Subroutine Get_figure_FS_value.....	93
Chapter 5.14:	Subroutine New_coordinates.....	95
<b>Chapter 6:</b>	<b>Storage format of the height data.....</b>	<b>97</b>
<b>Chapter 7:</b>	<b>Storage format of the morphological data.....</b>	<b>100</b>
<b>Chapter 8:</b>	<b>Geographical data requirements for line calculations, field strength predictions and storage of line-data.....</b>	<b>103</b>
	Chapter 8.1: Data requirements.....	103
	Chapter 8.1.1: The different cases.....	103
	Chapter 8.1.2: Border lines to involved countries.....	103
	Chapter 8.1.3: Cross border lines (cross border ranges).....	104
	Chapter 8.1.4: x-km lines of preferential frequencies.....	105
	Chapter 8.1.5: Closed borderlines of the own and of involved countries (for calculation of the position of mobiles).....	107
Chapter 8.2:	Storage format of the line data.....	108
Chapter 8.3:	How to create the required database with the "BORDER"-program.....	109
Chapter 8.4:	FORTTRAN program to convert ASCII – line-data to HCM-format.....	110
<b>Chapter 9:</b>	<b>Interface to the HCMMS_V7 subroutine in FORTRAN 90.....</b>	<b>111</b>
<b>Chapter 10:</b>	<b>Interface to the HCMMS_V7.DLL.....</b>	<b>116</b>
<b>Annex:</b>	<b>List of error codes and Info(i) values.....</b>	<b>121</b>

---

## GENERAL

---

This Harmonized Calculation Method for Mobile Service (HCM-MS) is part of the HCM Agreement '14. With the new non-strict C\_modes model the field strength prediction can also be used for services not noted in the HCM Agreement '14 by setting input parameters accordingly.

### General Note

In the description (unless stated different):

- all angles are in degrees or radians,
- all heights are in meters,
- all distances are in kilometers,

### List of common subroutines

The HCM MS V7 program uses a lot of additional subroutines. The list of the common subroutines is:

- calculation of the distance between two points (Tx and receiving point) (Calc\_distance)
- calculation of the azimuth from one point to another point (Calc\_direction)
- read the height of a given point from the terrain database (Point\_height)
- read the morphological information of a given point from the morphological database (Point\_type)
- get the profile between two points (heights or morphological information) (Tx and receiving point) (Profile)
- calculate the gain/loss of a 3D-Antennamodel, combined from horizontal and vertical pattern (Antenna\_Correction)
- convert co-ordinates to text format (CooConv)
- calculation of the clearance angle correction factor (TCA\_correction\_calculation)
- calculate  $\Delta h$  (Dh\_calculation)
- calculation of the correction factor  $k\Delta h$  (Dh\_Correction)
- calculation of the (1kW) land- and sea field strength from curves (Get\_FS\_from\_figures)
- get the field strength values from the figures (land and sea) (Get\_figure\_FS\_value)
- calculation of the new co-ordinates from a given point in a given direction with a given distance (New\_coordinates)
- calculation of the position of the mobile station (Position\_of\_mobile)

### Interface

The HCM MS V7 program is only a subroutine. The interface to the HCM MS V7 is described in chapter 9.

## Remarks

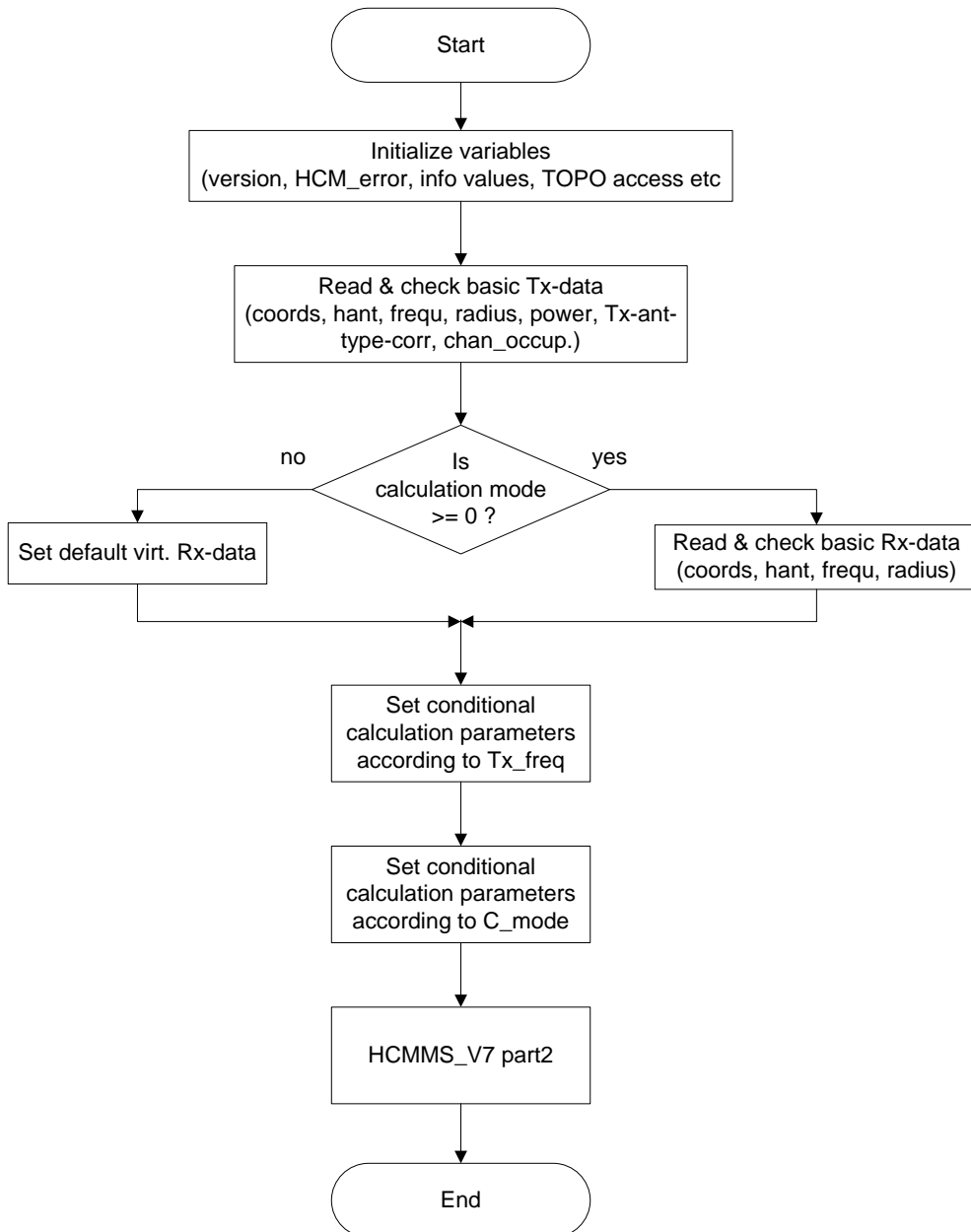
When an error occurs, an error code is generated and the subroutine is terminated. A list of error codes and their description is given in chapter 11.

Certain circumstances of calculation are indicated by info values, a list is annexed.

# CHAPTER 1: HCMMS\_V7 SUBROUTINE

## PART 1

### Chapter 1: HCMMS\_V7 subroutine (part1)



@'15



## Description

This subroutine HCMMS\_V7 is the Harmonized Calculation Method itself. The HCMMS\_V7 subroutine performs calculation from a transmitting station to a receiving station or from a transmitting station to a co-ordination line. Because the HCMMS\_V7 is a subroutine, a surrounding program is required to run this software. An example for a small surrounding program is given in chapter 10. A more complex surrounding program is supplied by the HCM group and can be found on the web site of the HCM Agreement.

### Read and check basic Tx-data

This process is part of process described in chapter 1.1.

### Is calculation mode $\geq 0$ ?

#### Set default virt Rx-data

For point-to-point calculations detailed Rx-data is needed, for line calculations fictive Rx-data.

### Read and check basic Rx-data

If real Rx station, read avail. input data. This process is part of process described in chapter 1.1.

### Set conditional calc. parameters according to Tx-frequency

#### Set conditional calc. parameters according to C\_mode

For "strict HCM" calculations (C\_mode 0/-1) default values depending on frequency according to Table of Annex 1 are selected, for others the values are set according to C\_mode.

## Table of calculation modes

C_mode	Type of calculation	HCMmode	Applied T%	Applied $h_2$
+12	P2P	non strict	1	noc
+11	P2P	non strict	50	noc
+10	P2P	non strict	10	noc
+1 ... +9	Deprecated	-	-	-
0	P2P (normal HCM Agreement **)	strict	1 or 10*	noc
-1	P2L (border-/line normal HCM Agreement**)	strict	1 or 10*	10m
-2 ... -8	Deprecated	-	-	-
-9	P2L	non strict	10	3
-10	P2L	non strict	10	10
-11	P2L	non strict	50	3

P2P = point-to-point calculation

P2L = point-to-(border)line calculation

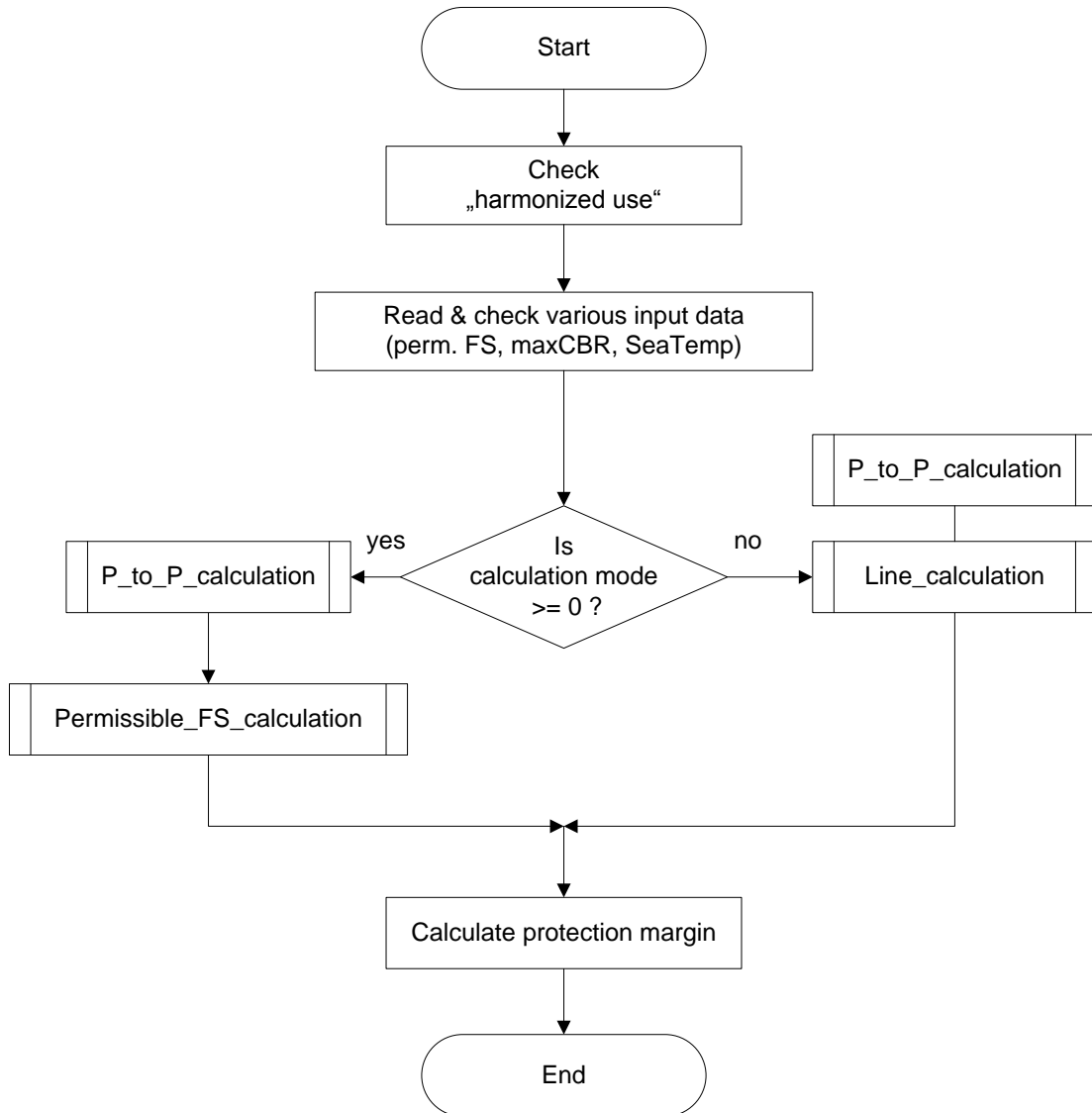
noc = no change

strict = Table of Annex 1 of the HCM Agreement

\* = derived from input value 10Z (Channel occupation)

\*\* = Normal HCM Agreement (or strict) calculation mode means calculation for all services covered by Table of Annex 1 of the HCM Agreement with values in rightmost column

Chapter 1: HCMMS\_V7 subroutine (part 2)



### **Check harmonized use**

For “harmonized use” it has to be the same service in the same band (at the moment TETRA only).

### **Is the calculation mode < 0?**

A decision is made, if it is a line calculation (calculation mode is negative) or a point-to-point calculation (calculation mode is 0 or positive).

### **Subroutine P\_to\_P\_calculation**

If the calculation mode is 0 or positive, a point-to-point calculation is performed. This process is described in chapter 2.

### **Subroutine Permissible\_FS\_calculation**

The permissible field strength is calculated. This process is described in chapter 4.

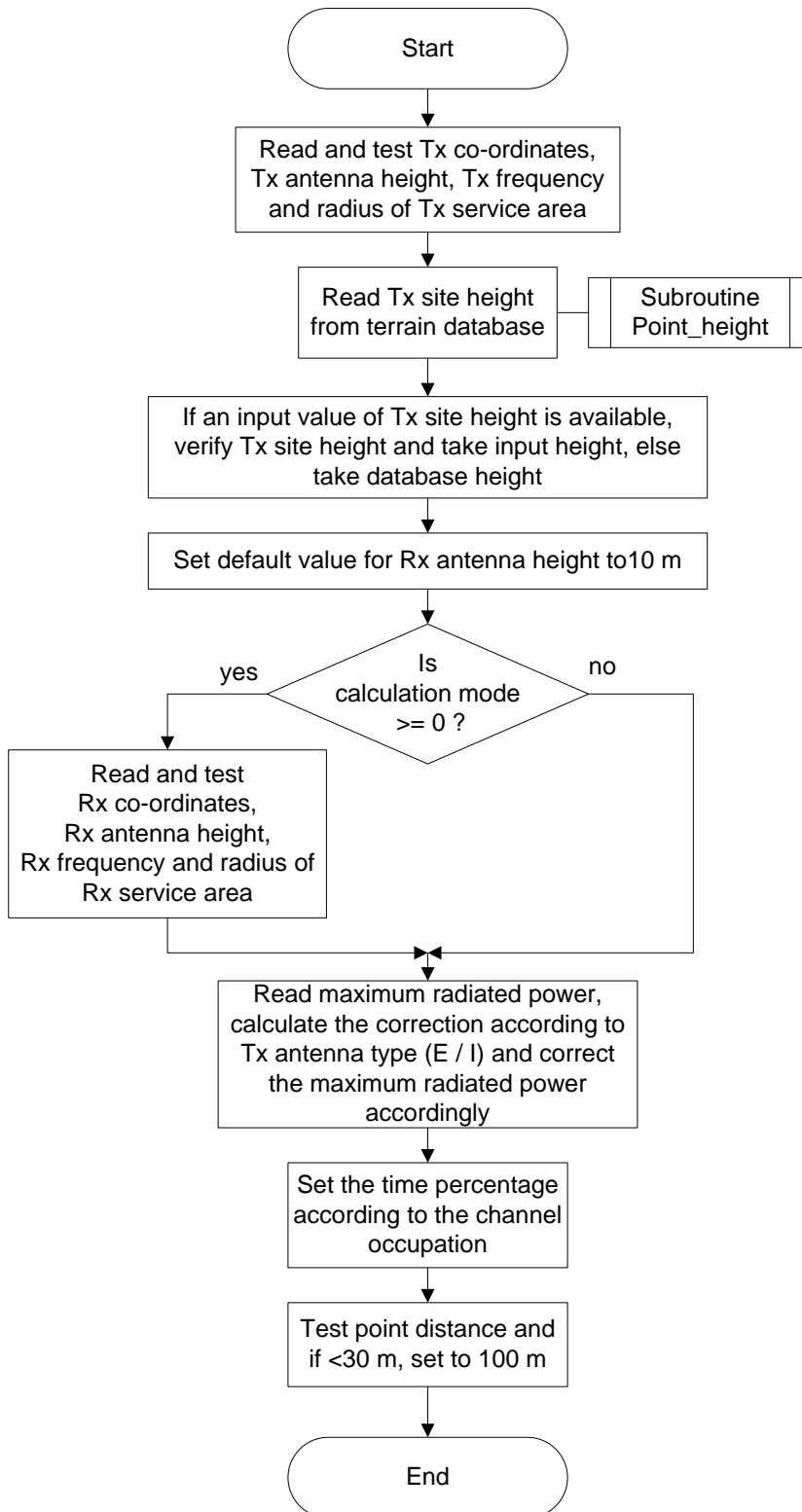
### **Subroutine Line\_calculation**

The maximum field strength on a line or the cross-border field strength is calculated. This process is described in chapter 3. This subroutine is using the subroutine P\_to\_P\_calculation.

### **Calculate the protection margin**

The protection margin is calculated. It is the permissible field strength minus the calculated field strength.

Chapter 1.1: Read and test input values



### **Read and test Tx co-ordinates, Tx antenna height, Tx frequency and radius of Tx service area**

The Tx data is read and tested. If an error in the format of data occurs, an HCM\_error code is generated (see chapter 11).

### **Read Tx site height from terrain database**

The site height of Tx is read from the terrain database with the Point\_height subroutine. This subroutine is described in chapter 5.3.

### **If an input value of Tx site height is available, verify Tx site height and take input height, else take database height**

If an input value for the Tx site height is available, this input height is compared with the height of the terrain database. If the heights differ, an Info(i) value is set depending on the difference value.

### **Read and test Rx co-ordinates, Rx antenna height, Rx frequency and radius of Rx service area**

In case of point-to-point calculations, the Rx values are read and tested if an error in the format of data occurs; an HCM\_error code is generated (see chapter 11).

### **Read maximum radiated power, calculate the correction according to Tx antenna type (E / I) and correct the maximum radiated power accordingly**

The value of the maximum radiated power is read, tested and if the type of the antenna is 'I', this value is reduced by 2.1 dB.

### **Set the time percentage according to the channel occupation**

If the channel occupation is 1, than the time percentage is set to 1%, else it is set to 10%.

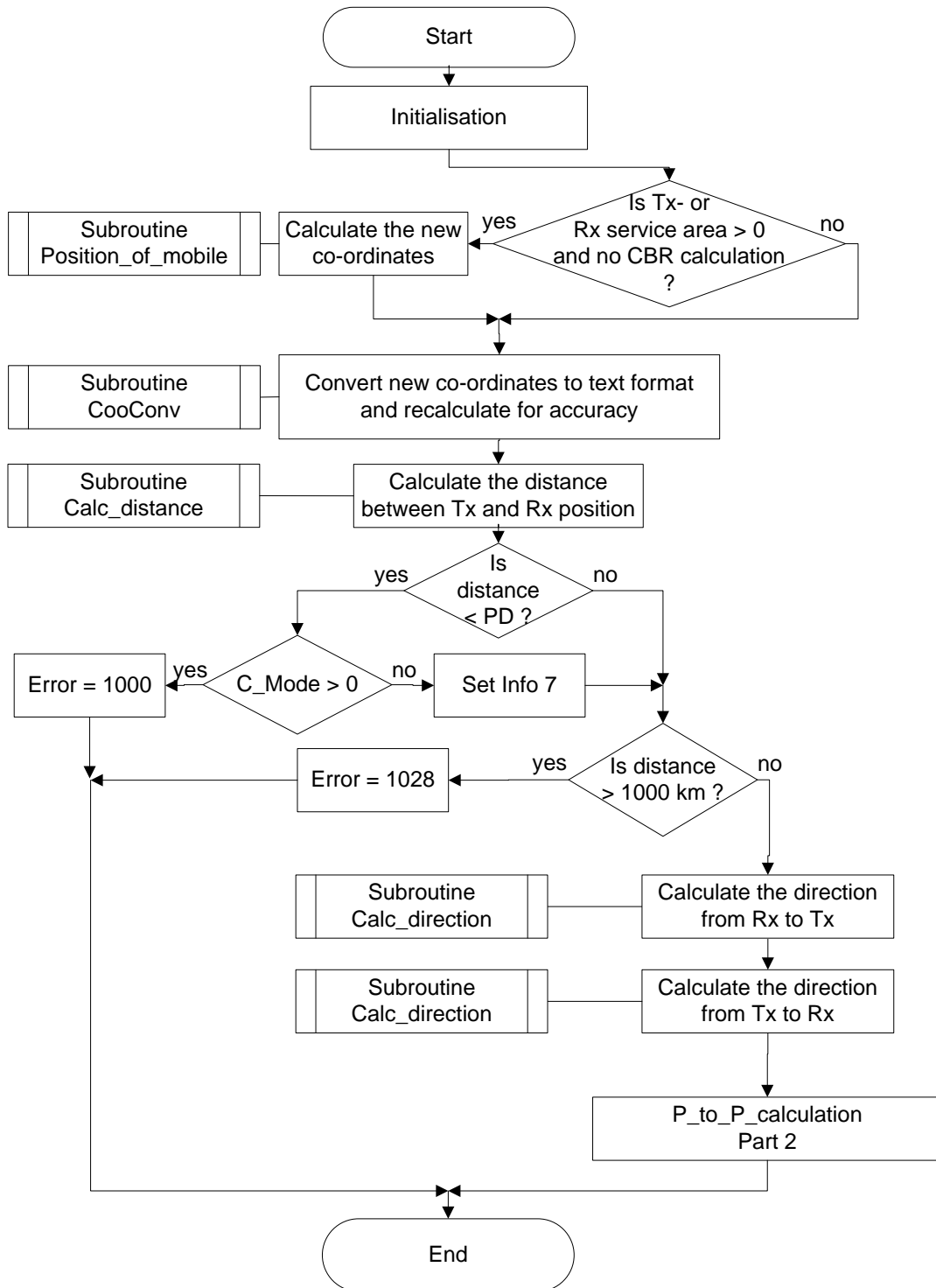
### **Test point distance and if <30 m, set to 100 m**

The point distance for the profile is an input value (PD) and is normally 100 m. If this value is less than 30 m, it is set to the default value of 100 m. For harmonized calculations the point distance should be set to 100 m. The default value of 100 m is chosen because the grid size of the HCM topo data is 3" by 3" or 3" by 6", which is approximately 100 m.

# CHAPTER 2: SUBROUTINE P\_TO\_P\_CALCULATION

## PART 1

Chapter 2: Subroutine P\_to\_P\_calculation  
Part 1



@'15

## Initialization

The new coordinates for Tx and Rx in a first step are set to the input values. All info values relating to this subroutine are cleared. Default field strength is set to 999.9.

## Calculate the new co-ordinates

The new co-ordinates of Tx and / or Rx are calculated with the subroutine Position\_of\_mobile. This subroutine is described in chapter 2.1. The new co-ordinates are only calculated with the Position\_of\_mobile subroutine, if there is no cross-border calculation! In case of cross-border calculation, the position of a mobile Tx is calculated in the subroutine CBR\_Coordinates which is described in chapter 3.1.

## Convert the new co-ordinates to text format and recalculate for accuracy

The new calculated co-ordinates are converted from internal decimal format to text format with the subroutine CooConv. This subroutine is described in chapter 5.8. Co-ordinates are converted because they are given as an output value in degrees, minutes and seconds. To cope for loss of resolution by conversion to text format, decimal coordinates are recalculated from text.

## Calculate the distance between Tx and Rx position

The distance between Tx and Rx position is calculated with the subroutine Calc\_distance. This subroutine is described in chapter 5.1.

## Is dist < PD? Is C\_mode < 0?

If the distance is smaller than the point distance then set Info(7) for (border)line calculations or return error=1000 for P2P.

## Is distance > 1000 km?

The range of the propagation curves is limited to 1000 km, so no calculation is possible beyond this distance.

## Calculate the direction from Rx to Tx Calculate the direction from Tx to Rx

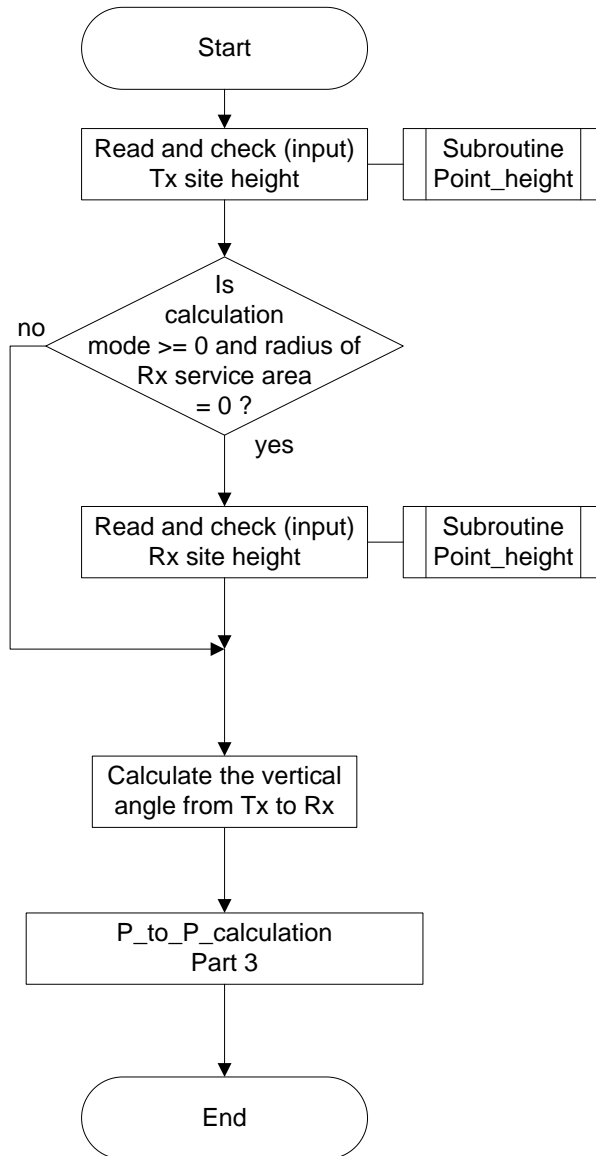
The directions are calculated with the subroutine Calc\_direction. This subroutine is described in chapter 5.2.



## **P\_to\_P\_calculation Part 2**

This process is described in the next flow chart.

Chapter 2: Subroutine P\_to\_P\_calculation  
Part 2



### **Read and check (input) Tx site height**

If an input for Tx site height is given, it is checked against the topo database else the value from the topo-db is taken.

The site height is read from the topo-db with the Point\_height subroutine. This subroutine is described in chapter 5.3.

### **Is C\_mode $\geq$ 0 and Rx service area=0?**

If it is a calculation to a real Rx (no mobile, no line) we need actual Rx site height.

### **Read and check (input) Rx site height**

The site height is either input or read from the topo-db with the Point\_height subroutine.

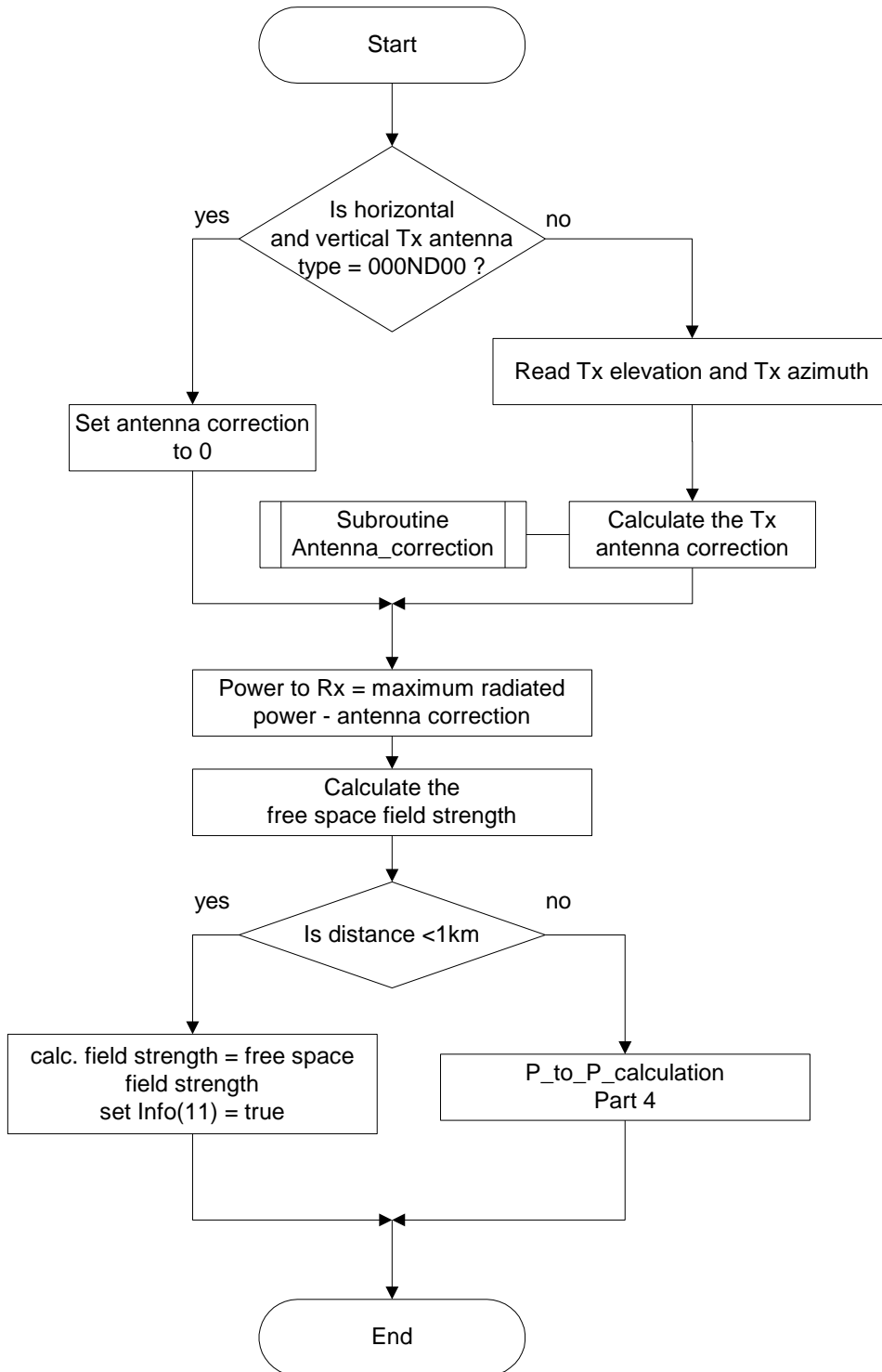
### **Calculate the vertical angle from Tx to Rx**

The vertical angle from Tx to Rx is calculated.

### **P\_to\_P\_calculation Part 3**

This process is described in the next flow chart.

Chapter 2: Subroutine P\_to\_P\_calculation  
Part 3



### **Is horizontal and vertical Tx antenna type = 000ND00?**

If the Tx antenna is a non-directional antenna, the antenna correction is set to 0, else the antenna correction is calculated.

### **Calculate the Tx antenna correction**

Taking into account the horizontal and vertical antenna types and the horizontal and vertical difference angles, the antenna correction of the Tx antenna is calculated with the Antenna\_correction subroutine. This subroutine is described in chapter 5.7.

### **Calculate the free space field strength**

The free space field strength is calculated with the formula:

$$\text{Free space field strength} = 77.0 - 20 * \log(\text{Distance}) + \text{Power to Rx}$$

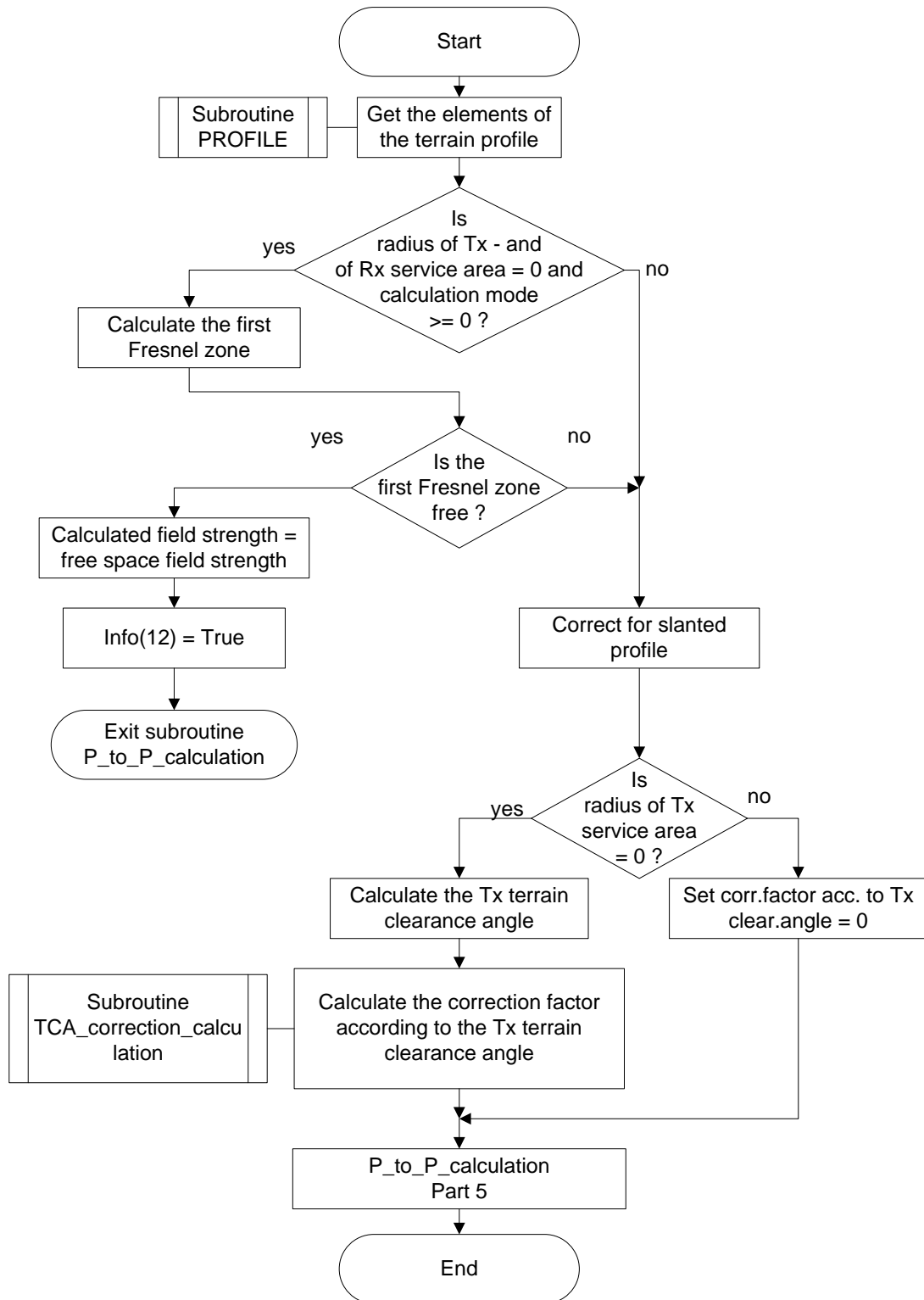
### **Calculated field = free space field strength set Info(11) = true**

The calculated field strength is set equal to the free space field strength for distances less than 1 km, because the propagation curves only start at 1 km. The Info(11) value is set to indicate this situation.

### **P\_to\_P\_calculation Part 4**

This process is described in the next flow chart.

Chapter 2: Subroutine P\_to\_P\_calculation  
Part 4



@'15

### **Get the elements of the terrain profile**

Between Tx and Rx position, all terrain heights in a defined grid are read from the terrain database. This is done with the PROFILE subroutine. This subroutine is described in chapter 5.5.

### **Calculate the first Fresnel zone**

Only if Tx and Rx are fixed stations and only in the case of a point to point calculation, the first Fresnel zone is calculated and it is checked, if it is free. If it is free, the calculated field strength is set equal to the free space field strength and the Info(12) value is set to indicate this situation.

### **Calculate the Tx clearance angle**

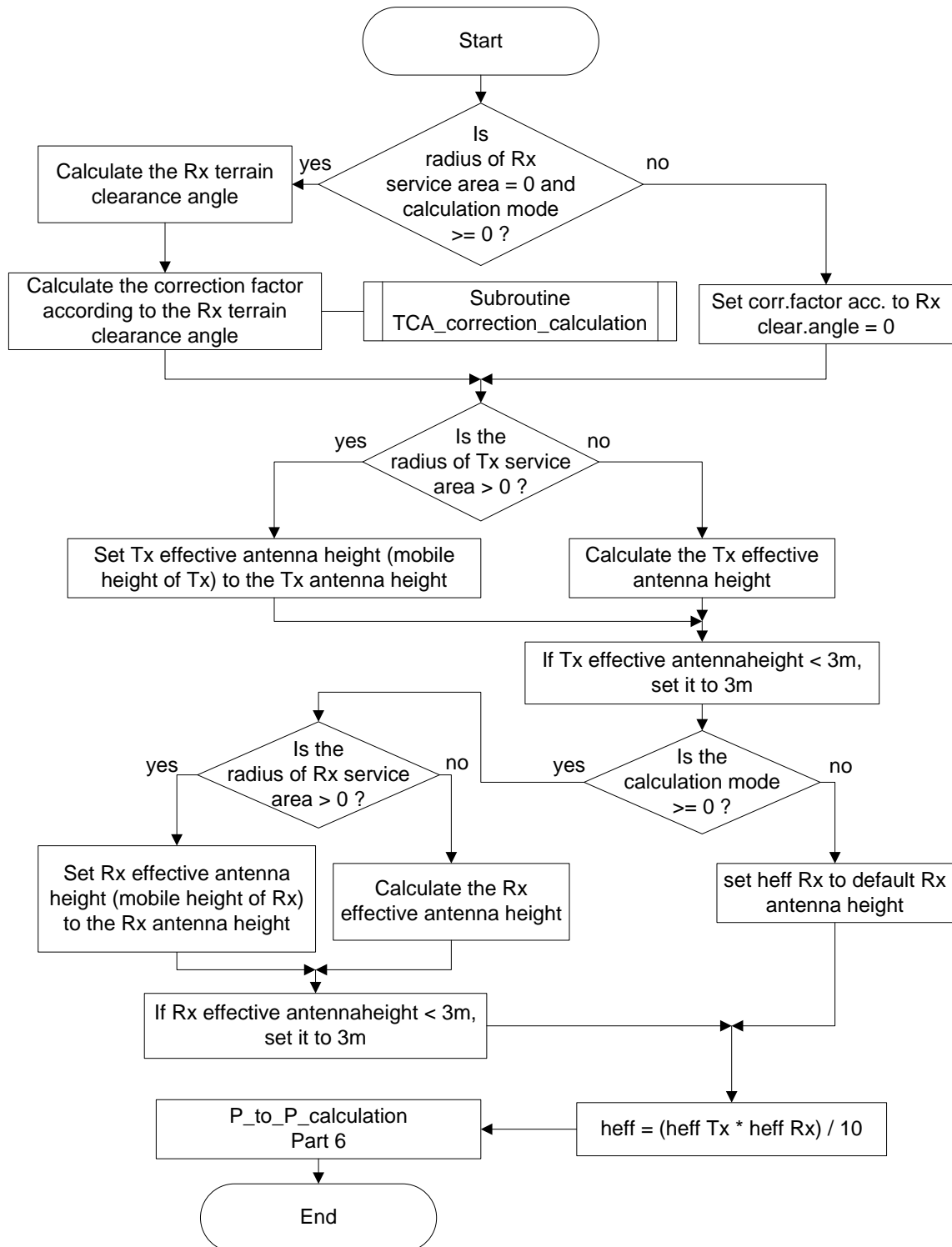
#### **Calculate the correction factor according to the Tx terrain clearance angle**

Depending on C\_mode prepare for slanted/normal profile and if Tx is a fixed station, the terrain clearance angle and further the correction factor is calculated with the subroutine TCA\_correction\_calculation (as described in chapter 5.9.) else set to zero.

### **P\_to\_P\_calculation Part 5**

This process is described in the next flow chart.

Chapter 2: Subroutine P\_to\_P\_calculation  
Part 5



@'15



**Is radius of Rx Service area=0 and C\_mode>=0?**

**Calculate the Rx clearance angle**

**Calculate the correction factor according to the Rx terrain clearance angle**

If Rx is not a mobile and not virtual (line calculation), the terrain clearance angle and further the correction factor is calculated with the subroutine TCA\_correction\_calculation (as described in chapter 5.9.) else set to zero.

**Is the radius of Tx service area > 0?**

If the Tx is a mobile, the effective antenna height of the Tx is set to the input value of the Tx antenna height, else the effective antenna height of the Tx is calculated.

**Is the radius of Rx service area > 0?**

Only in case of a point-to-point calculation, it is checked if the Rx is a mobile.

If the Rx is a mobile, the effective antenna height of the Rx is set to the input value of the Rx antenna height, else the effective antenna height of the Rx is calculated.

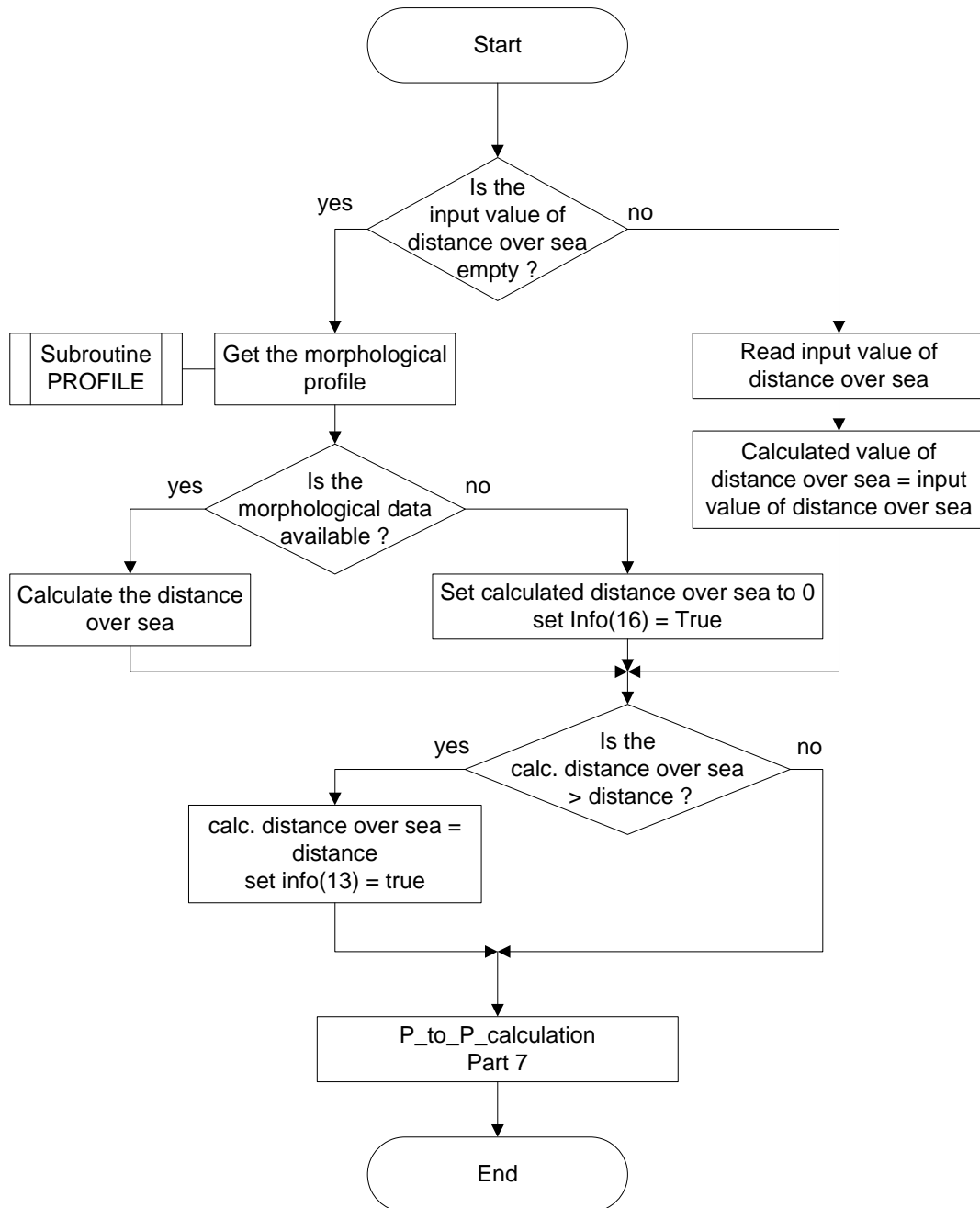
**Is the calculation mode >= 0?**

This subroutine is also used inside the subroutine Line\_calculation. For line calculation the effective antenna height of the receiver is set to the antenna height.

**P\_to\_P\_calculation Part 6**

This process is described in the next flow chart.

Chapter 2: Subroutine P\_to\_P\_calculation  
Part 6



@'15

### **Is the input value of distance over sea empty?**

If an input value of distance over sea is available, this value is read and used for the value of calculated distance over sea.

If no input value is available, the distance over sea is calculated taking the information of the morphological database.

### **Get the morphological profile**

The morphological profile is read with the PROFILE subroutine. This subroutine is described in chapter 5.5.

### **Is the morphological data available?**

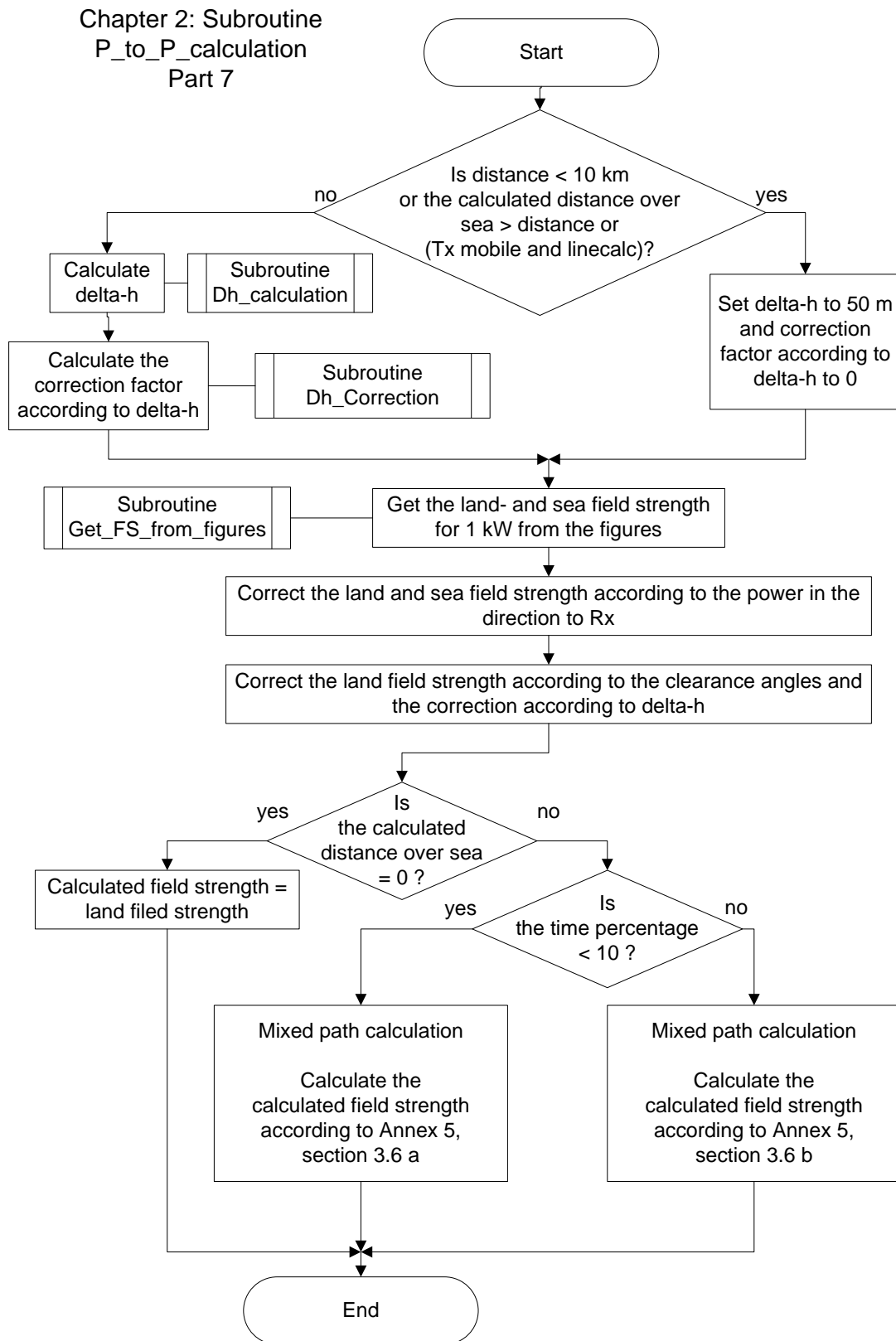
If the morphological data is available, the distance over sea is calculated, else it is set to 0 and the Info(16) value is set to indicate this situation.

### **Is the calculated distance over sea > distance?**

If the calculated distance over sea is greater than the distance between Tx and Rx (which may be the case if you supply an input value of distance over sea), the calculated distance is set to the distance between Tx and Rx and Info(13) is set to indicate this situation.

### **P\_to\_P\_calculation Part 7**

This process is described in the next flow chart.



@'15

### **Calculate delta-h**

If the distance is less than 10 km or there is no land path or Tx is mobile and it is line calculation, then delta-h is set to 50m, else the terrain irregularity delta-h is calculated with the subroutine Dh\_calculation.

The Subroutine Dh\_calculation is described in chapter 5.10.

### **Get the land- and sea field strength for 1 kW from the figures**

The land – and sea field strength for 1 kW is calculated from the figure values. This is performed with the subroutine Get\_FS\_from\_figures. This subroutine is described in chapter 5.12.

### **Correct the land field strength according to the clearance angles and the correction according to delta-h**

The 1 kW land field strength is corrected with the correction factor according to the Tx terrain clearance angle, the correction factor according to the Rx terrain clearance angle and the correction factor according to the terrain irregularity.

### **Correct land- and sea field strength according to the power in direction to Rx**

Because the land- and sea field strength calculated up till now are for 1 kW, this values need to be corrected according to the real power.

### **Calculated field strength = land field strength**

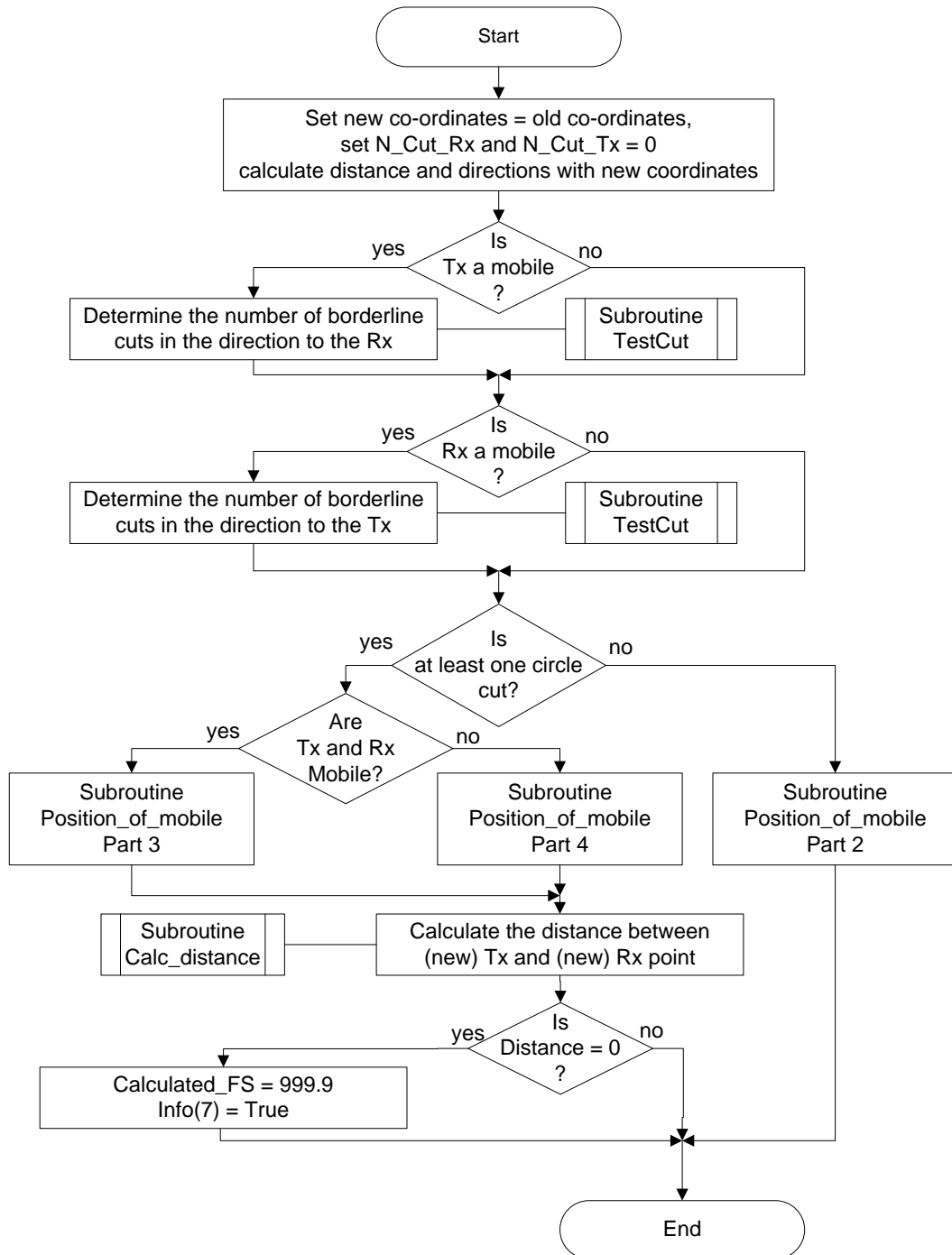
If the whole path is land, the calculated field strength is the land field strength.

### **Mixed path calculation**

Depending on the time percentage, the calculated field strength is calculated according to Annex 5, section 3.6a or section 3.6b.

PART 1

Chapter 2.1: Subroutine Position\_of\_mobile  
Part 1



This subroutine calculates the position of Tx or Rx for further calculation, taking into account the radius of the service areas and the border lines.

**Set new co-ordinates = old co-ordinates,  
set N\_cut\_Rx and N\_cut\_Tx to zero  
calculate distance and directions with new co-ordinates**

This subroutine is also used for line calculations. To ensure that for all calculations to all points of the calculation line the original Tx point together with distance and directions are used, the above mentioned settings and calculations are done.

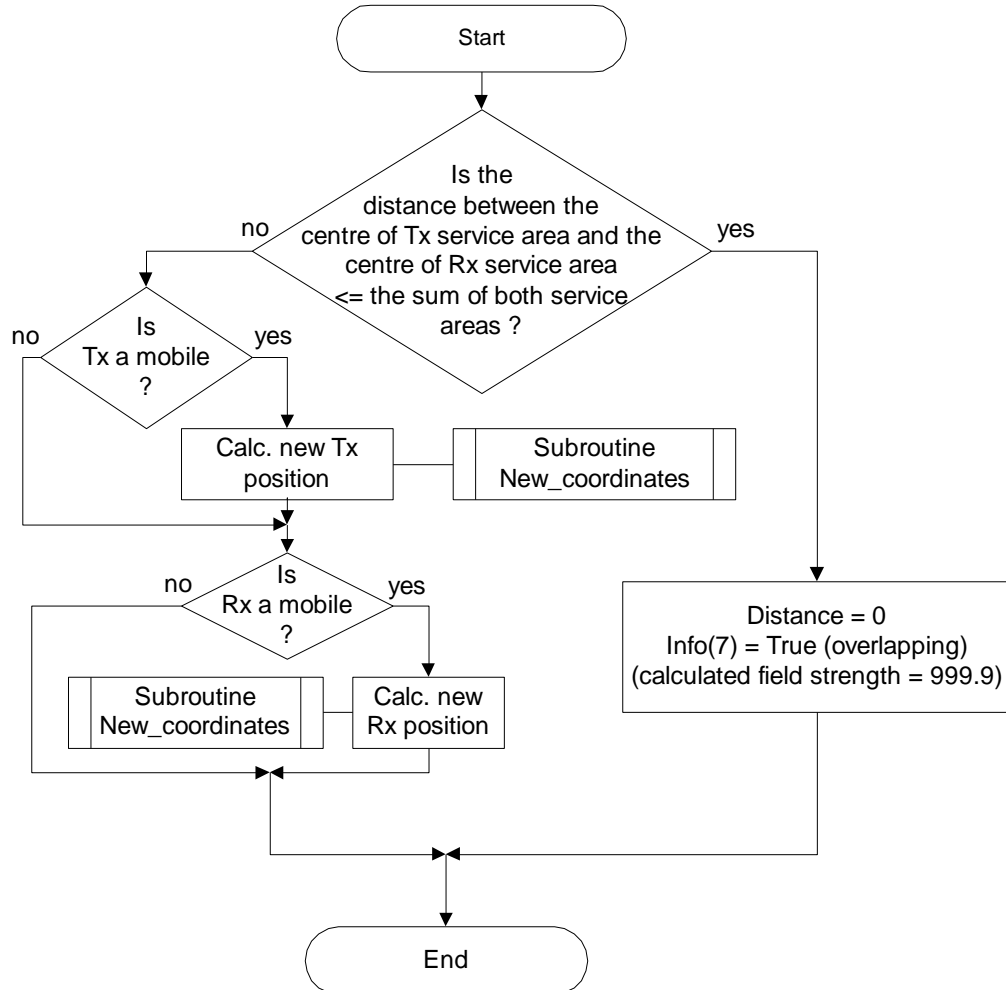
**Is Tx a mobile?  
Is Rx a mobile?**

If Tx, Rx or both are a mobile, the number of borderline cuts is determined (see subroutine TestCut). If at least one service area is cutting the border and both stations are mobiles, the program will use the subroutine Position\_of\_mobile Part 3. If at least one service area is cutting the border and only one station is a mobile, the program will use the subroutine Position\_of\_mobile Part 4. In case no service area is cut the program will continue with the subroutine Position\_of\_mobile Part 2.

**Calculate the distance between (new) Tx and (new) Rx point**

The distance is calculated using the subroutine Calc\_distance (see chapter 5.1). If this distance is 0, the calculated field strength is set to 999.9 and Info(7) is set to True to indicate the overlapping of the service areas.

Chapter 2.1: Subroutine Position\_of\_mobile  
Part2



@'15



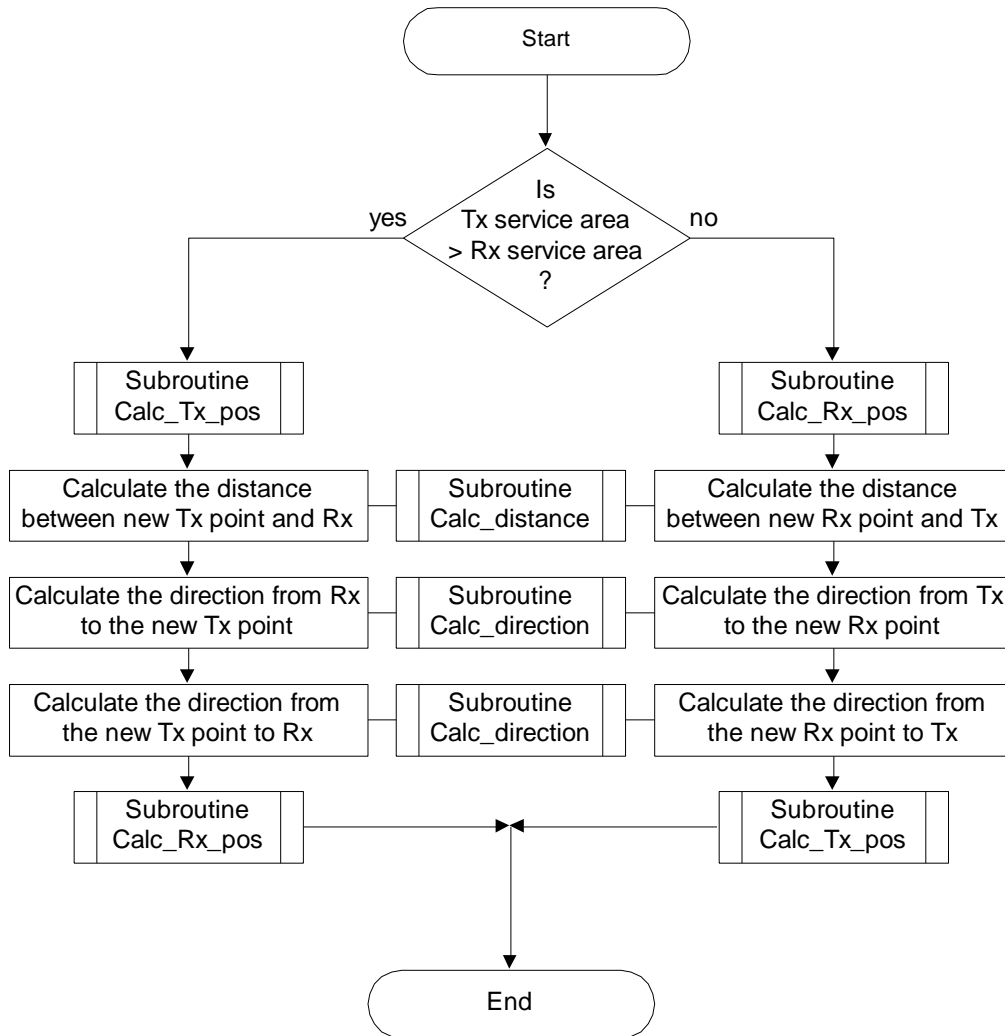
This flow-chart describes the situation when no borderline is cut.

**Is the distance between the centre of Tx service area and the centre of Rx service area  $\leq$  the sum of both service areas?**

If the service areas of both mobiles are overlapping, or if a fixed station is located within the service area of a mobile, the distance is set to zero, an info value (7) is given and the calculated field strength is set to 999.9.

If there is no overlapping, the closest positions between both mobiles, or between a mobile and a fixed station, is calculated. This is performed by using the subroutine New\_coordinates (see chapter 5.14).

Chapter 2.1: Subroutine Position\_of\_mobile  
Part 3



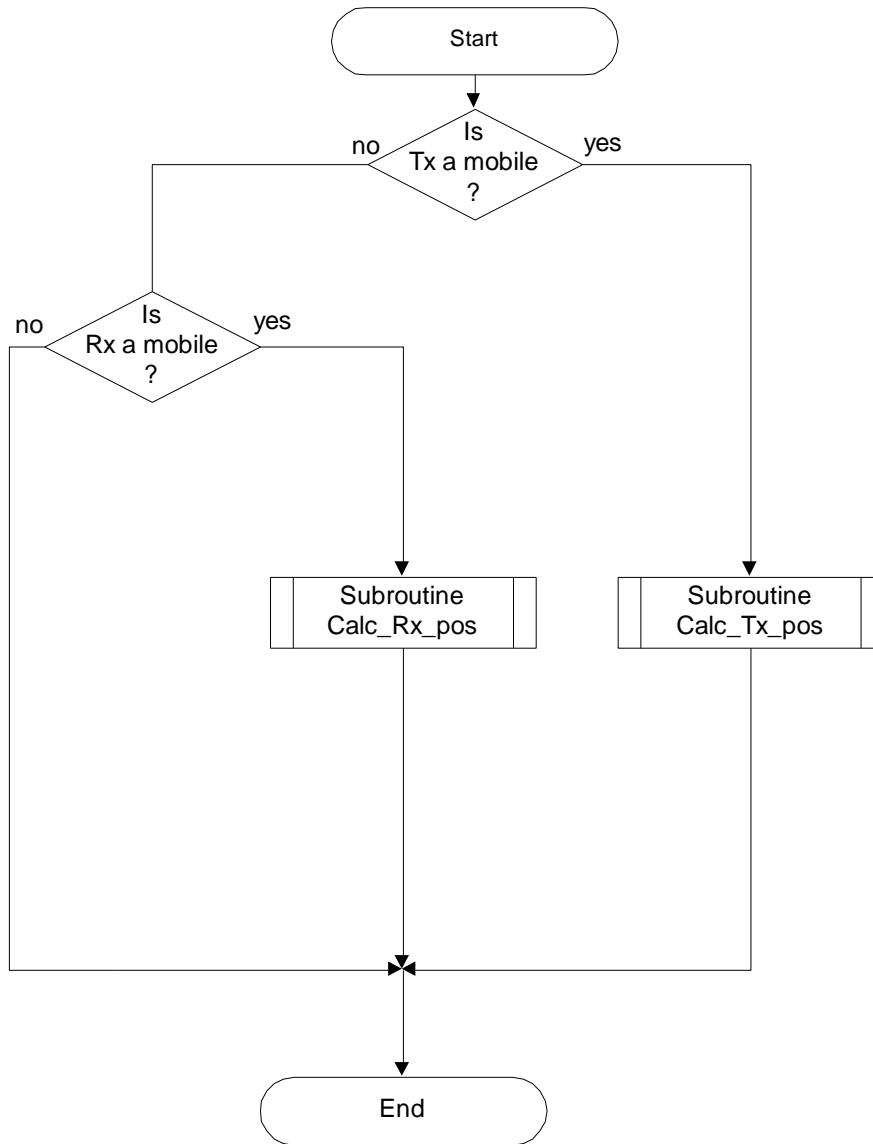
This flow-chart describes the situation when both stations are mobiles and at least one service area is cut by a borderline.

### Is Tx service area > Rx service area?

For correct further calculations it is important that the station with the bigger service area is calculated first. Only in this case you can ensure that the nearest point to the other station will be found in all special borderline situations.

The new position of the stations is calculated with the subroutines Calc\_Tx\_pos (see chapter 2.1.3) and Calc\_Rx\_pos (see chapter 2.1.4). Before the new position of the other station can be calculated, the distance and the direction between the two stations need to be recalculated.

Chapter 2.1: Subroutine Position\_of\_mobile  
Part 4



@'15

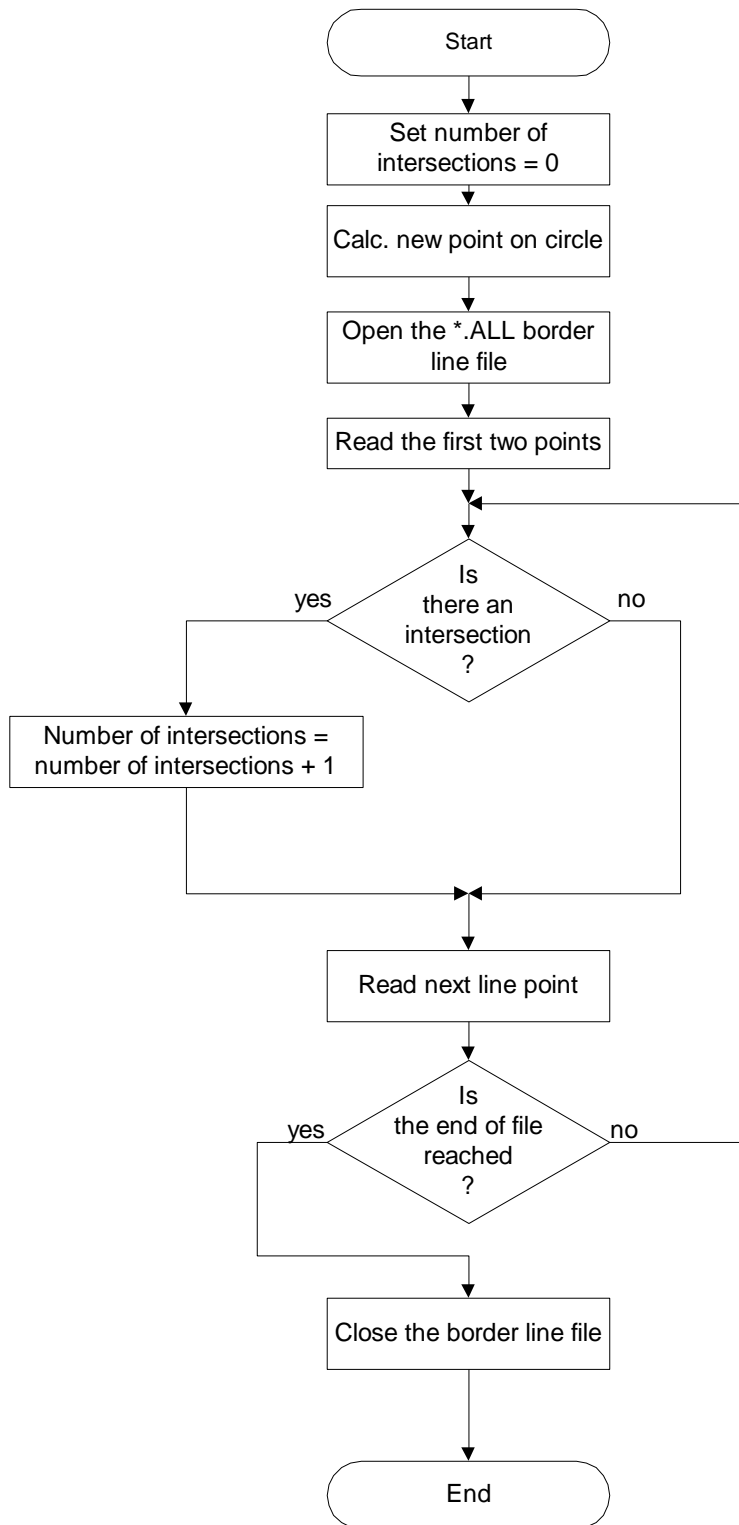
This flow-chart describes the situation when only one station is a mobile and the service area is cut by a borderline.

**Is Tx a mobile?**

**Is Rx a mobile?**

If the Tx is a mobile, the position of the mobile is calculated with subroutine Calc\_Tx\_pos (see chapter 2.1.3); if Rx is a mobile, the position of the mobile is calculated with subroutine Calc\_Rx\_pos (see chapter 2.1.3).

Chapter 2.1.1: Subroutine TestCut



@'15

This subroutine determinates the number of intersections between the complete borderline and the radius of the service area in the direction of the other station.

#### **Set number of intersections = 0**

Initialization of the counter.

#### **Calc new point on circle**

The point on the edge of the service area in the direction of the other station is calculated.

#### **Open the \*.ALL border line file**

The file containing the closed borderline of the own country is opened.

#### **Read the first two points**

The first two points of the closed borderline are read.

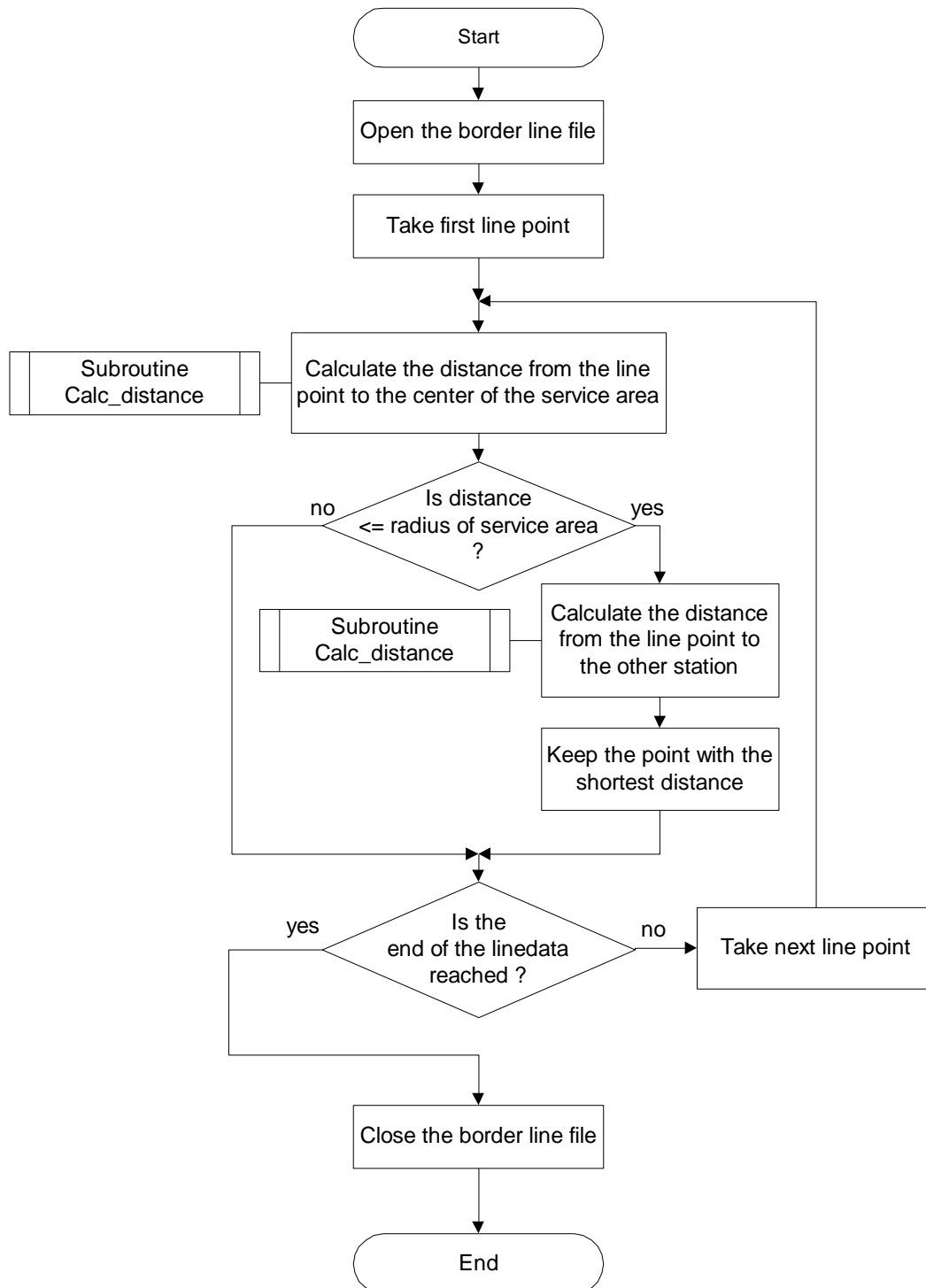
#### **Is there an intersection?**

It is tested if the radius of the service area intersects the line between the two points of the closed border line. If there is an intersection, the counter Number of intersections is incremented by 1.

#### **Read next line point**

The next point of the closed borderline is read until all points are checked.

Chapter 2.1.2: Subroutine NearestLinePoint



@'15



### **Calculate the distance from the line point to the center of the service area**

The distance is calculated with the subroutine Calc\_distance. This subroutine is described in chapter 5.1. This process is repeated for all points of the selected border line.

Only the points inside the service area are taken into account for further calculations.

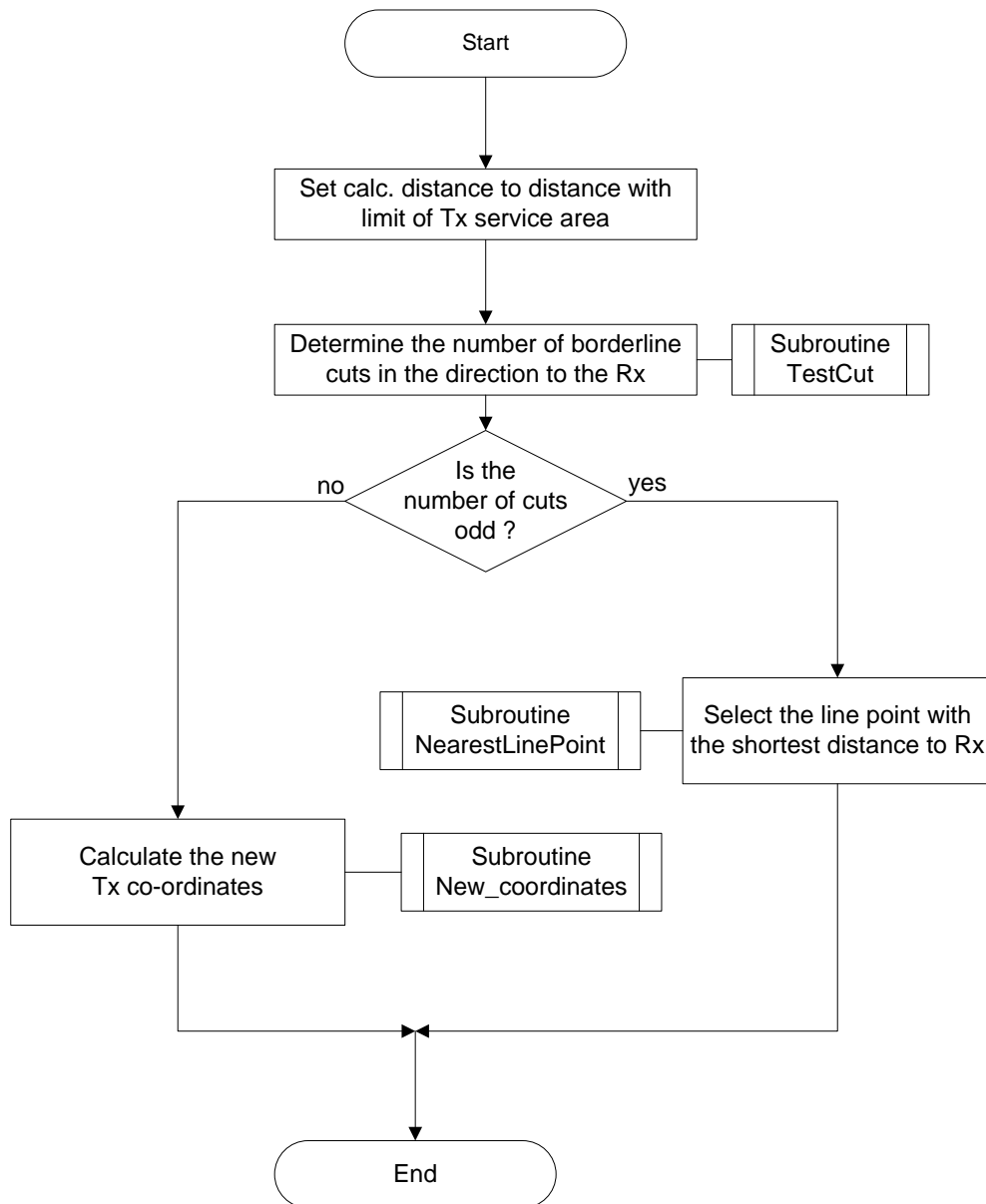
### **Calculate the distance from the new line point to the other station**

The distance is calculated with the subroutine Calc\_distance. This subroutine is described in chapter 5.1.

### **Keep the point with the shortest distance**

The line point with the shortest distance is kept. This point is the nearest line point.

Chapter 2.1.3: Subroutine Calc\_Tx\_pos



@'15

This subroutine calculates the position of Tx in case Tx is a mobile.

### **Set calc. distance to distance with limit of Tx service area**

For further calculations only the points on the edge of the service area are taken into account. The part of the borderline which is inside the circle of service area of the mobile is taken into account.

### **Determine the number of borderline cuts in the direction to the Rx**

This calculation determines the number of cuts of the borderline in the direction to Rx; if the number of cuts is even, this means that the edge of the service area of Tx in this direction is again in your own country. If the number of cuts is odd, this means that Rx is in the neighbouring country and the borderline is taken into account as part of the edge of the service area.

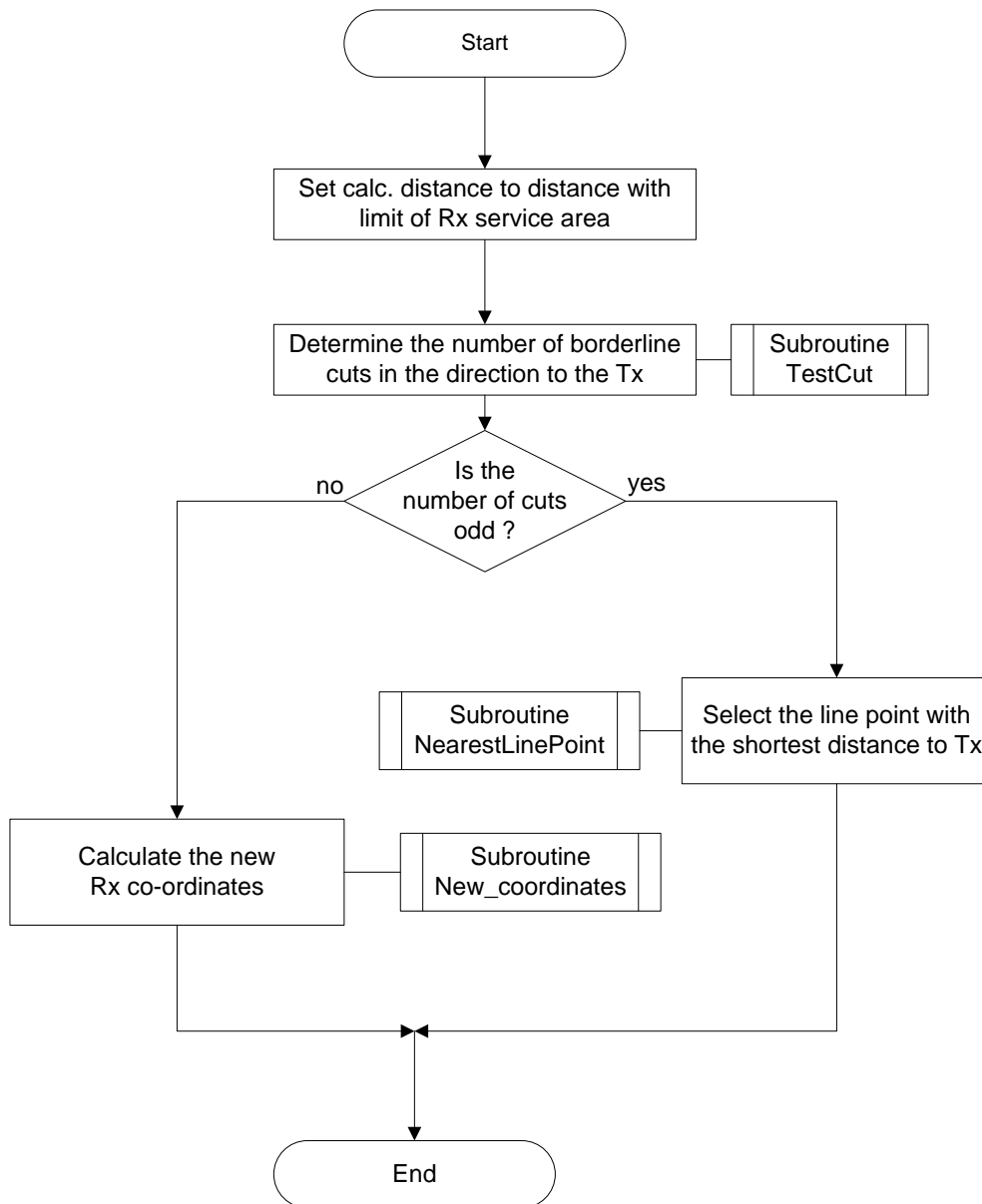
### **Calculate the new Tx co-ordinates**

The new Tx co-ordinates are on the circle representing the edge of the service area. The co-ordinates are calculated with the subroutine `New_coordinates` (see chapter 5.14).

### **Select the line point with the shortest distance to Rx**

The new Tx co-ordinates are on the part of the borderline cut by the circle representing the edge of the service area. The nearest point to Rx is selected by using the subroutine `NearestLinePoint` (see chapter 2.1.2).

Chapter 2.1.4: Subroutine Calc\_Rx\_pos



@'15

This subroutine calculates the position of Rx in case Rx is a mobile.

### **Set calc. distance to distance with limit of Rx service area**

For further calculations only the points on the edge of the service area are taken into account. The part of the borderline which is inside the circle of service area of the mobile is taken into account.

### **Determine the number of borderline cuts in the direction to the Tx**

This calculation determines the number of cuts of the borderline in the direction to Tx; if the number of cuts is even, this means that the edge of the service area of Rx in this direction is again in your own country. If the number of cuts is odd, this means that Tx is in the neighbouring country and the borderline is taken into account as part of the edge of the service area.

### **Calculate the new Rx co-ordinates**

The new Rx co-ordinates are on the circle representing the edge of the service area. The co-ordinates are calculated with the subroutine New\_coordinates (see chapter 5.14).

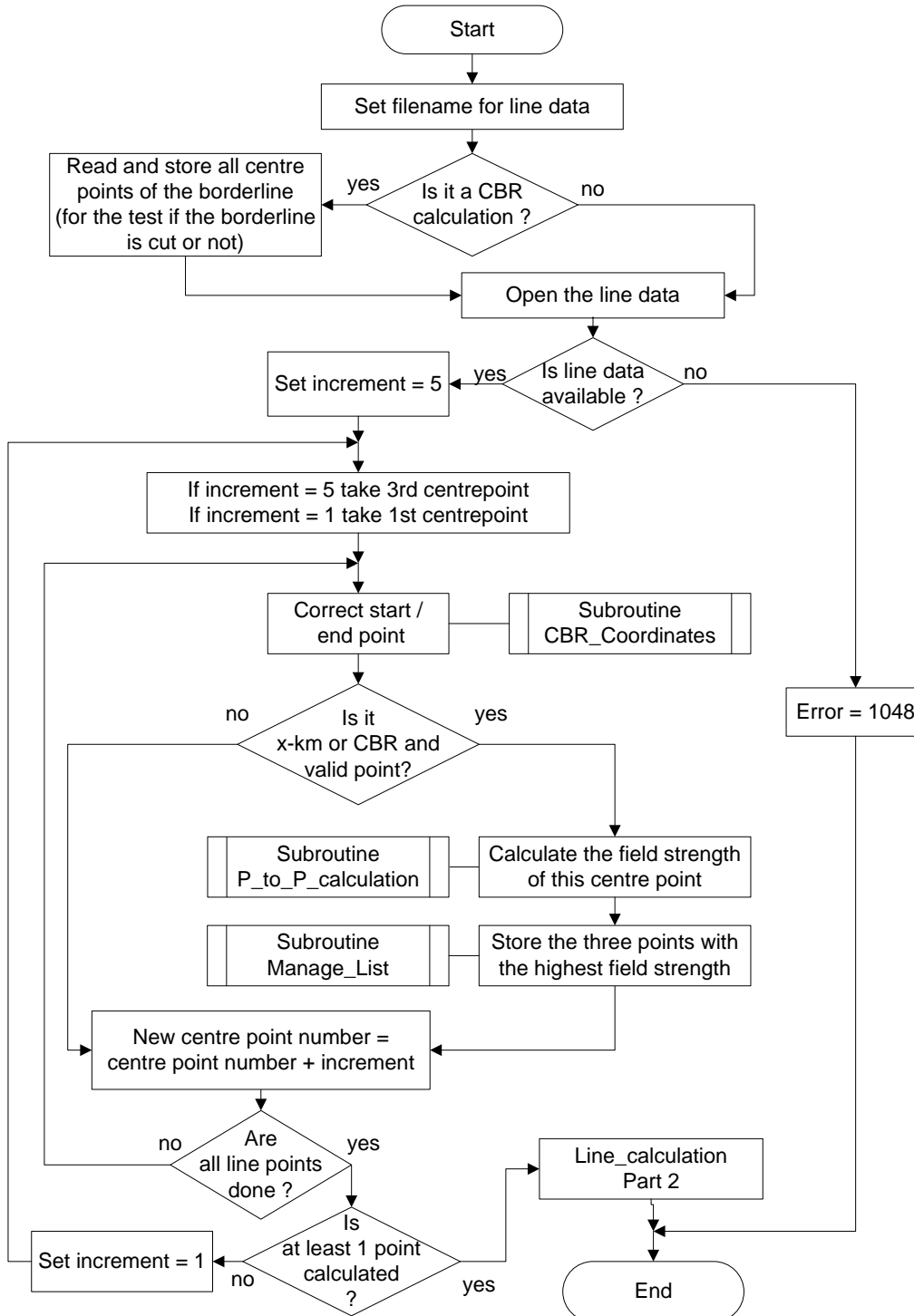
### **Select the line point with the shortest distance to Tx**

The new Rx co-ordinates are on the part of the borderline cut by the circle representing the edge of the service area. The nearest point to Tx is selected by using the subroutine NearestLinePoint (see chapter 2.1.2).

# CHAPTER 3: SUBROUTINE LINE\_CALCULATION

## PART 1

### Chapter 3: Subroutine Line\_calculation Part 1



@'15

All line data are stored in records. Each record contains 10 line points following each other and an additional 11th “center point” of these 10 points.

The line calculation is performed in a iteration process to shorten the calculation process.

First, the calculation is performed to each 5th centre point of the line data, starting at the 3rd centre point. The record numbers of the three points with the highest field strengths are stored. (e.g. for center points 13, 38 and 63) if no point with relevant fs is found, this part is repeated for every centre point.

Second it is calculated to the +/- 2 neighbouring centre points of the previous stored records. Again the three record numbers for the points with the highest field strengths are stored. (e.g. center points 13, 37 and 67 are stored).

Finally each point inside the three stored records (30 points) is calculated; the point with the highest field strength is the final result.

### **Set filename for line data**

The filename for the line data is set using the country codes of the two involved country and the distance to the borderline (see description of the line data in chapter 8).

### **Read and store all centre points of the borderline**

In case of a cross-border range calculation, all centre points of the closed borderline of the whole affected country (xxx.ALL file) are read and stored for the test if a borderline point is cut or not (CBR calculations are only performed if a neighbouring country is affected).

### **Open the line data**

For all calculations the appropriate borderline, the CBR line or the X-km line is opened. In case the data is not available the program terminates with error code 1048.

### **Correct start/end point**

New Tx coordinates in case of mobile Tx and new reception point coordinates for a CBR calculation is calculated using subroutine CBR\_Coordinates. This subroutine is described in chapter 3.1.

### **Is it a x-km or CBR calculation and a valid point?**

Valid points are those points where at least one point of the propagation path is in the affected country, this is result of subroutine Test\_cut1 (inverse logic).

### **Calculate the field strength of this center point**

This calculation is performed using the subroutine P\_to\_P\_calculation that is described in chapter 2.

### **Store the record numbers for the three centre points with the highest field strength**

The record numbers for the three centre points with the highest field strength are stored using subroutine Manage\_List that is described in chapter 3.3.

### **Is at least 1 point calculated?**

For small co-ordination line files it may happen that no valid point is found with the highest increment. In this case this step is repeated for each centre point.

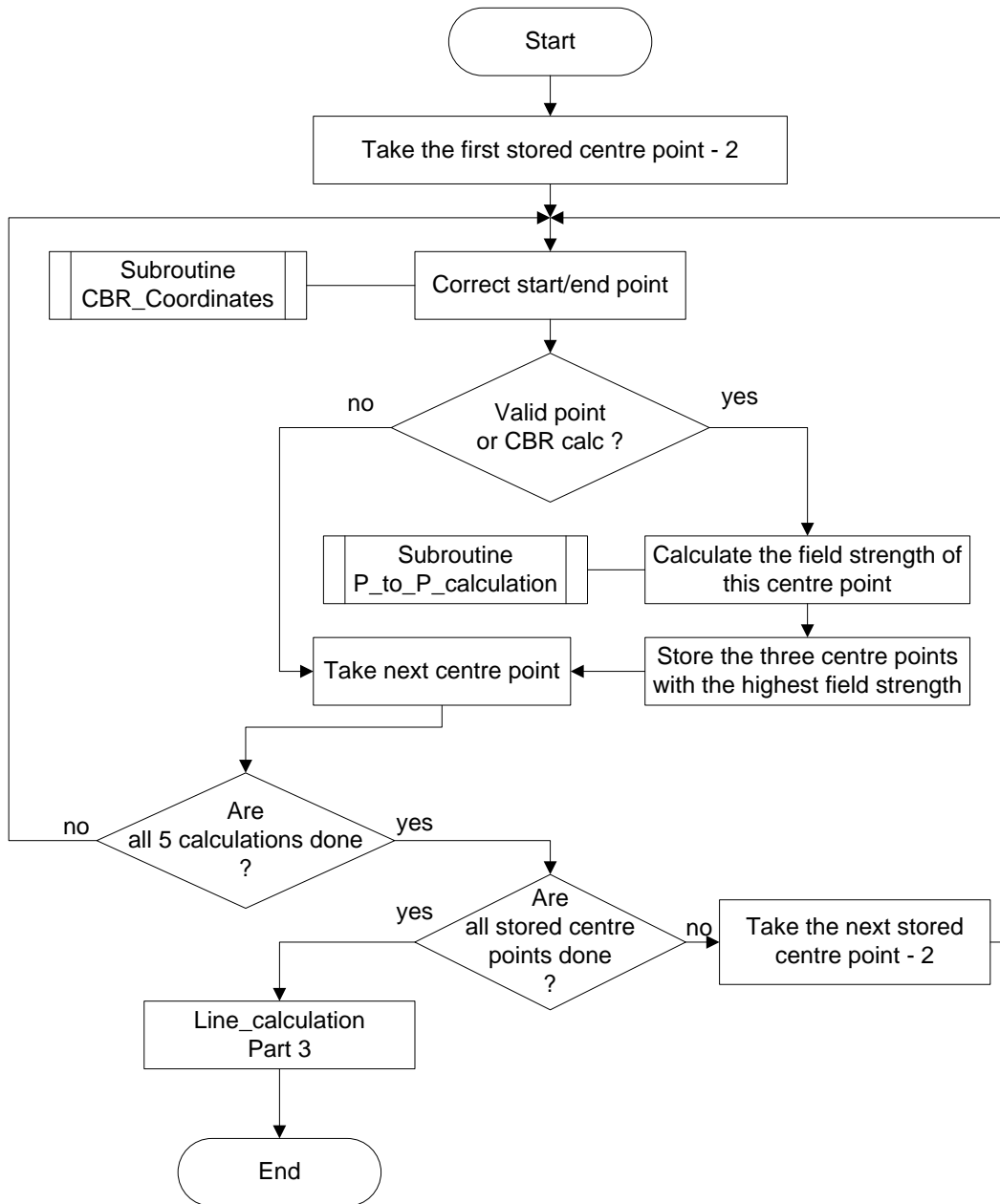
### **Line\_calculation Part 2**

This process is described in the next flow chart.



PART 2

Chapter 3: Subroutine  
Line\_calculation Part 2



@'15

### **Correct start/end point**

New Tx coordinates in case of mobile Tx and new reception point coordinates for a CBR calculation is calculated using subroutine CBR\_Coordinates. This subroutine is described in chapter 3.1.

### **Is it a x-km or CBR calculation and a valid point?**

Valid points are those points where at least one point of the propagation path is in the affected country, this is result of subroutine Test\_cut1 (inverse logic).

### **Calculate the field strength of this point**

This calculation is performed using the subroutine P\_to\_P\_calculation that is described in chapter 2.

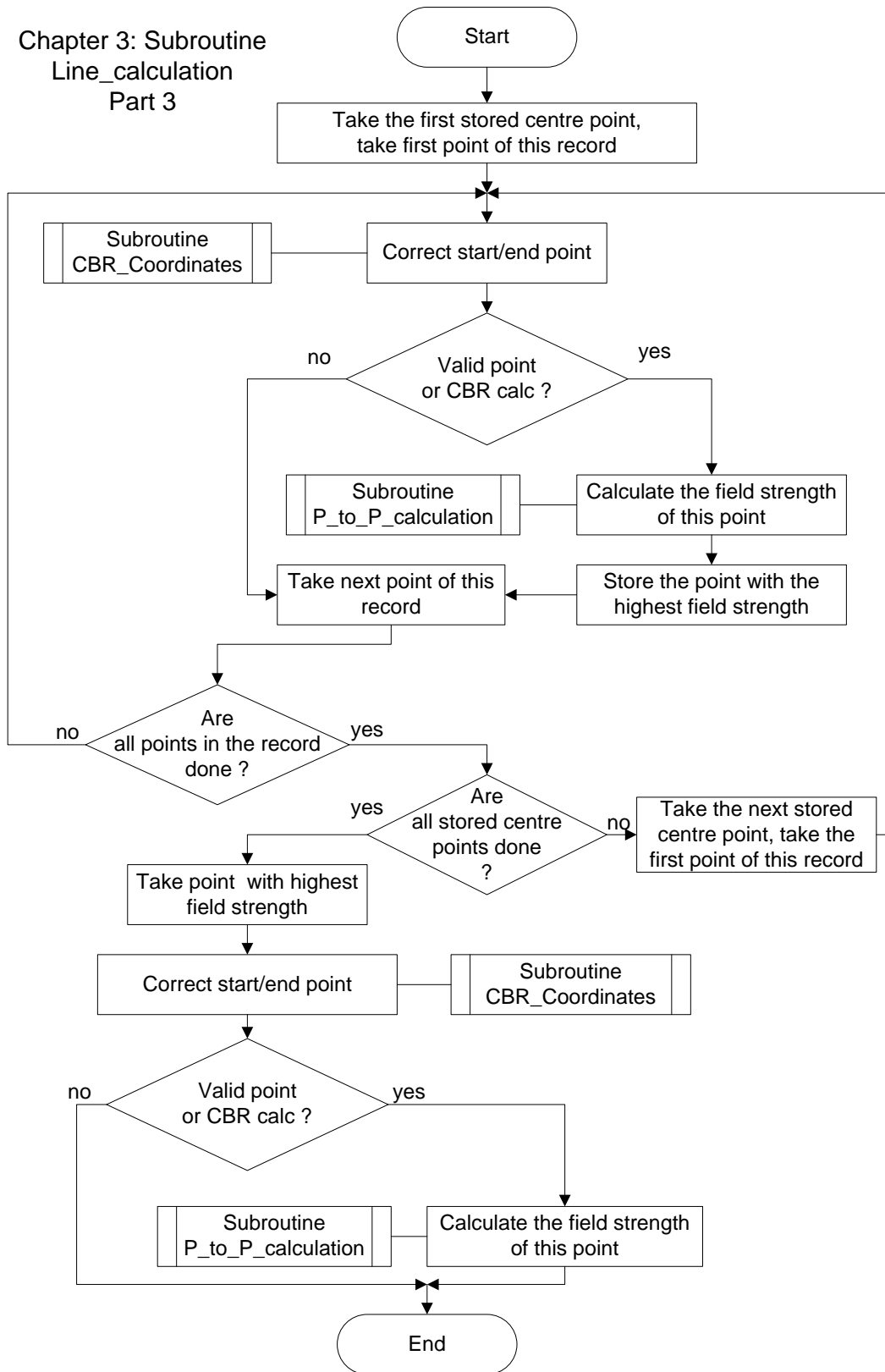
### **Store the three record numbers for centre points with the highest field strength**

The record numbers for the three centre points with the highest field strength are stored using subroutine Manage\_List that is described in chapter 3.3.

### **Line\_calculation Part 3**

This process is described in the next flow chart.

Chapter 3: Subroutine  
Line\_calculation  
Part 3



@'15

### **Correct start/end point**

New Tx coordinates in case of mobile Tx and new reception point coordinates for a CBR calculation is calculated using subroutine CBR\_Coordinates. This subroutine is described in chapter 3.1.

### **Is it a x-km or CBR calculation and a valid point?**

Valid points are those points where at least one point of the propagation path is in the affected country, this is result of subroutine Test\_cut1 (inverse logic).

### **Calculate the field strength of this point**

This calculation is performed using the subroutine P\_to\_P\_calculation that is described in chapter 2.

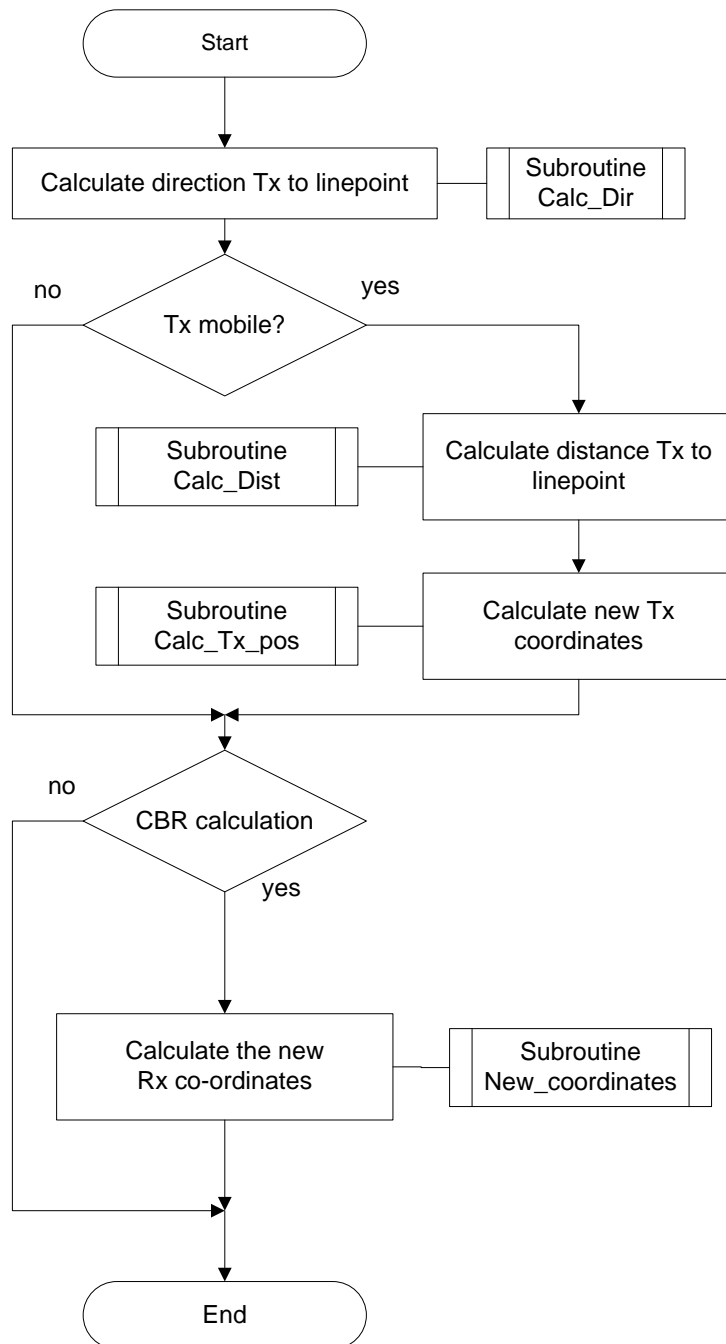
### **Store the point with the highest field strength**

If field strength is higher than maximum so far, replace stored point with current.

### **Recalculate details for point with highest field strength**

After the 30 calculations are done, the calculation to the point with the highest field strength has to be redone to get the correct output values. (At the end the highest might have been not the last and all information is lost due to following calculations).

Chapter 3.1: Subroutine CBR\_Coordinates



@'15

### **Calculate the direction from the Tx to the line point**

The direction is calculated with the subroutine Calc\_direction. This subroutine is described in chapter 5.2.

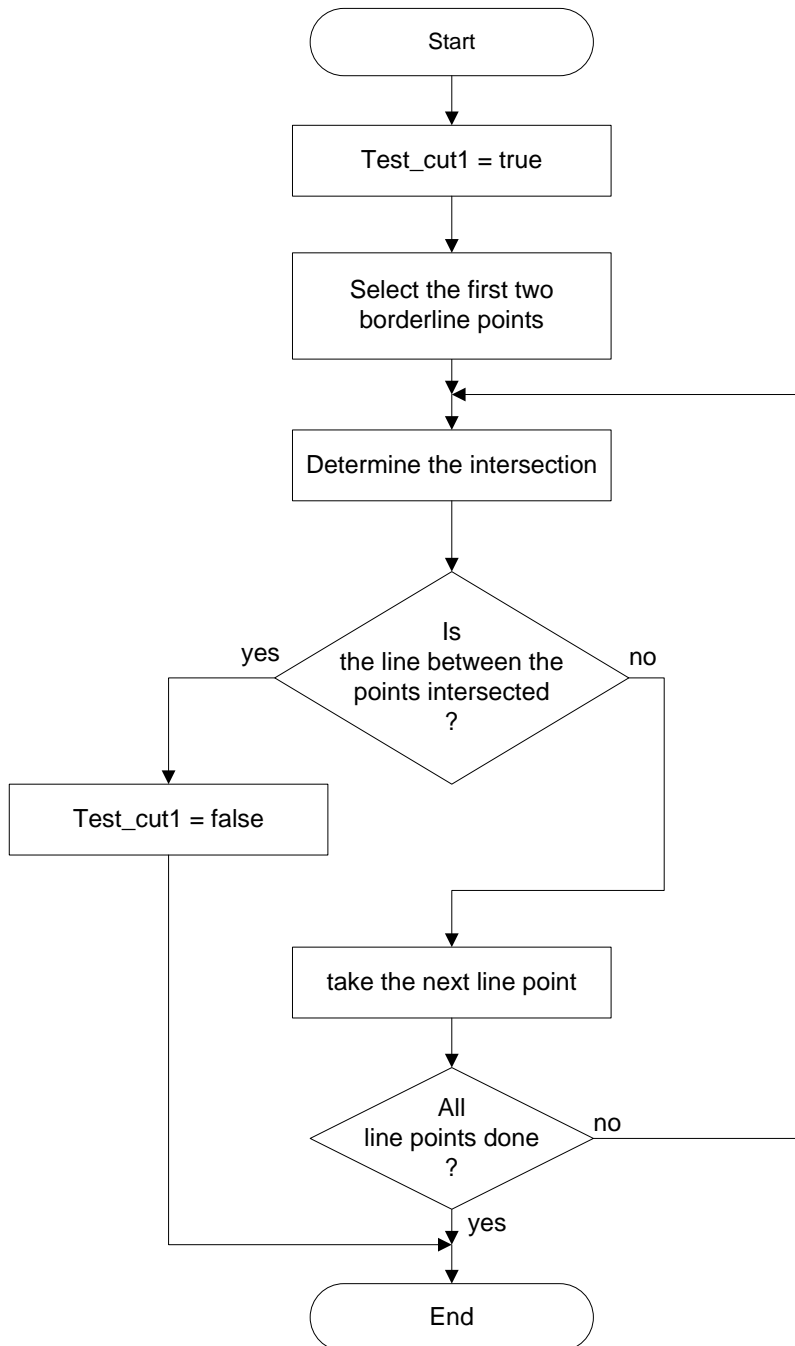
### **Tx mobile?**

If the transmitter is mobile, the coordinates only mark the centre of the operation area, the point closest to the relevant line point is calculated using Calc\_Tx\_pos

### **CBR calculation?**

If d2b (distance to border) is negative, then it is CBR calculation and new CBR co-ordinates are calculated with the subroutine New\_coordinates. This subroutine is described in chapter 5.14. Direction is same as Tx-borderlinepoint, distance is -CBR\_D (d2b).

Chapter 3.2: Subroutine Test\_cut1



@'15

This subroutine only determines if the closed borderline is cut. The number of cuts is not important.

This subroutine uses only the centre points of the borderline. These points are already stored in the subroutine Line\_calculation chapter 3 part 1.

### Select the first two line points

This process starts by selecting the first two borderline centre points.

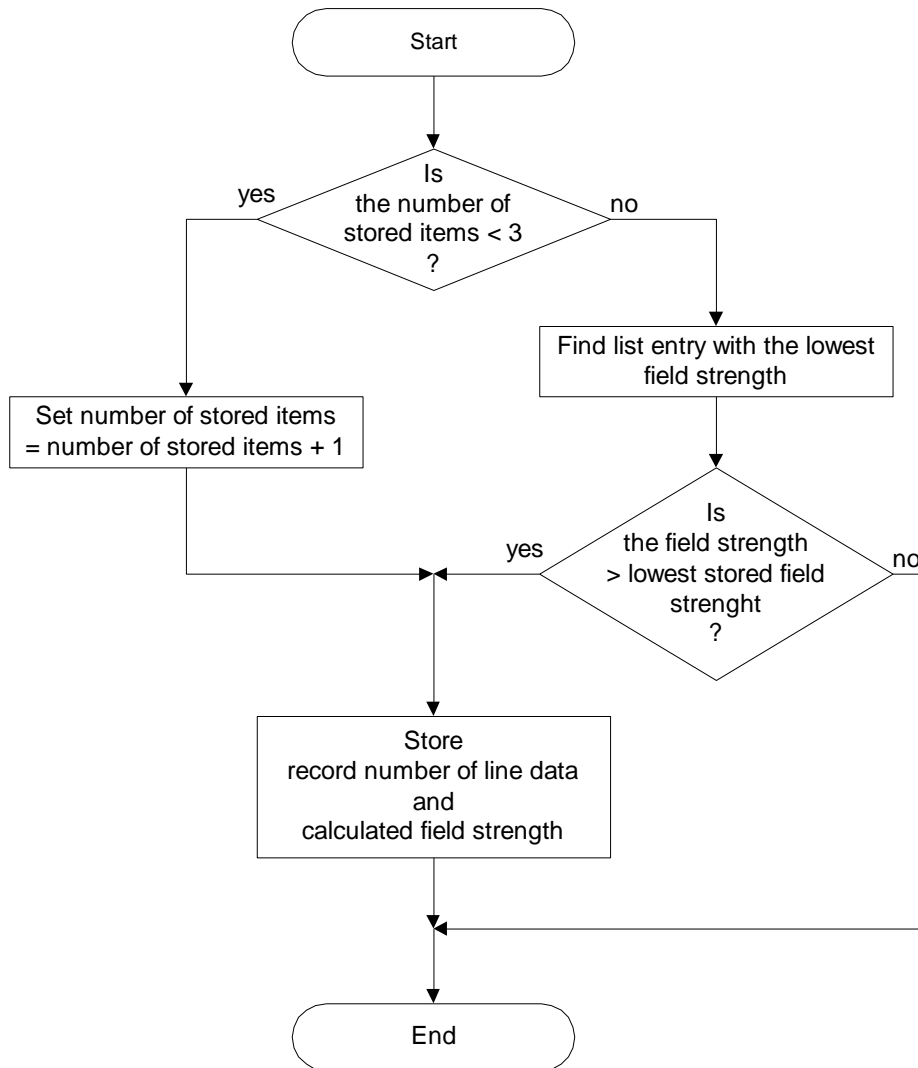
### Is the line between the points intersected?

If the line between any two neighbouring centre points (single line of the tested countries border polygon) is intersected by the propagation path once, test\_cut1 is set to false and the process terminates. Single cut is sufficient; there is no need to test additional points.

If the line is not intersected, the next line centre point is taken and the intersection of the line between the new point and the ending point of the previous test is tested and so on until all parts of the border polygon are tested.



Chapter 3.3: Subroutine Manage\_List



@'15

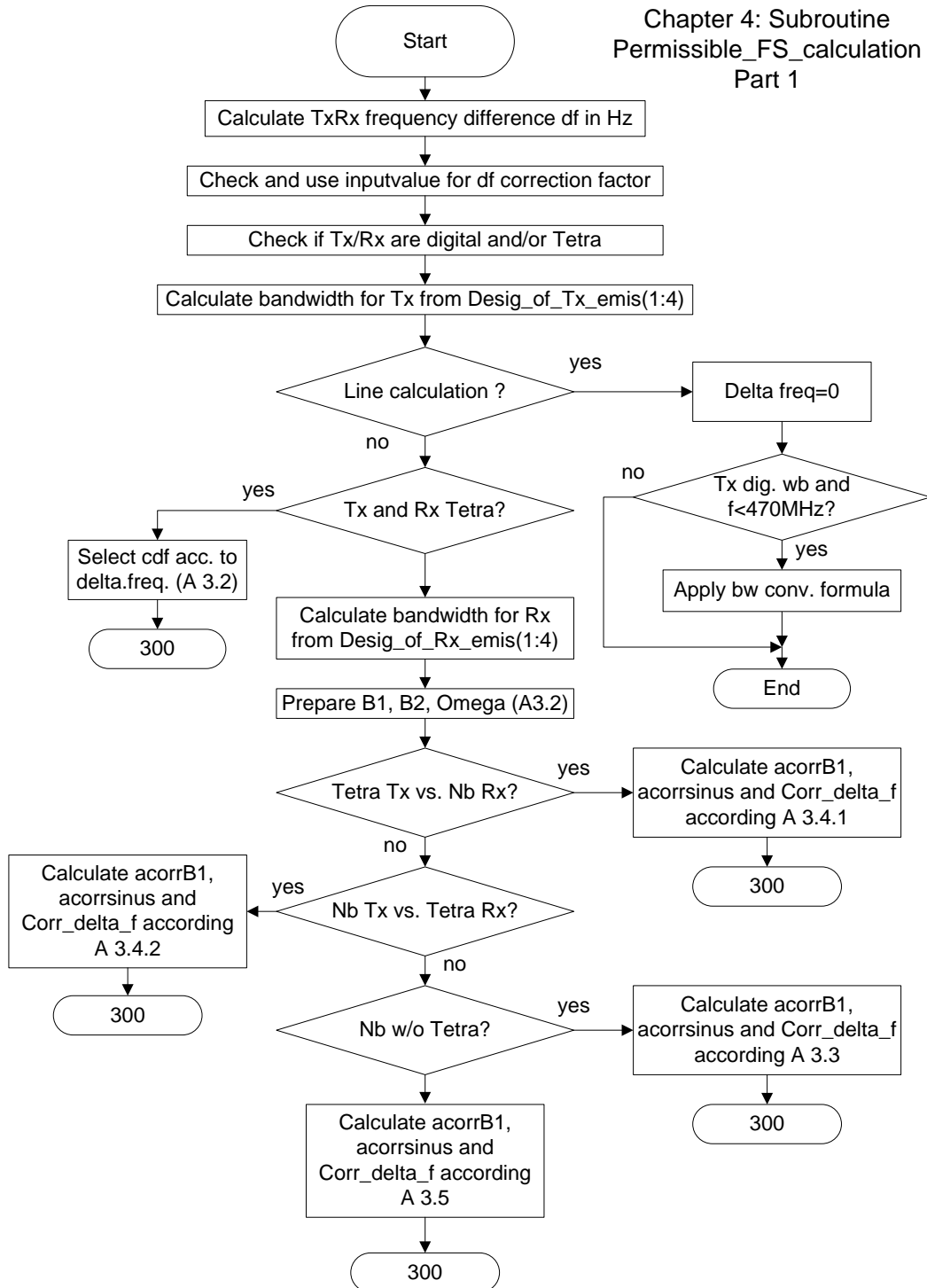
For the whole process of line calculations, two lists of record numbers and field strengths are available. List 1 is used for the first and third iteration process. List 2 is used for the second iteration process.

#### **Is number of stored items < 3?**

If the number of stored items is less than 3, then the number of stored items is increased and the record number of the co-ordinates of the new calculated point and the calculated field strength are stored in this list place.

If the number of stored records is 3 (all list places are occupied), then the entry with the lowest field strength is found and if the field strength is greater than the stored field strength this entry is overwritten with the new values.

PART1



In this subroutine the permissible field strength (either supplied or default from table 1) is corrected according to the frequency difference, the Rx antenna diagram and antenna gain and the depolarization loss.

#### **Calculate TxRx the frequency difference in Hz**

The absolute difference between Tx and Rx frequency is calculated in Hz.

#### **Is an input value of correction factor according to delta-f available?**

If an input value of correction factor according to delta-f is available, this value is read and used and Info(14) value is set to indicate this situation.

#### **Check if Tx/Rx are digital and/or Tetra Calculate bandwidth for Tx**

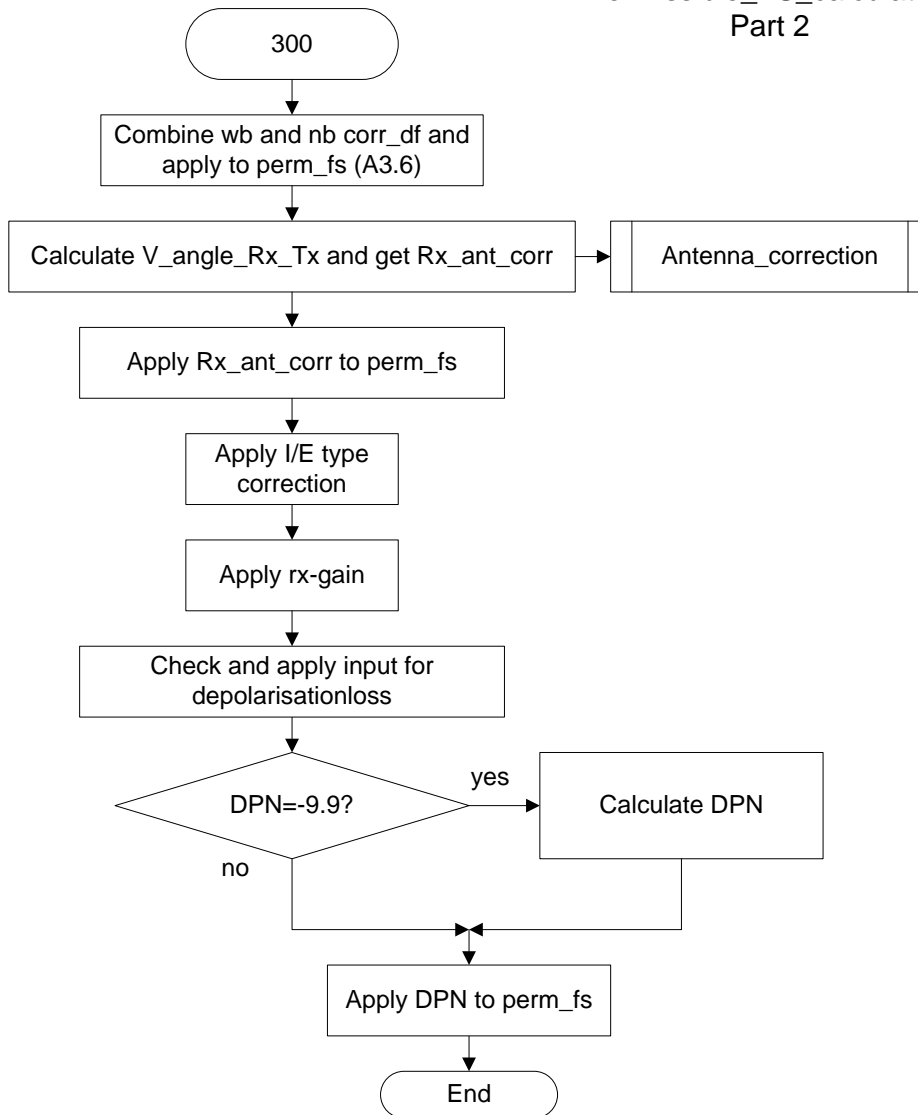
The type of involved stations (digital, TETRA) is determined and numerical bandwidth is determined.

#### **Line calculation?**

If it is linecalculation C\_mode, then after possible bw conversion (for dig. wb Tx below 470 MHz) everything is done.

#### **Calculate acorrB1, acorr sinus and Corr\_delta\_f according A X.Y.**

According to the different combination of involved stations the necessary parameters (acorrB1, acorr sinus and Corr\_delta\_f) for approximating the correction factors according to delta-f are prepared for wb and nb



@'15

### **Combine wb and nb corr\_df and apply to perm\_fs**

The wb and nb contribution to the corr\_delta\_f are summed up according Annex 3.6.

### **Calculate the Rx antenna correction**

This process is done with the subroutine Antenna\_correction. This subroutine is described in chapter 5.7.

### **Apply correction factors for Rx antenna, E/I-type, Rx-antenna gain, DPN**

The mentioned factors are applied to perm\_fs

---

## CHAPTER 5: COMMON SUBROUTINES

---

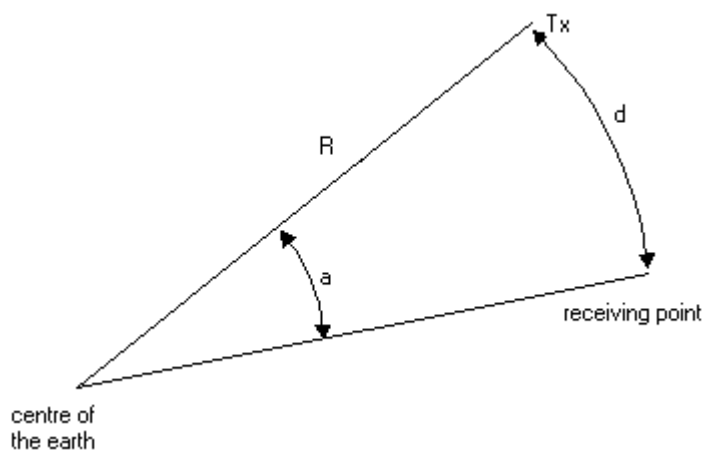
### CHAPTER 5.1: SUBROUTINE CALC\_DISTANCE

---

This routine calculates the distance from point A to point B.

#### Calculate distance

Point 1 (Tx)	longitude=A latitude=B	d=distance a=arc-distance
Point 2(receiving point)	longitude=C latitude=D	



Radius of the earth at the equator:  
6378.137 km

Radius at the poles:  
6356.752 km

Weighted radius of the earth R:  
6371.0 km

Arc-distance on the great circle:

$$a = \arccos(\sin B * \sin D + \cos B * \cos D * \cos(C - A))$$

Weighted length of one degree of a great circle on the earth surface:

$$\frac{2R\pi}{360^\circ} = 111.2 \text{ km}$$

Distance on the surface:

$$d[\text{km}] = 111.2 * a$$

n.b: since term  $\cos(C-A)$  tends to get very small in our application, leading to bad resolution for distance values, this general formula is reformulated by algebraic conversion avoiding such terms using  $\text{atan2}$ .

This subroutine calculates the azimuth from point A to point B.

### Calculate the azimuth

The azimuth is calculated according to the formula

$$azimuth = \arccos\left(\frac{\sin(LAB) - \sin(LAA) * \cos(D)}{\cos(LAA) * \sin(D)}\right)$$

where

$$D = \arccos(\sin(LAA) * \sin(LAB) + \cos(LAA) * \cos(LAB) * \cos(LOB - LOA))$$

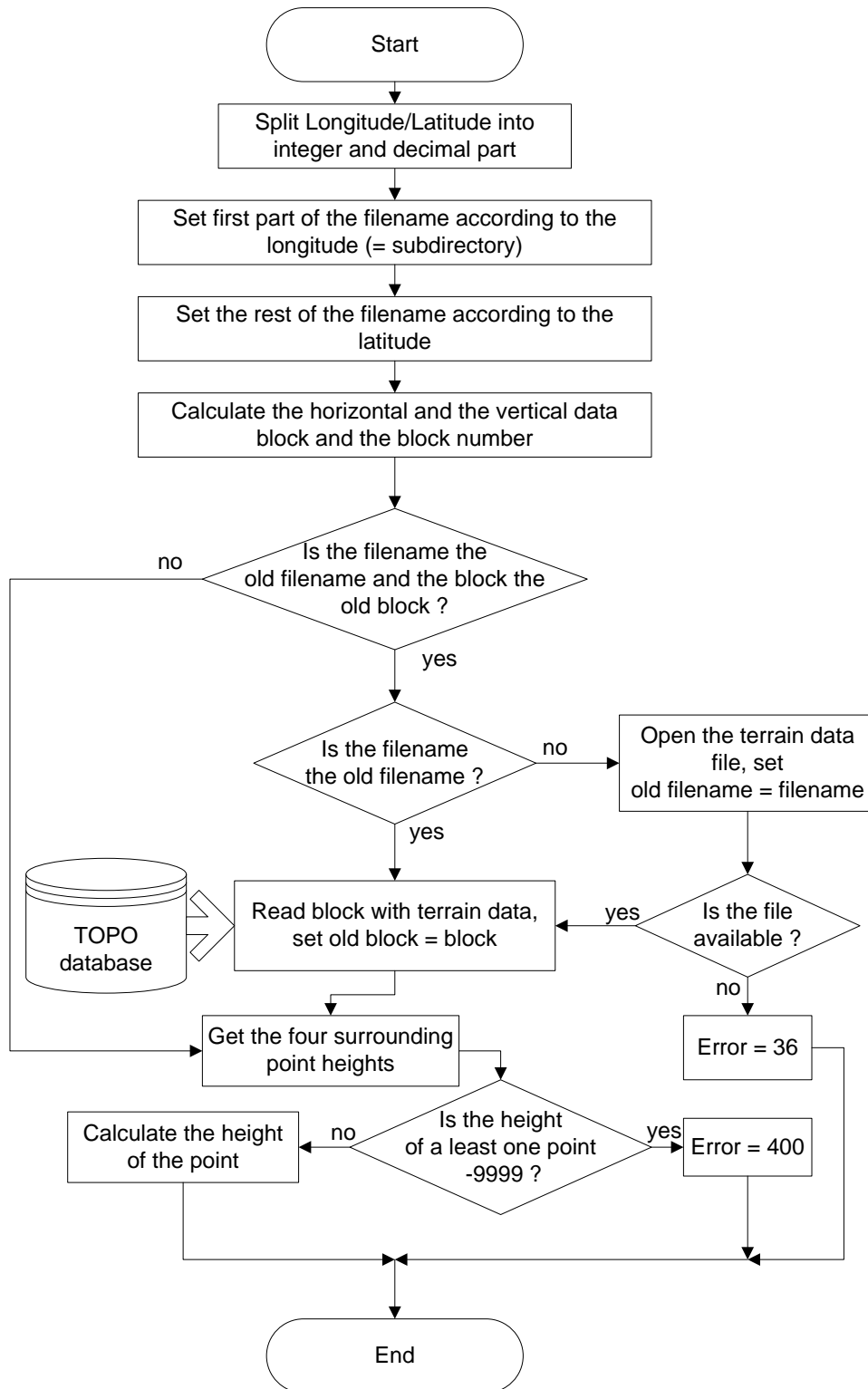
and

LOA = longitude of point A  
LAA = latitude of point A  
LOB = longitude of point B  
LAB = latitude of point B

n.b: since term  $\cos(LOB-LOA)$  tends to get very small in our application, leading to bad resolution for direction values, this general formula is reformulated by algebraic conversion avoiding such terms using  $\text{atan2}$ .



Chapter 5.3: Subroutine Point\_height



@'15

This process determines the height of a point. The height information is stored in files on a disk. The structure of this files containing the height information is described in detail in Chapter 6.

### **Split Longitude and Latitude into integer and decimal part**

After normalizing coordinates to  $0^\circ \dots 359.9^\circ$  /  $-90^\circ \dots +90^\circ$  coordinates are split into degree and remainder part for further building the filename, calculating the row and column number of the datablock in the file.

### **Set first part of the filename according to the longitude (= subdirectory)**

The first part of the filename is set using the longitude and orientation of the point (e.g. 'E015' for 15 degrees East).

### **Set the rest of the filename according to the latitude**

The rest of the filename is set using the latitude and orientation of the point (e.g. 'N52' for 52 degrees North).

The ending of the filename is set depending on the latitude. Above  $50^\circ$  latitude (N and S), the resolution of the data in East – West direction is 6 seconds, below  $50^\circ$  latitude it is 3 seconds. Therefore the ending of the filename is set to e.g. '.33E' or '.63E'

### **Calculate the horizontal and the vertical data block and the block number**

The row and column number of the datablock in the file and the position of the requested height entry in the block are determined using the previously calculated information..

### **Check if block was previously read or read from file**

If the block is already in memory use it, if not read from file to memory. Remember blocknumber for next request.

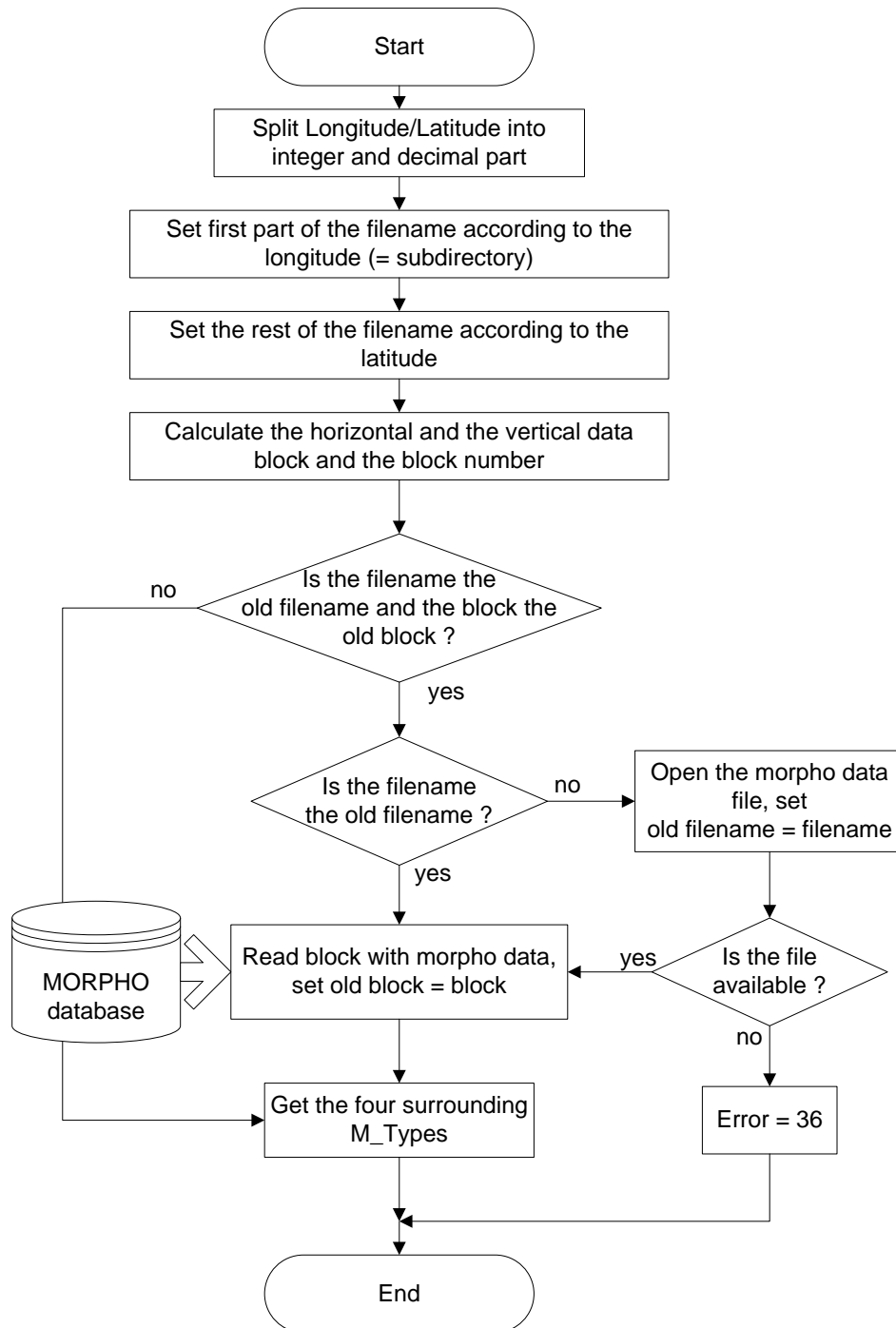
### **Get the four surrounding point heights**

The heights of the 4 surrounding grid points are determined.

### **Calculate the height of the point**

The resulting height is calculated by interpolating between the 4 surrounding points.

Chapter 5.4: Subroutine Point\_type



@'15

This process determines the morphological information of a point. The information is stored in files on a disk. The structure of these files containing the morphological information is described in detail in Chapter 7.

### **Split Longitude and Latitude into integer and decimal part**

After normalizing coordinates to  $0^\circ \dots 359.9^\circ$  /  $-90^\circ \dots +90^\circ$  coordinates are split into degree and remainder part for further building the filename, calculating the row and column number of the datablock in the file.

### **Set first part of the filename according to the longitude (= subdirectory)**

The first part of the filename is set using the longitude and orientation of the point (e.g. 'E015' for 15 degrees East).

### **Set the rest of the filename according to the latitude**

The rest of the filename is set using the latitude and orientation of the point (e.g. 'N52' for 52 degrees North).

The ending of the filename is set depending on the latitude. Above  $50^\circ$  latitude (N and S), the resolution of the data in East – West direction is 6 seconds, below  $50^\circ$  latitude it is 3 seconds. Therefore the ending of the filename is set to e.g. '.33M' or '.63M'

### **Calculate the horizontal and the vertical data block and the block number**

The row and column number of the datablock in the file and the position of the requested height entry in the block are determined using the previously calculated information..

### **Check if block was previously read or read from file**

If the block is already in memory use it, if not read from file to memory. Remember blocknumber for next request.

### **Get the four surrounding M-Types**

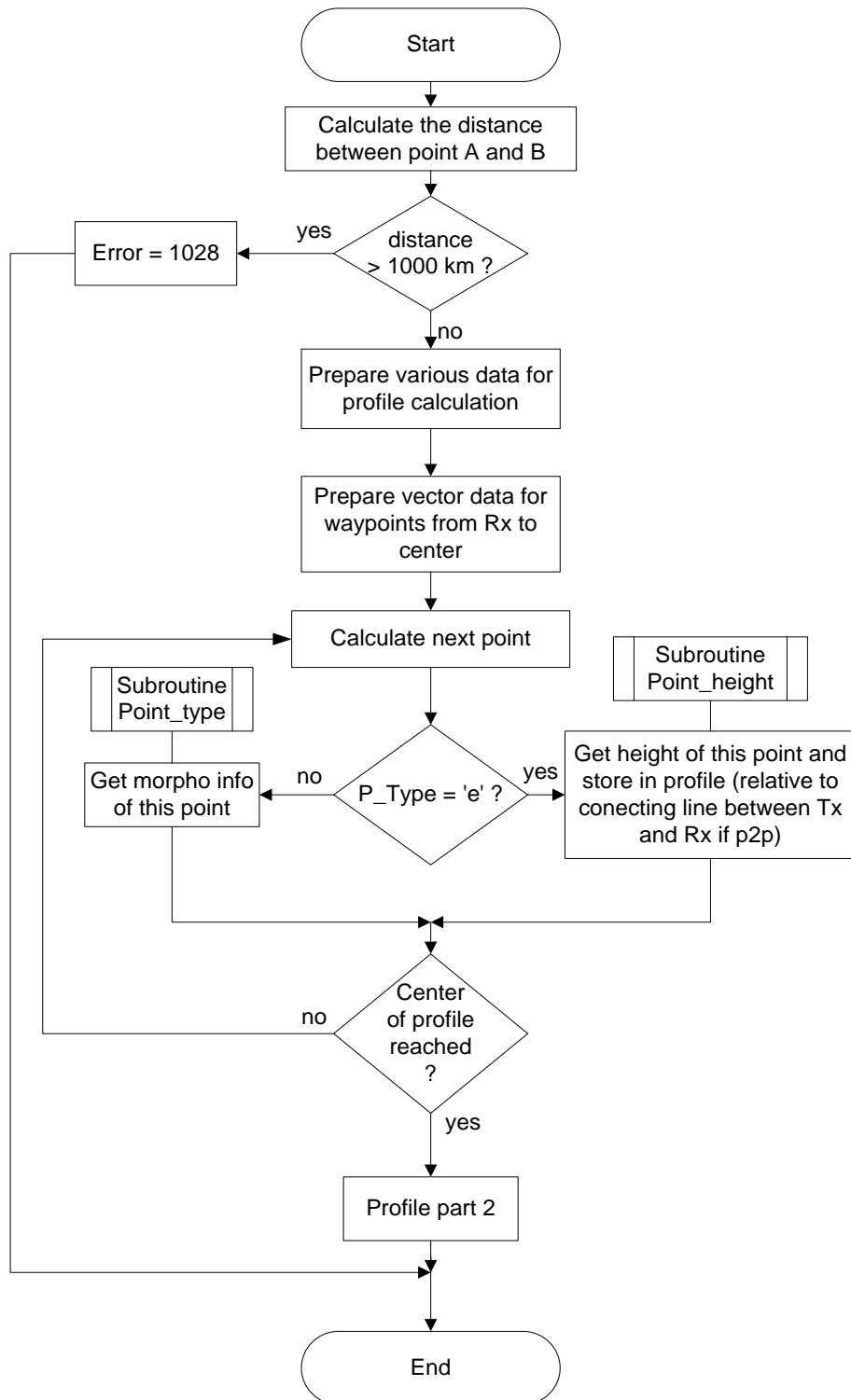
The information of the 4 surrounding grid points is determined.

### **Get M-Type of the nearest point**

The resulting morphological information is taken from the nearest point.

PART 1

Chapter 5.5: Subroutine Profile part 1



@'15

This process determines the height profile or the morphological profile between two points.

### **Calculate the distance between point A and B**

The distance is calculated using Calc\_distance. If it is more than 1000 km, the HCM\_error is set to 1028 and the subroutine terminates.

### **Prepare various data for waypoints from Rx to center**

Number of waypoints is determined. Values for slanted profile are prepared. Constant factors for spherical triangulation are calculated and the step size is transformed from direction and distance to spherical vector.

The switch to spherical trigonometry for better accuracy and splitting path into two parts (end to center) are done to get same profile not depending on direction.

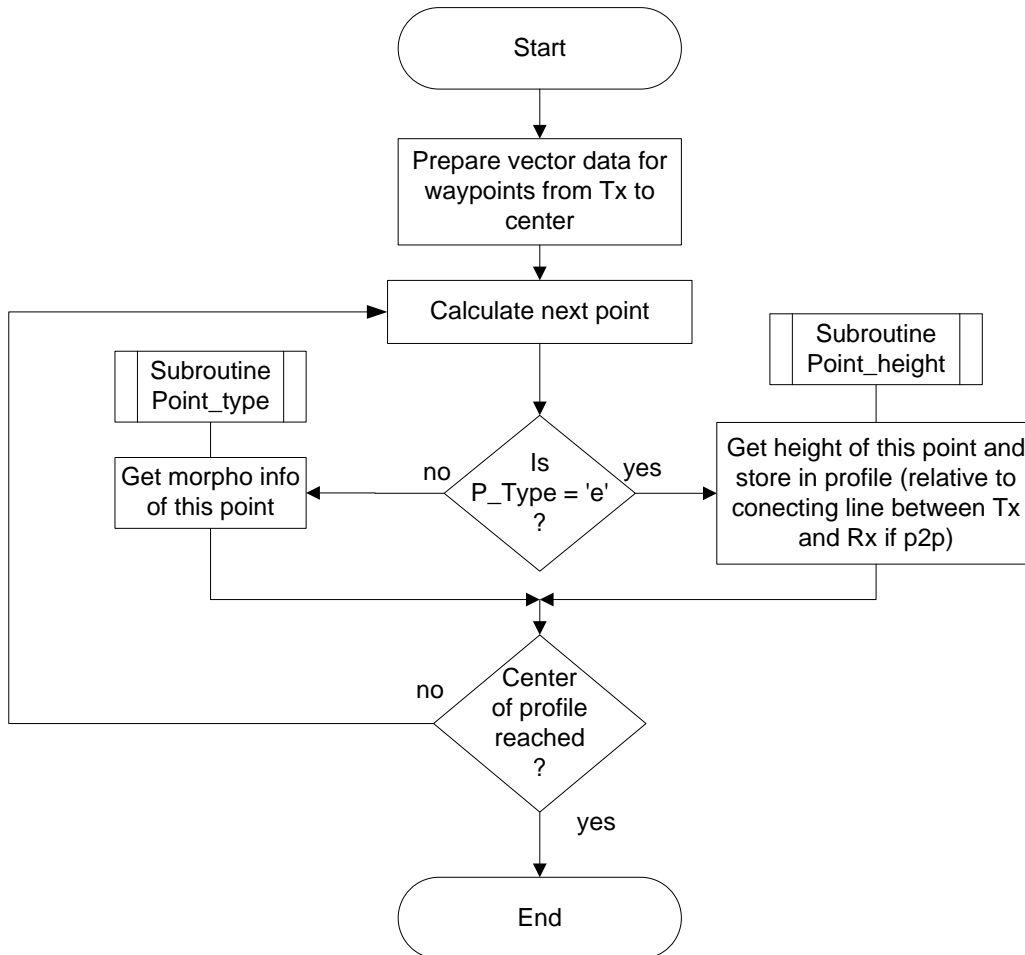
### **Calculate next point**

The geographical co-ordinates of the next point are calculated and converted from spherical to planar.

### **Get height or type of this point and store in profile (relative to connecting line between Tx and Rx if point2point)**

The terrain height or type of the point is read (from the database). If slanted profile is requested (depending on C\_mode) the difference to the height of the point on the connecting line between the Tx and Rx location is built, else the height or type is stored in the profile.

Chapter 5.5: Subroutine Profile part 2



@'15

This process determines the height profile or the morphological profile between two points.

#### **Prepare various data for waypoints from Tx to center**

Constant factors for spherical triangulation are calculated and the step size is transformed from direction and distance to spherical vector, now with reversed direction.

#### **Calculate next point**

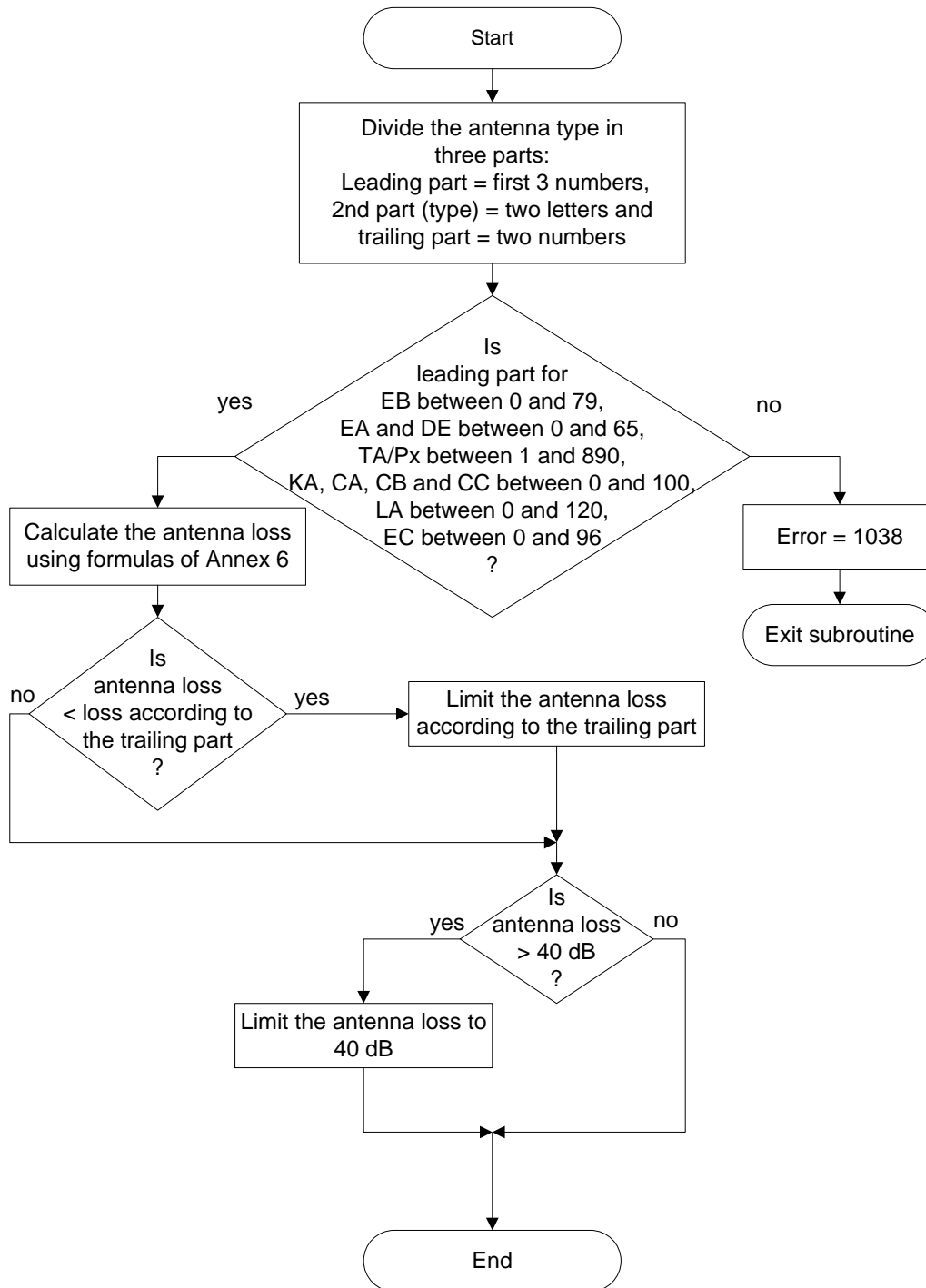
The geographical co-ordinates of the next point are calculated and transformed from spherical to planar.

#### **Get height or type of this point and store in profile relative to connecting line between Tx and Rx**

The terrain height or type of the point is read (from the database). If slanted profile is requested the difference to the height of the point on the connecting line between the Tx and Rx location is built, else the height or type is stored in the profile.



Chapter 5.6: Subroutine Antenna



@'15

This process determines the loss of an antenna according to the radiation pattern in a given direction.

### Divide the antenna type in three parts

The first part of this subroutine splits the input antenna character string into three parts as described in point 1.1 of Annex 6 of the HCM Agreement:

- the leading part (first 3 numbers)
- the type (2 letters)
- the trailing part (last 2 numbers)

**Is leading part for EB between 0 and 79, EA and DE between 0 and 65, TA/Px between 1 and 890, KA, CA, CB and CC between 0 and 100, LA between 0 and 120, EC between 0 and 96?**

This part of the subroutine checks if the leading part is in line with the antenna types limits as defined in Annex 6.

type	Valid values for leading part
EA	001 - 065
EB	001 - 079
EC	001 - 096
DE	001 - 065
TA/Px	001 - 890
LA	001 - 120
KA	000 - 100
CA	000 - 100
CB	000 - 100
CC	000 - 100

For other types no validation check is performed.

### Calculate the antenna loss using formulas of Annex 6

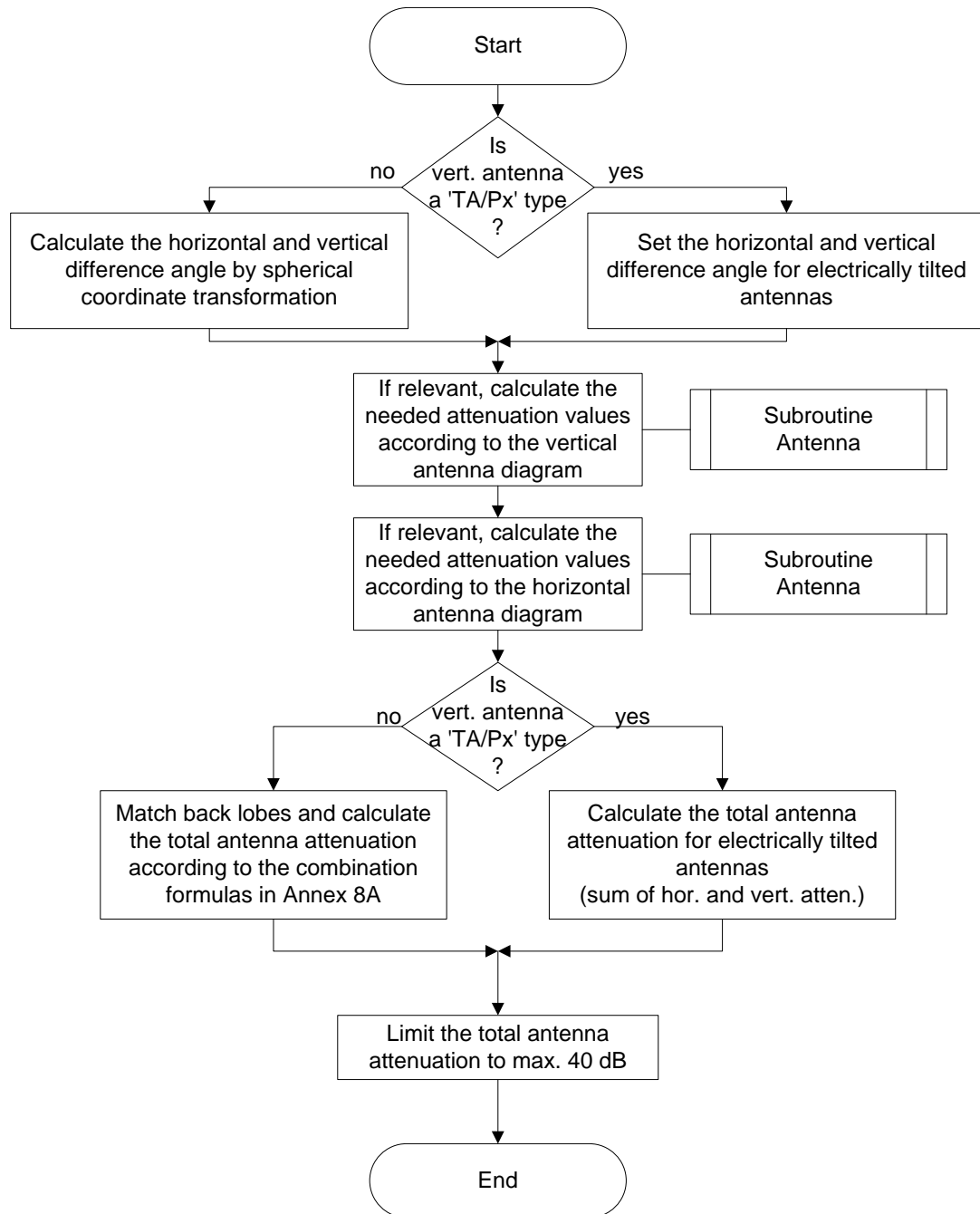
For EA, EB, EC, DE, TA/Px, LA, KA, CA, CB and CC type antennas, the angle and antenna codes are used directly with the formulas in Annex 6 of the HCM Agreement to calculate the loss.

For V\* and W\* type antennas, the antenna code is further analyzed to derive the correct formula according to Appendix 4 & 5 to Annex 6 of the HCM Agreement.

For ND type antennas the loss is set to 0 dB.

The result is limited to -40 dB, giving an antenna loss in the range of -40 ... 0 dB.

Chapter 5.7: Subroutine Antenna\_Correction

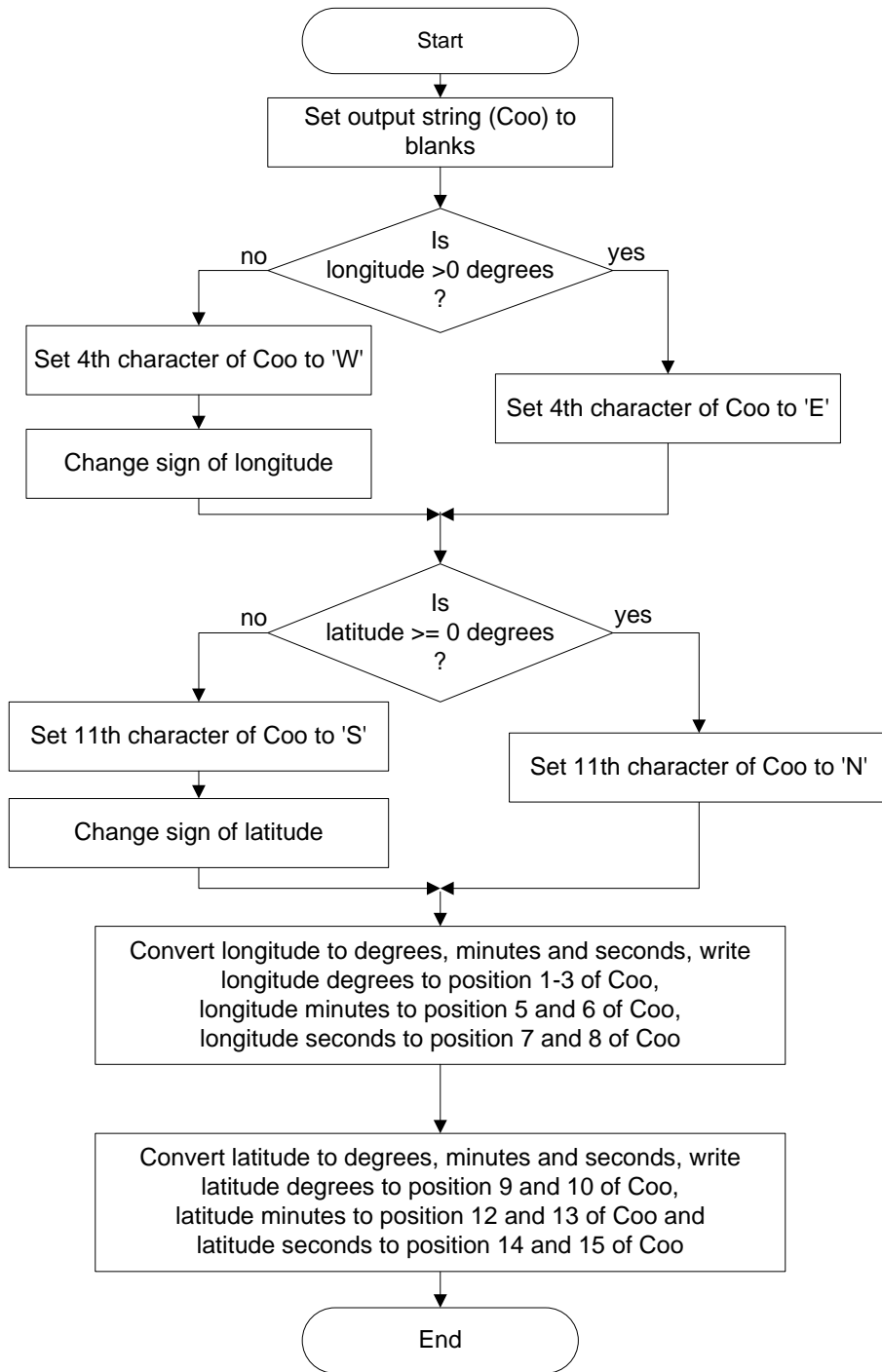


@'15

This process determines the total loss of an antenna in a given three-dimensional direction as fully described in the Annex 8A of the HCM Agreement.

For Px antennas the “x” (valid range “A” .. “Z”) is converted into a tilt value from 0° to 25°.

Chapter 5.8: Subroutine CooConv



This process converts co-ordinates from decimal to text format.

The output is in the format '111E223344N5500'

where

'111' is the longitude degrees

'E' is the indicator for East, for West is 'W'

'22' is the longitude minutes

'33' is the longitude seconds

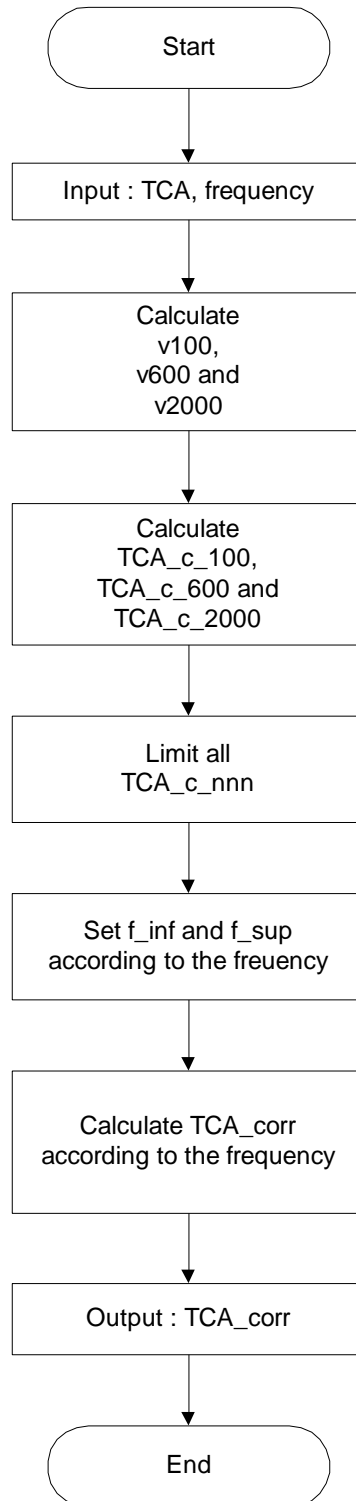
'44' is the latitude degrees

'N' is the indicator for North, for South it is 'S'

'55' is the latitude minutes

'00' is the latitude seconds

Chapter 5.9: Subroutine TCA\_correction\_calculation



@'15

This process calculates the correction factor according to the terrain clearance angle (TCA) according to Annex 5.

### Calculate $v_{100}$ , $v_{600}$ and $v_{2000}$

$$v_{100} = 0.649 * TCA$$

$$v_{600} = 1.592 * TCA$$

$$v_{2000} = 2.915 * TCA$$

### Calculate $TCA_{c\_100}$ , $TCA_{c\_600}$ and $TCA_{c\_2000}$

$$TCA_{c\_100} = 9.1 - (6.9 + 20 * \log(\sqrt{((v_{100} - 0.1)^2 + 1)} + v_{100} - 0.1))$$

$$TCA_{c\_600} = 13.1 - (6.9 + 20 * \log(\sqrt{((v_{600} - 0.1)^2 + 1)} + v_{600} - 0.1))$$

$$TCA_{c\_2000} = 17.3 - (6.9 + 20 * \log(\sqrt{((v_{2000} - 0.1)^2 + 1)} + v_{2000} - 0.1))$$

### Limit $TCA_{c\_nnn}$

Limits:       $TCA_{c\_100}$       = range from 0 to - 32  
                  $TCA_{c\_600}$       = range from 0 to - 35  
                  $TCA_{c\_2000}$     = range from 0 to - 36

### Set $f_{inf}$ and $f_{sup}$ according to the frequency

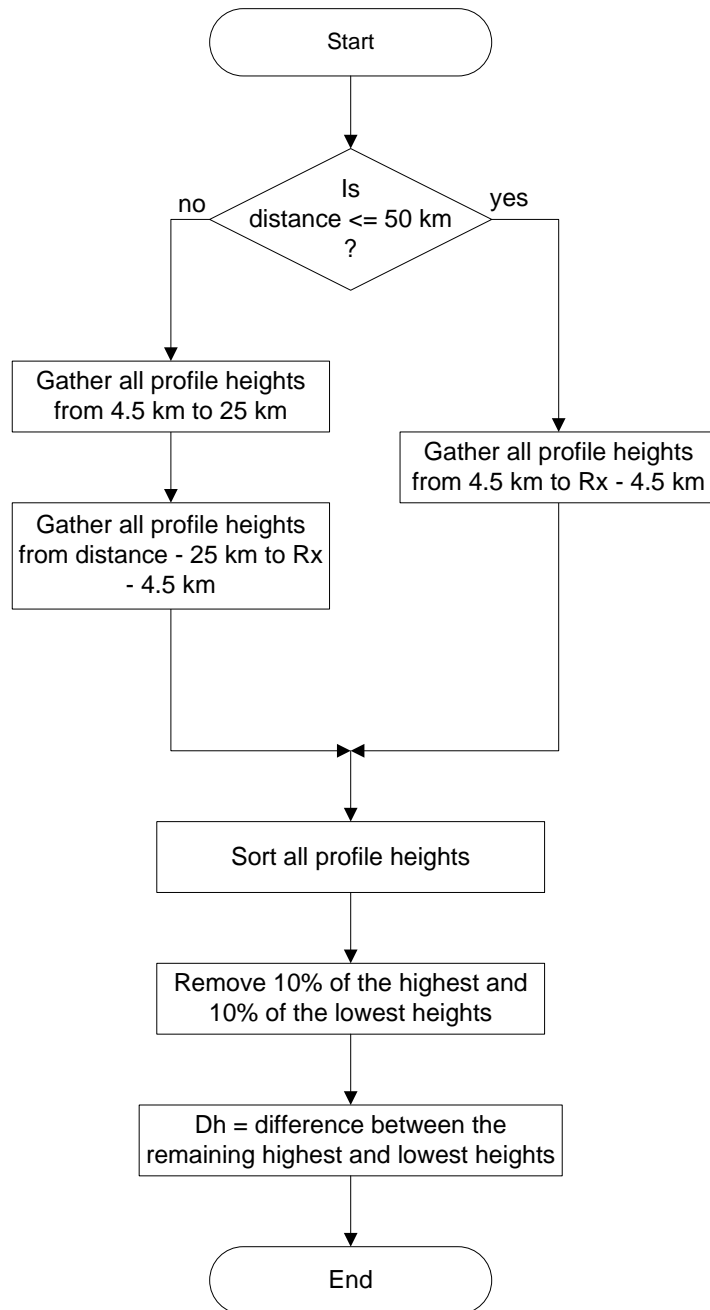
The nominal frequencies  $f_{inf}$  and  $f_{sup}$  (100, 600 or 2000 MHz) are set according to the frequency.

### Calculate $TCA_{corr}$ according to the frequency, $f_{inf}$ and $f_{sup}$

The total correction factor  $TCA_{corr}$  is calculated using linear interpolation.



Chapter 5.10: Subroutine Dh\_calculation



@'15

This process calculates the terrain irregularity  $Dh$ .

### **Is distance $\leq 50$ km?**

If the distance is less than or equal to 50 km, then the profile of the total distance minus two times 4.5 km (at the beginning and at the end) is determined, else two parts of the profile are taken, first at the Tx site from 4.5 to 20 km and second at the Rx site 4.5 km from the Rx to 25 km from the Rx (in direction to the Tx).

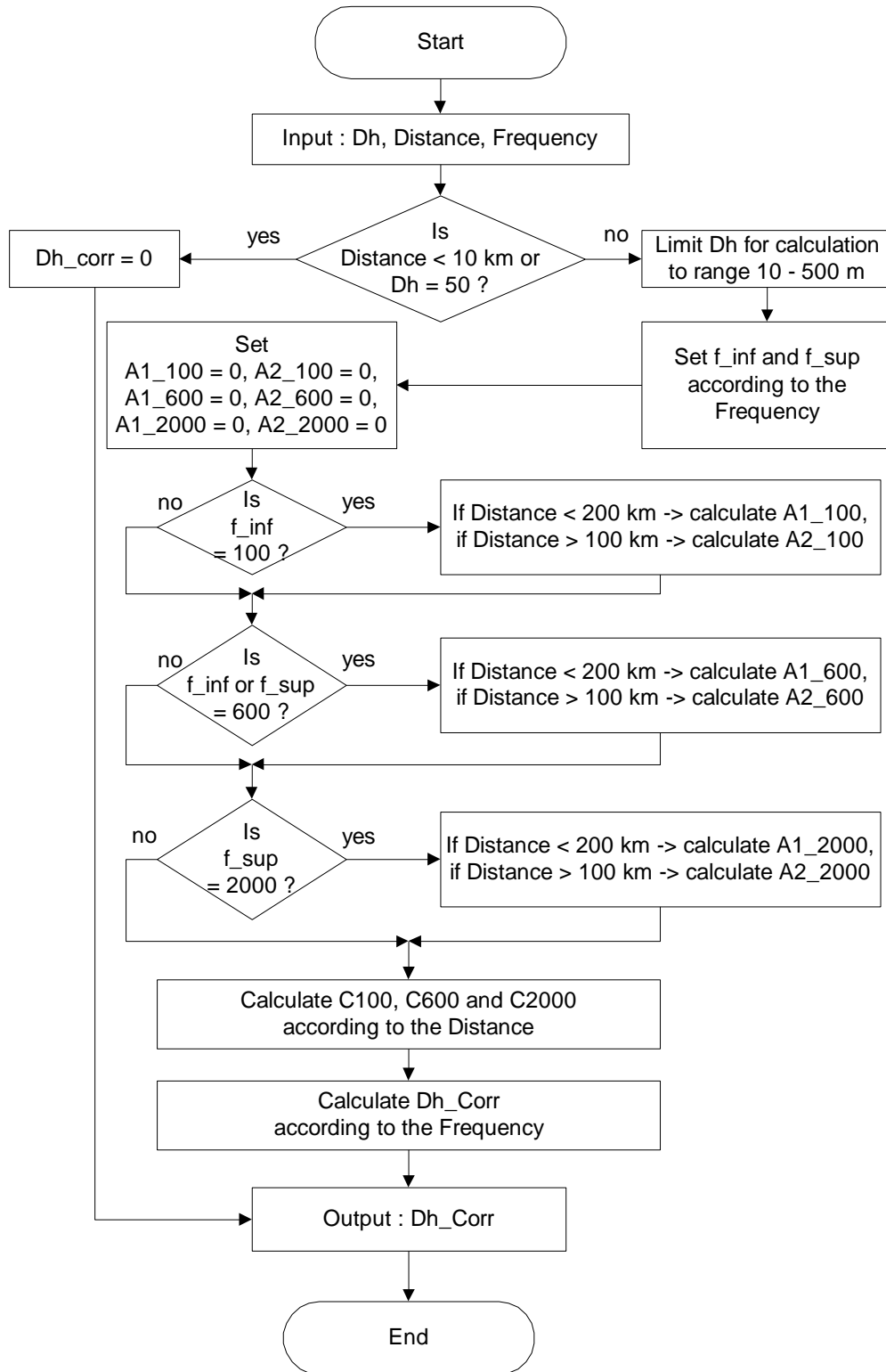
### **Sort all profile heights**

All taken heights are sorted.

### **Remove 10% of the highest and 10% of the lowest heights**

If e.g. 100 heights are taken and sorted, the 10 (=10%) first and 10 last heights are removed and the remaining difference between the lowest and highest height is the terrain irregularity  $Dh$  ( $Dh = h_{90} - h_{11}$ ).

Chapter 5.11: Subroutine Dh\_Correction



@'15

This process calculates the correction factor according to the terrain irregularity  $D_h$ .

#### **Limit $D_h$ for calculation to range 10 – 500 m**

Only for the calculation process,  $D_h$  is limited to the range 10 – 500 m.

#### **Set $f_{inf}$ and $f_{sup}$ according to the Frequency**

The values of  $f_{inf}$  and  $f_{sup}$  are set to one of the nominal frequencies 100, 600 or 2000 MHz according to the Frequency.

According to the values of  $f_{inf}$  and  $f_{sup}$  and the distance, the intermediate values of  $A1_{100}$ ,  $A2_{100}$ ,  $A1_{600}$ ,  $A2_{600}$ ,  $A1_{2000}$  and  $A2_{2000}$  are calculated.

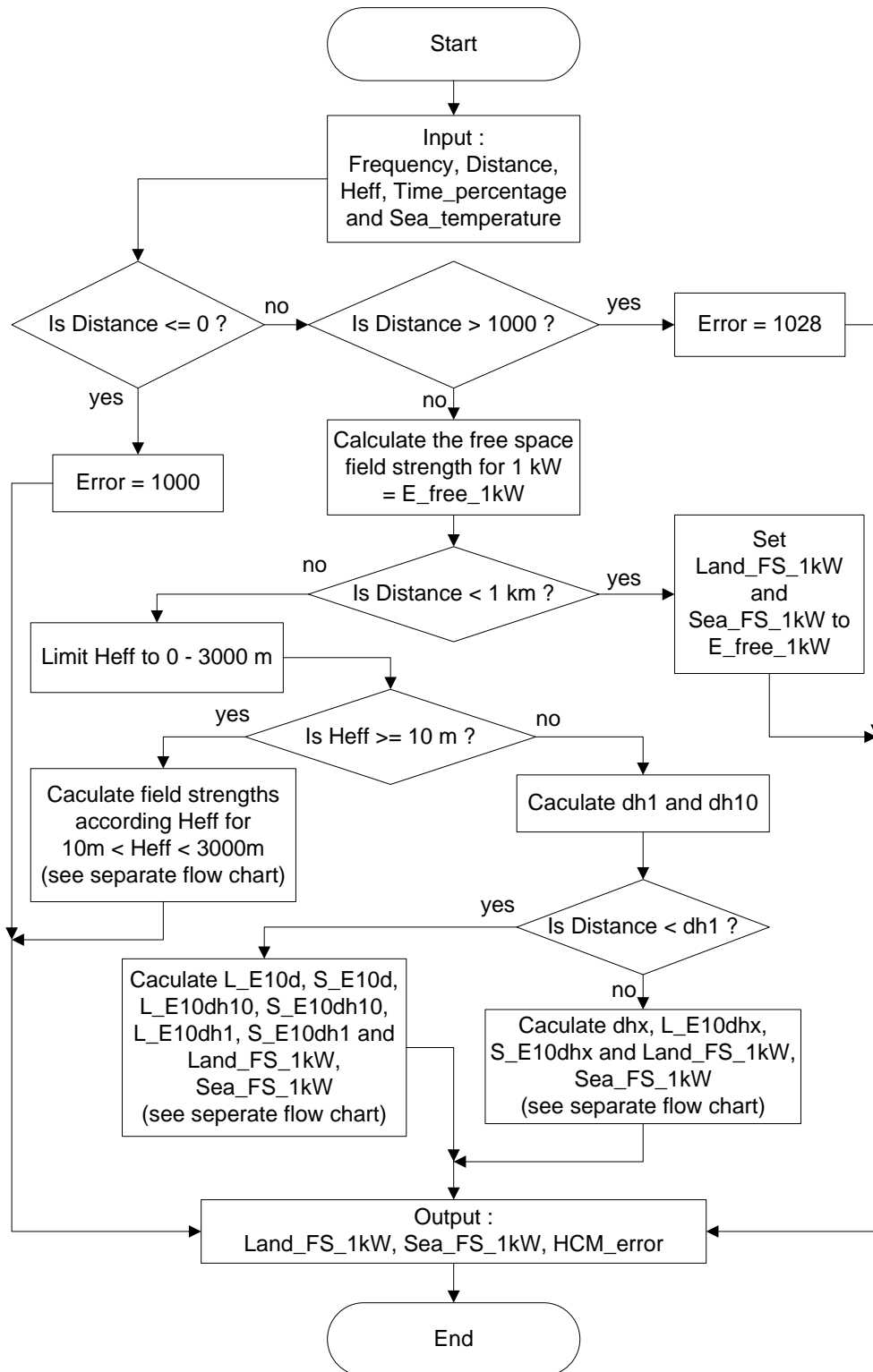
#### **Calculate $C100$ , $C600$ and $C2000$ according to the Distance**

The correction factors for 100, 600 and 2000 MHz ( $C100$ ,  $C600$  and  $C2000$ ) are calculated for the given distance.

#### **Calculate $D_h_{Corr}$ according to the Frequency**

Finally, the correction factor according to the terrain irregularity is calculated for the given frequency.

Chapter 5.12: Subroutine Get\_FS\_from\_figures



@'15

This process calculates the land- and sea field strength for 1 kW from the figures.

In a first step, the distance is checked if it is greater than 0 and maximum 1000 km. If not, the respective error codes are set and the process terminates.

### **Calculate the free Space field strength for 1 kW = E\_free\_1kW**

The free space field strength for 1 kW is calculated. This value is required for further steps.

### **Is Distance < 1 km?**

If the distance is less than 1 km, then the land- and sea field strength for 1 kW are set to the free space field strength for 1 kW.

### **Calculate field strength according Heff for 10 m < Heff < 3000 m (see separate flow chart)**

This process calculates the 1 kW field strength for Heff greater than or equal to 10 m and is described in chapter 5.12.1.

### **Calculate dh1 and dh10**

The distances dh1 and dh10 are calculated using formulas of Appendix 2 to Annex 5, chapter 1.2.

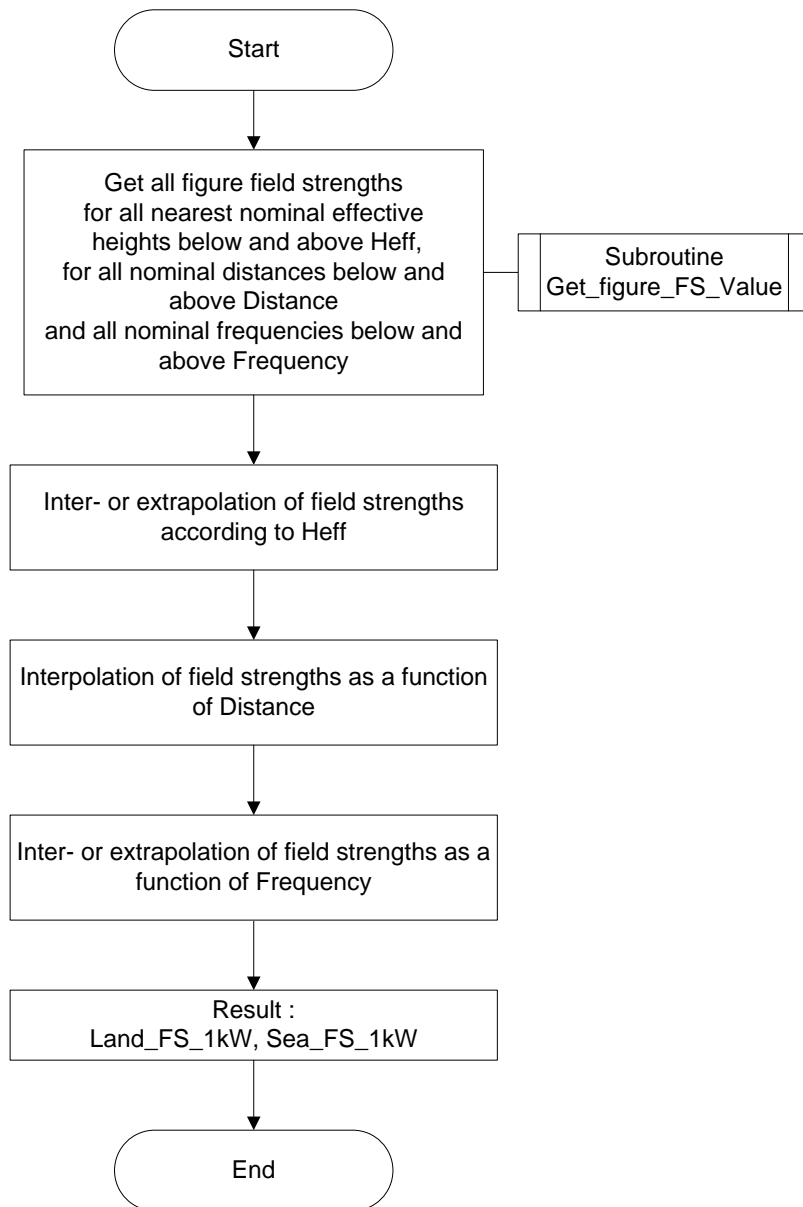
### **Calculate L\_E10d, S\_E10d, L\_Edh10, S\_Edh10, L\_E10dh1, S\_E10dh1 and Land\_FS\_1kW, Sea\_FS\_1kW (see separate flow chart)**

If the distance is less than dh1, then the land- and sea field strengths for 1 kW are calculated using this process. This process is described in chapter 5.12.2.

### **Calculate dhx, L\_E10dhx, S\_E10dhx and Land\_FS\_1kW, Sea\_FS\_1kW (see separate flow chart)**

If the distance is equal or greater than dh1, then the land- and sea field strengths for 1 kW are calculated using this process. This process is described in chapter 5.12.3.

Chapter 5.12.1: Caculate field strengths according Heff for 10m < Heff < 3000m



This process calculates the land- and sea field strength for 1 kW from the figures if Heff is between 10 and 3000 m.

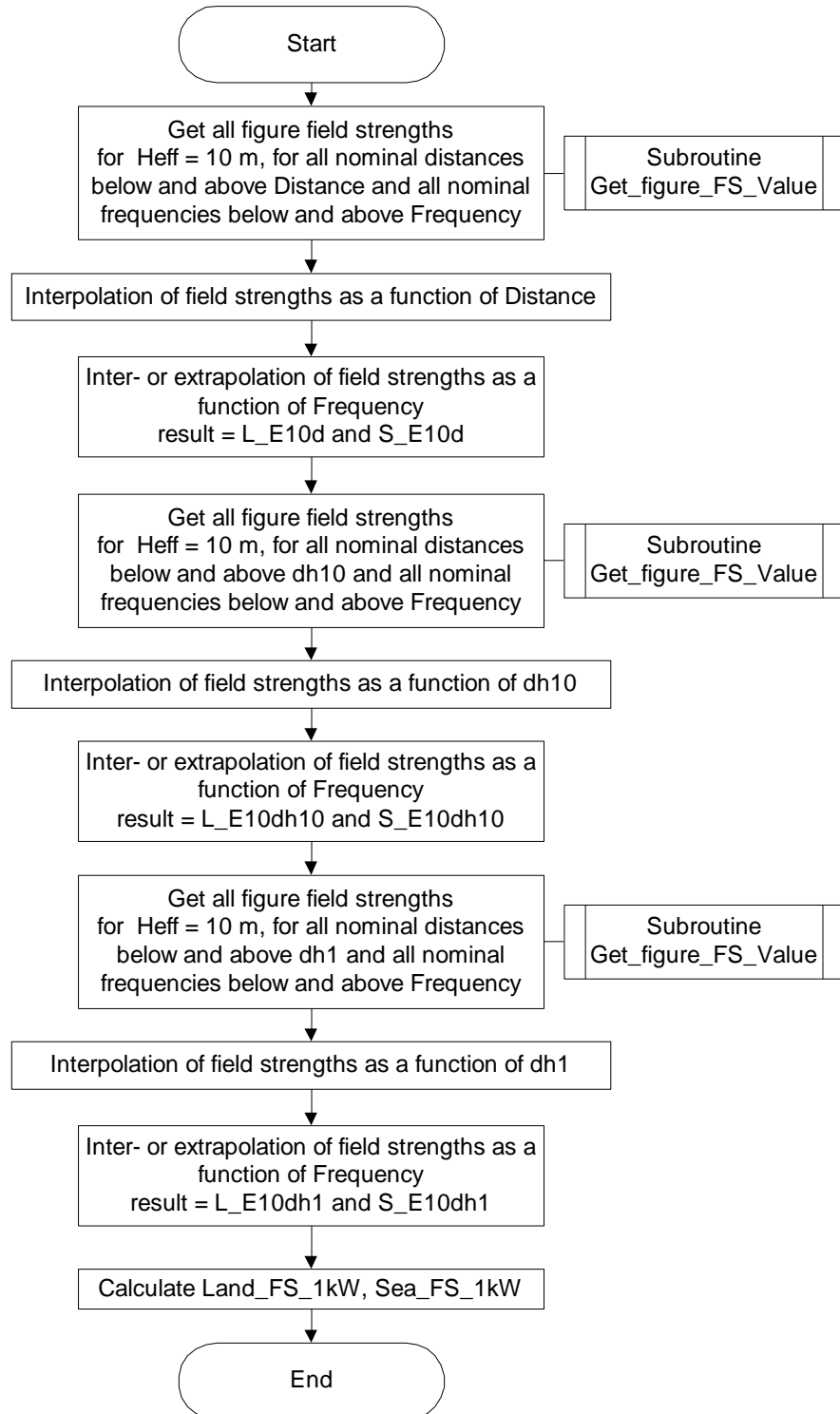
### **Subroutine Get\_figure\_FS\_Value**

This process determines the figure values and is described in chapter 5.13.

All the values are interpolated using formulas of Annex 5. The result is the land- and sea field strength for 1 kW.



Chapter 5.12.2: Caculate L\_E10d, S\_E10d, L\_E10dh10, S\_E10dh10, L\_E10dh1, S\_E10dh1 and Land\_FS\_1kW, Sea\_FS\_1kW



@'15

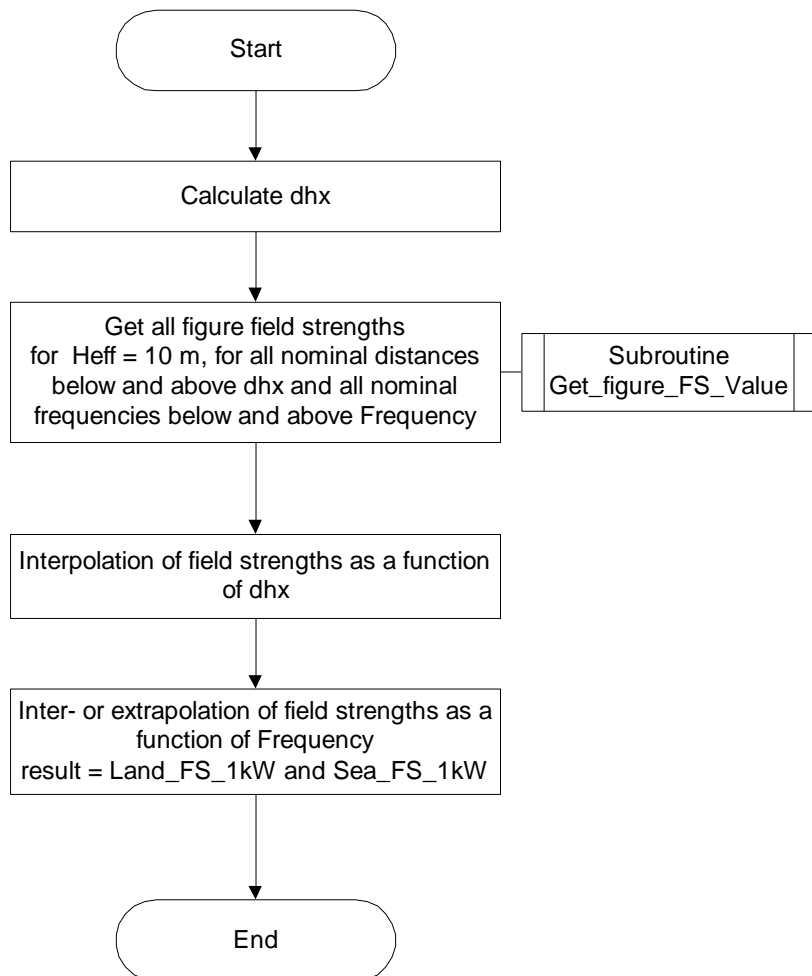
This process calculates the land- and sea field strength for 1 kW from the figures if  $H_{eff}$  is less than 10 m and the distance is less than  $dh_1$ .

### **Subroutine Get\_figure\_FS\_Value**

This process determines the figure values and is described in chapter 5.13.

The whole calculation is based on the formulas of Appendix 2 to Annex 5, chapters 1.2, 2 and 3. The outcome is the land- and sea field strength for 1 kW.

Chapter 5.12.3: Caculate dhx, L\_E10dhx, S\_E10dhx  
and Land\_FS\_1kW, Sea\_FS\_1kW



@'15

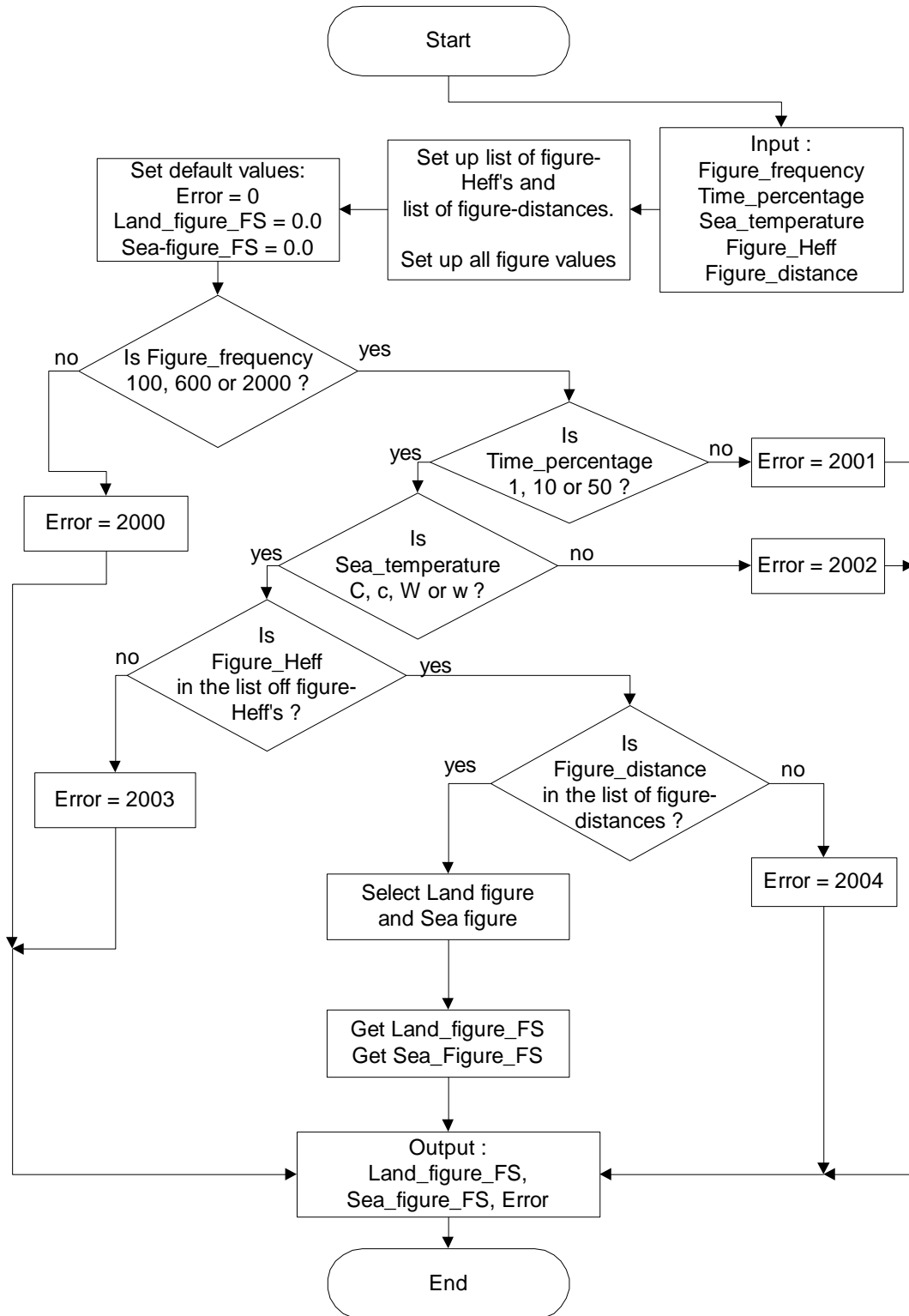
This process calculates the land- and sea field strength for 1 kW from the figures if  $H_{eff}$  is less than 10 m and the distance is equal or greater than  $dh_1$ .

### **Subroutine Get\_figure\_FS\_Value**

This process determines the figure values and is described in chapter 5.13.

The whole calculation is based on the formulas of Appendix 2 to Annex 5, chapters 1.2, 2 and 3. The outcome is the land- and sea field strength for 1 kW.

Chapter 5.13: Subroutine Get\_figure\_FS\_value



@'15

This process gets one figure value for the land field strength and one for the sea field strength from the field strength curves.

The field strength values are stored for the following distances (km):

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,  
16, 17, 18, 19, 20, 25, 30, 35, 40, 45, 50, 55, 60,  
65, 70, 75, 80, 85, 90, 95, 100, 110, 120, 130, 140,  
150, 160, 170, 180, 190, 200, 225, 250, 275, 300,  
325, 350, 375, 400, 425, 450, 475, 500, 525, 550,  
575, 600, 625, 650, 675, 700, 725, 750, 775, 800,  
825, 850, 875, 900, 925, 950, 975, 1000

For each distance, the field strength values for the following effective antenna heights are stored:

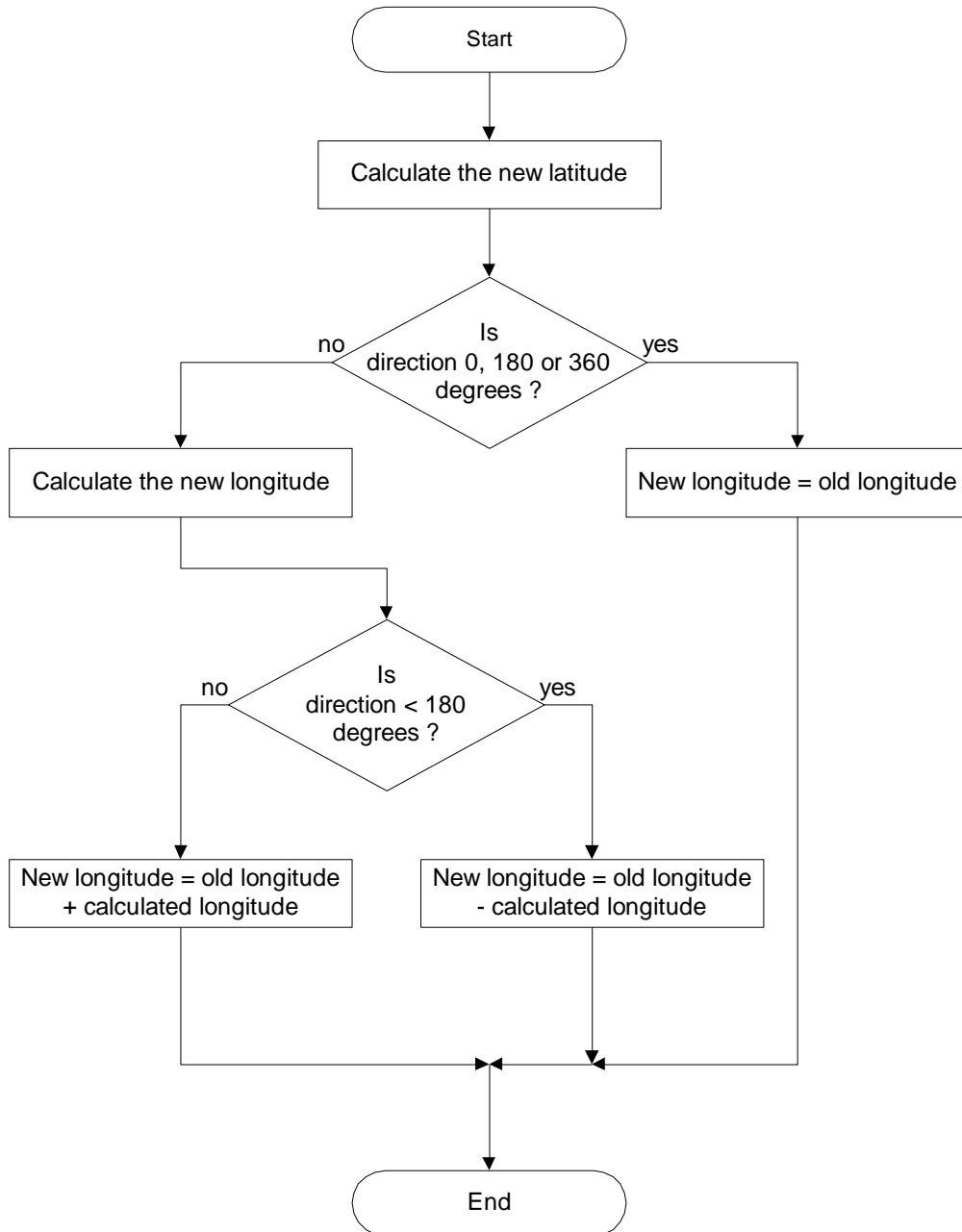
10, 20, 37.5, 75, 150, 300, 600, 1200 m and the maximum field strength (for 1 kW).

All this values are stored for the frequencies 100, 600 and 2000 MHz and for time percentages 1, 10 and 50%.

For the sea field strength there are two sets of data, one for cold sea and one for warm sea.

In this process, the selection of the required data elements is done taking into account the frequency, the time percentage, the sea temperature, the effective antenna height and the distance.

Chapter 5.14: Subroutine New\_coordinates



@'15

This process calculates the new co-ordinates starting from the co-ordinates of a known point going into the known direction with a known distance.



---

## CHAPTER 6: STORAGE FORMAT OF THE HEIGHT DATA

---

The height data used by the HCM programs (Mobile Service and Fixed Service) may come from different sources.

Some data may come from military source (Digital Terrain Elevation Data (DTED) Level 1 Coverage); some data may come from national geographic institutes.

A fallback possibility to get terrain height data is to use GTOPO30 data. This data is free available at the Internet, but it only has a resolution of 30 seconds (approximately 1 km). GTOPO30 data converted to the HCM format are available at the Internet site of the HCM Agreement (<http://hcm.bundesnetzagentur.de>).

A second fallback possibility is to use the SRTM3 topo data, converted to HCM format. This data has a resolution of 3 seconds by 3 seconds but is only available for latitudes from 61S to 61N. This data is measured from the space and therefore all heights are not the heights above sea level, but the heights include the morphology (e.g. heights of buildings, forests, ...). This data is also available on the Internet site of the HCM Agreement.

The HCM terrain height data has a resolution of 3 seconds in the North - South direction and 3 or 6 seconds in the East - West direction (depending on the latitude). If the latitude is less than 50 degrees (North or South), the resolution is 3 seconds, if the latitude is greater or equal to 50 degrees, the resolution is 6 seconds.

All source data has to be converted to WGS84 format and to the above-mentioned resolution. Data from different sources needs to be combined to a common database covering all HCM Agreement Signatory countries (plus an additional range of approximately 100 km).

The storage format of this data is:

- All elevation data consists of 2 bytes Integer-values (Fixed Binary integers). If the elevation value is negative, first the MSB (Bit #15) has to be cleared and second the complement has to be built to get the correct elevation value.
- Terrain data of a 5 x 5 minute square (approximately 9 x 5 km) is combined in one data-record. Strips in the North and East are added to the data inside the square. The reason is: If you want to have the elevation of a given point, in most cases this point is located between 4 points in the grid of the stored data. To get the correct elevation, you have to interpolate between these 4 points. The western and southern grid-points are always present (example: wanted point 8 degrees 0 minute 1 second → record 8 degrees 0 minute to 5 minutes is read; so the western grid-point (= 8 degrees 0 minute 0 second) is present. In the case where the wanted point is for example 8 degrees 4 minutes 59 seconds, the eastern grid-point is 8 degrees 5 minutes 0 second. Normally this point is not inside the read record, but part of another record (8 degrees 5 minutes to 10 minutes). To prevent the program from reading an additional record, the strip 5 minutes 0 second is added to the record 0 minute to 5 minutes. For the same reason, a strip in the North is also added. It is therefore possible to get the right elevation of a wanted point reading only one record of data.
- 12 x 12 records (=144 records; =1 x 1 degree) are stored in one file.

- The filename is (example): E007N50.63E

where     E007 = Longitude of the South-West corner  
           N50 = Latitude of the South-West corner  
           63 = Resolution in seconds longitude (6) and latitude (3)  
           E = Elevation data (M for morphological data)

- Position of records inside the file:

North

	133	134	135	136	137	138	139	140	141	142	143	144		
	121												132	
	109												120	
	97												108	
	85												96	
	73												84	
West	61												72	East
	49												60	
	37												48	
	25												36	
	13												24	
	1	2	3	4	5	6	7	8	9	10	11	12		

South

- All files with the same longitude are stored in the same subdirectory. The name of the subdirectory is equal to the first four characters of the filename (example: E007).
- All subdirectories are stored in the (top level) directory, e.g. 'TOPO'. A valid filename with the full path therefore is:

C:\TOPO\E010\E010N45.33E

**Note**

In older versions of the HCM software, only the drive letter 'C' may be replaced by any other valid drive letter and the top level directory is fixed to 'TOPO'. In version 7 of the HCM software, it is possible to define the path of the topo-data and only the name of the subdirectory and the file name is fixed, but the old system still works alternatively.



## CHAPTER 7: STORAGE FORMAT OF THE MORPHOLOGICAL DATA

---

The morphological data required for the HCM programs is provided by the TWG HCM SWG Program (Technical Working Group Harmonized Calculation Method Sub Working Group Program) and is available at the Internet site of the HCM Agreement (<http://hcm.bundesnetzagentur.de>).

The data is elaborated using GTOPO30 data (see Chapter 6).

The morphological database is a raster database with the same grid and structure as the terrain height database (see Chapter 6).

Each entry consists of two bytes, one for the predominant height of the surface (trees, buildings) and one for the class of the morphology information.

Because each grid point represents an area of 3 x 3 (3 x 6) seconds, more than one class of morphology is possible, e.g. a part is buildings, an other part is trees. In those cases there are different heights for this area. It is possible, to define more than one class, but only one height.

The height information is one byte. Therefore it is possible to define heights from 0 m to 255 m.

The height is the predominant height of the area represented by this grid point, e.g. if there are 70% buildings with 10 m height, 20% trees with 12 m height and 10% roads with 0 m height, 10 m is taken to represent this area.

The class of morphology consists of one byte. Therefore 8 different classes (bits) are possible. For the fixed service land, sea and coastal area are required, for the mobile service only land and sea are required.

- all bits are 0                      normal land
- bit 0 is 1                          sea, ocean
- bit 1 is 1                          small lake, river, small portions of water (no sea, no ocean!)
- bit 2 is 1                          coastal area
- bit 3 is 1                          villages, towns (buildings)
- bit 4 is 1                          trees
- bits 5 to 7                        for future use

In general, a morphological database is not required for all countries applying the HCM software. If there is no sea or coastal area (e.g. Austria, Slovakia), the use of a morphological database is not mandatory.

The morphological database offered by the TWG HCM SWG Program does not have height information (all heights are 0 m). Only the morphological classes 'normal land', 'sea / ocean' and 'coastal area' are supplied.

All morphological data consists of 2 bytes. The first byte represents the class of morphology; the second byte is the height information.

Morphological data of a 5 x 5 minutes square (approximate 9 x 5 km) is combined in one data-record.

- 12 \* 12 records (=144 records; =1 \* 1 degree) are stored in one file.

- The filename is (example): E007N50.63M  
 where     E007 = Longitude of the South-West corner  
           N50 = Latitude of the South-West corner  
           63 = Resolution in seconds longitude (6) and latitude (3)  
           E = Elevation data (M for morphological data)
- Position of records inside the file:

North

	133	134	135	136	137	138	139	140	141	142	143	144		
	121												132	
	109												120	
	97												108	
	85												96	
	73												84	
West	61												72	East
	49												60	
	37												48	
	25												36	
	13												24	
	1	2	3	4	5	6	7	8	9	10	11	12		

South

- All files with the same longitude are stored in the same subdirectory. The name of the subdirectory is equal to the first four characters of the filename (example: E007).
- All subdirectories are stored in the (top level) directory, e.g. 'MORPHO'. A valid filename with the full path therefore is:

C:\MORPHO\E010\E010N45.33M

**Note**

In older versions of the HCM software, only the drive letter 'C' may be replaced by any other valid drive letter and the top level directory is fixed to 'MORPHO'. In version 7 of the HCM software, it is possible to define the path of the topo-data and only the name of the subdirectory and the file name is fixed, but the old system still works alternatively.

- In Europe, a 5 x 5 minutes square contains north of 50 degrees latitude 101 x 51 values = 5.151 values and south of 50 degrees latitude 101 x 101 values = 10.201 values. One value = 2 bytes. The length of data records therefore is 10.302 or 20.402 bytes.
- Record description:  
     Length: fixed, 10.302 or 20.402 bytes  
     No carriage control!
- The data inside the record is combined from East to West and from South to North.



## CHAPTER 8: GEOGRAPHICAL DATA REQUIREMENTS FOR LINE CALCULATIONS, FIELD STRENGTH PREDICTIONS AND STORAGE OF LINE-DATA

---

### CHAPTER 8.1: DATA REQUIREMENTS

---

#### CHAPTER 8.1.1: THE DIFFERENT CASES

---

For distance calculations, field strength predictions and calculation of the position of a mobile, following geographical data are required:

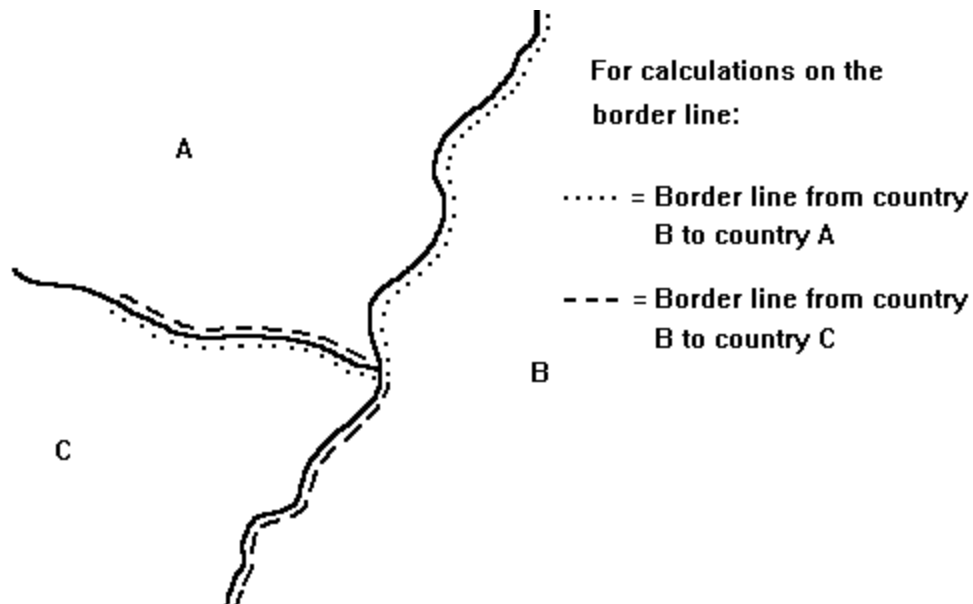
- Borderlines to involved countries,
- Cross border lines according to the Annex 1 of the HCM Agreement,
- X-km lines for preferential frequencies (for own and foreign preferential frequencies).
- Closed borderlines of the own and of involved countries (for calculation of the position of mobiles).

These cases are described below.

#### CHAPTER 8.1.2: BORDER LINES TO INVOLVED COUNTRIES

---

To determine the necessary borderline for the involved countries, the borderline data of the involved countries are required up to a distance of 100 km within these countries.

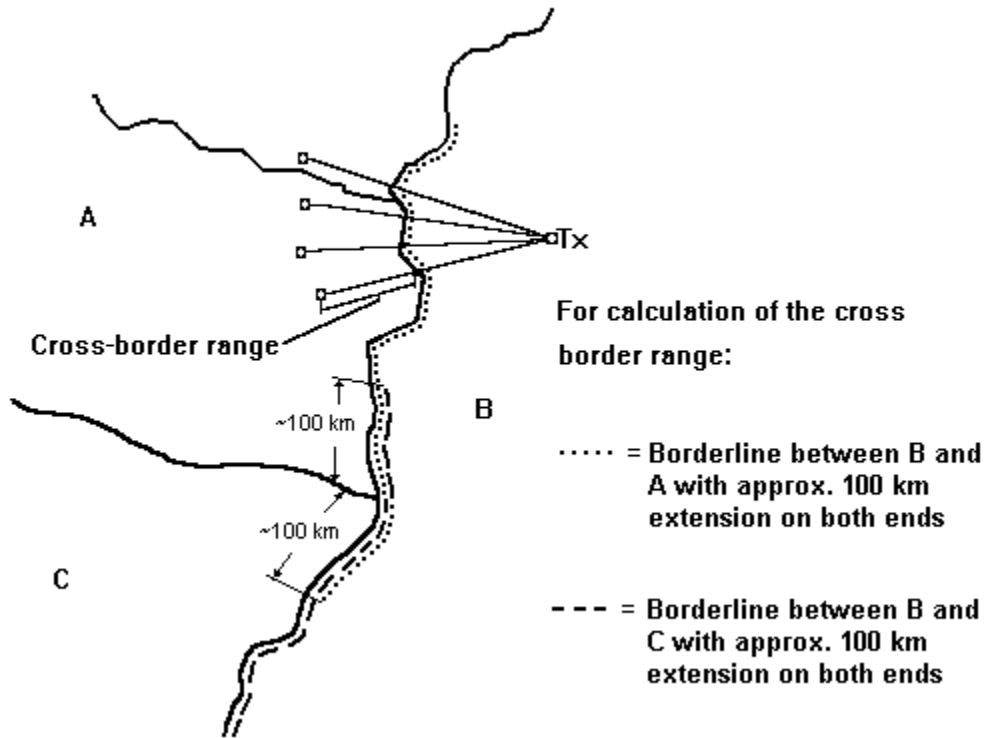


---

CHAPTER 8.1.3: CROSS BORDER LINES (CROSS BORDER RANGES)

---

For calculating the cross border range for Tx the common borderline e.g. between country B and A is used with an extension of approx. 100 km on both ends as shown below.





---

CHAPTER 8.1.4: X-KM LINES OF PREFERENTIAL FREQUENCIES

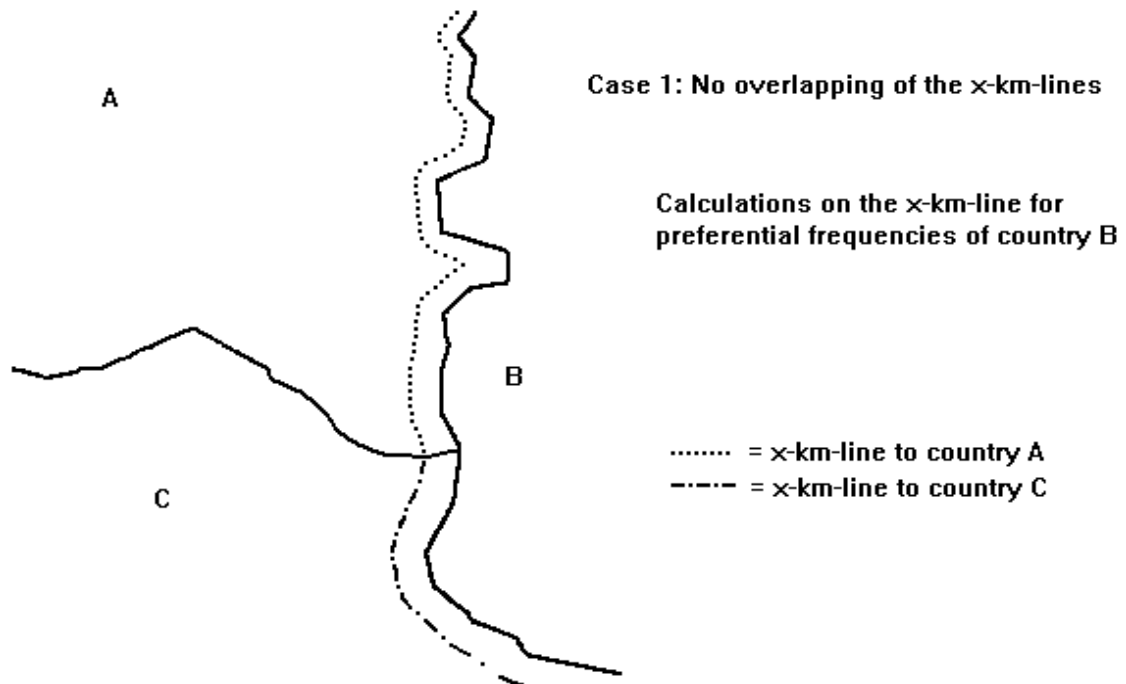
---

To determine the x-km lines in every country involved, the data of these x-km lines with or without an overlap on one or both sides of the border are required. The requirement of an overlap depends on the shape of the borderlines.

8.1.4.1. X-KM LINES WITHOUT AN OVERLAP

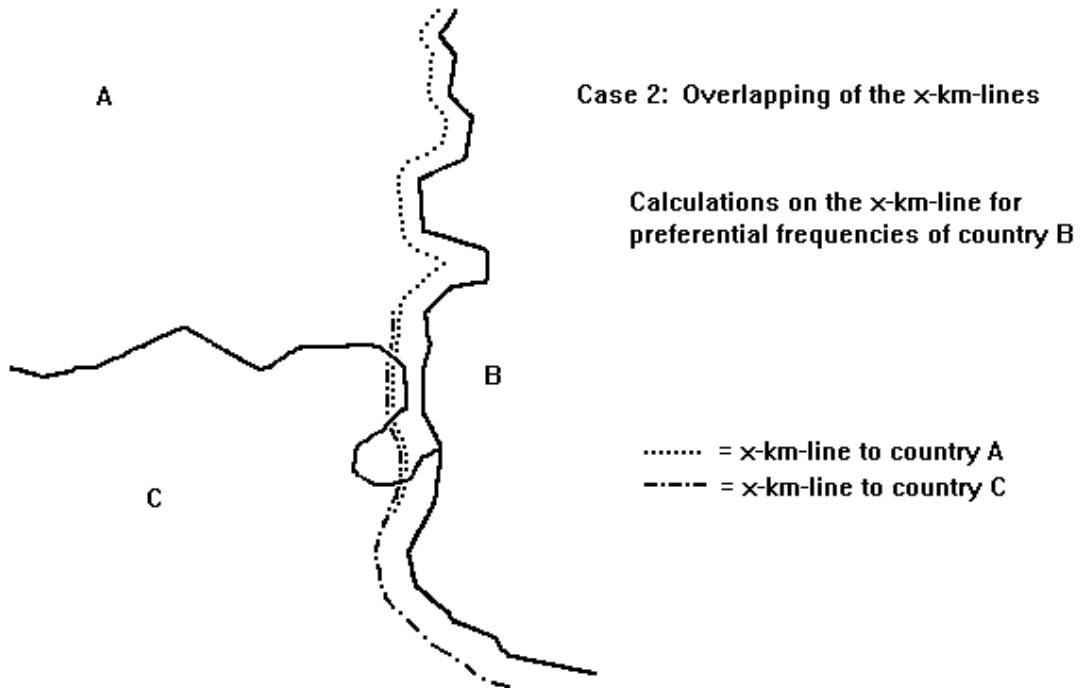
---

The normal condition without an overlap is shown in the picture below.



8.1.4.2. X-KM LINES WITH AN OVERLAP

In some cases, if the borderline e.g. looks like the picture below, an overlap of the x-km-lines is required. How long the overlap is and where the line starts or ends, is subject of a bilateral agreement between the involved countries.

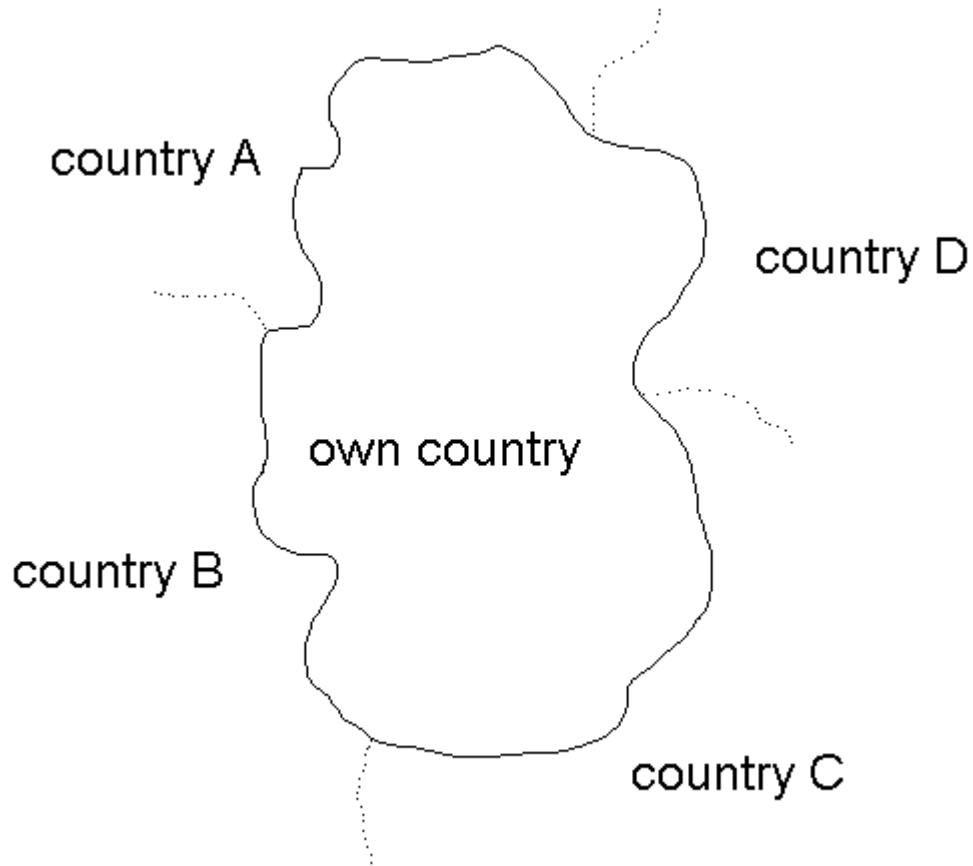


---

CHAPTER 8.1.5: CLOSED BORDERLINES OF THE OWN AND OF INVOLVED COUNTRIES (FOR CALCULATION OF THE POSITION OF MOBILES)

---

To determine the position of a mobile (if the borderline cuts the service area of this mobile), a closed borderline of the country is required. To obtain a closed borderline the last line point of the file has to be equal to the first line point of the file.



The closed border line of the country is stored in the BORDER directory with the name of the country (country code, 3 characters, missing characters are '\_') and the extension 'ALL', e.g. 'HOL.ALL' or 'F\_ALL'

All line data (borderlines, x-km lines, lines for calculating the cross border range) are stored in different files. The names of those files are build using the two country codes (country from – country to) and the distance to the borderline. If the country code is less than 3 letters, the missing places are filled with underscores (e.g. 'F\_'). Borderlines are stored with the extension '.000' (e.g. the borderline between HOL and BEL has the filename 'HOLBEL.000').

Lines to calculate the cross border range use the extension '.CBR'.

Preferential lines (x-km lines) use the value of the distance to the borderline as extension (e.g. '.015' for a 15 km line).

Closed lines to calculate the position of a mobile use the extension '.ALL'.

Line data files consist of fixed length records without carriage control.

To create a record, the co-ordinates of 10 points following each other are selected. The center of these 10 co-ordinates is calculated. This is the 11th point. All co-ordinates are in decimal form, longitude first, latitude second (e.g. 10.14567 45.39876). The result of all 11 points is 22 numbers. These 22 values are converted to radian (value x  $\pi$  / 180) and stored in REAL x 8 variables. All converted 22 values are stored in one record writing 22 x 8 = 176 bytes.

Record:                    long 1, lat 1, long 2, lat 2, ..., long 10, lat 10, long11, lat11

Part of a FORTRAN code to store the data (22 radian values of co-ordinates)

```
PROGRAM TEST
C
DOUBLE PRECISION COORD(22)
CHARACTER*176 LINE
C
EQUIVALENCE (COORD, LINE)
C
OPEN (UNIT=1, FILE='HOLBEL.000', ACCESS = 'DIRECT', RECL = 176)
WRITE (1, REC=1) LINE
C
C LINE and COORD use the same memory space (EQUIVALENCE statement) !
C
CLOSE (UNIT=1)
C
END
```

If for the last record the number of the remaining points is less than 10, the co-ordinates of the last available point is duplicated until 10 co-ordinates are reached.

## CHAPTER 8.3: HOW TO CREATE THE REQUIRED DATABASE WITH THE "BORDER"-PROGRAM

---

The "BORDER" program offers the option to create all required data files. As the input for calculations you need the borderline data from your own country, subdivided in portions according to the different involved countries and up to a distance of 100 km from the own border. All data consist of pairs of geographical co-ordinates of border points in decimal, WGS'84 format. It is required as a text file, in each line longitude and latitude of one border point, first longitude, second latitude. These two values have to be delimited by a blank, a comma or a semicolon (see HELP in the BORDER program, example: 9.523 51.324).

For other computers (e.g. UNIX) the BORDER program presents the possibility to store the line data in ASCII-format. This file(s) have to be moved to the other computer and a small program has to be run to convert this data to the format required for the HCM module. This FORTRAN program is listed in paragraph 4.

This BORDER program, with its user guide, is available for free on the Internet site of the HCM Agreement (<http://hcm.bundesnetzagentur.de>).

## CHAPTER 8.4: FORTRAN PROGRAM TO CONVERT ASCII - LINE-DATA TO HCM-FORMAT

---

```
C
C   ASC_BIN.FOR
C
C   Converting ASCII (border-) line data created with the "BORDER"
C   program to binary data.
C
C   DOUBLE PRECISION      X(22)
C   CHARACTER*176        Y
C   CHARACTER*50          IN, OUT
C   INTEGER               I, J, IOS, IN_L, OUT_L
C
C   EQUIVALENCE (X,Y)
C
C   *****
C
C   Ask for filenames:
10  WRITE (*,*)
    IN = '
    WRITE (*,*) ' Please enter the name of the file containing',
*             ' the ASCII data:'
    READ (*,'(A50)',END=900,ERR=10) IN
C
20  WRITE (*,*)
    OUT = '
    WRITE (*,*) ' Please enter the name of the file for the',
*             ' binary data:'
    READ (*,'(A50)',END=900,ERR=20) OUT
C
C   Length of filenames:
    IN_L = INDEX (IN, ' ') - 1
    IF (IN_L .LT. 1) GOTO 10
    OUT_L = INDEX (OUT, ' ') - 1
    IF (OUT_L .LT. 1) GOTO 20
C
    OPEN (UNIT=1, FILE=IN(1:IN_L), RECL=440)
    OPEN (UNIT=2, FILE=OUT(1:OUT_L), RECL=176, ACCESS='DIRECT')
C
    J = 1
C
100 READ (1,'(22F20.15)',IOSTAT=IOS) (X(I),I=1,22)
    IF (IOS .EQ. -1) GOTO 900
    IF (IOS .NE. 0) THEN
        WRITE (6,*) ' Error reading ASCII data, I/O-status = ',IOS
        GOTO 900
    END IF
    WRITE (2,REC=J,IOSTAT=IOS) Y
    IF (IOS .NE. 0) THEN
        WRITE (6,*) ' Error writing binary data, I/O-status = ',IOS
        GOTO 900
    END IF
    J = J + 1
    GOTO 100
900  CLOSE (UNIT=1)
    CLOSE (UNIT=2)
    END
```

## CHAPTER 9: INTERFACE TO THE HCMMS\_V7 SUBROUTINE IN FORTRAN 90

---

The HCMMS\_V7 program is only a subroutine. To run the program, a surrounding program is required.

The subroutine HCMMS\_V7 is written in FORTRAN 90 language (with Compaq Visual Fortran Professional Edition 6.6C). A simple surrounding program is listed below. A more complex surrounding program is supplied by the HCM SWG Program and can be found on the Internet site of the Agreement.

```
!  
!   Simple surrounding program for the HCMMS_V7 subroutine.  
!  
!   PROGRAM Test  
!  
!   IMPLICIT      NONE  
!   INCLUDE      'HCM_MS_V7_definitions.f90'  
!  
!   Prepare all input data  
!  
!   CALL HCMMS_V7  
!  
!   Show all output data  
!  
!   END PROGRAM Test  
!
```

All input- and output data is defined in the file 'HCM\_MS\_V7\_definitions.f90'. Therefore no arguments are included in the subroutine call.

## Description of all input data

Name	Data type	Description
C_mode	Integer*4	Calculation mode, possible values are:
		12 point to point calculation (t%=1%)
		11 point to point calculation (t%=50%)
		10 point to point calculation (t%=10%)
		0 point to point calculation (t%=Channel occ.)
		-1 point to line calculation (t%=Channel occ., 10m)
		-9 point to line calculation (t%=10%, 3m)
		-10 point to line calculation (t%=10%, 10m)
		-11 point to line calculation (t%=50%, 3m)
More information regarding the several calculation modes are available in the user guide (chapter 6).		
Coo_Tx	Character*15	Tx co-ordinates, format = '015E203052N2040'
Coo_Rx	Character*15	Rx co-ordinates, format = '015E203052N2040' only required if C_mode is 0 or positive
H_Tx_ant	Character*4	Tx antenna height in m
H_Rx_ant	Character*4	Rx antenna height in m only required if C_mode is 0 or positive
Tx_frequ	Character*12	Tx frequency, format = '00147.77000M'
Rx_frequ	Character*12	Rx frequency, format = '00147.77000M' only required if C_mode is 0 or positive
Rad_of_Tx_serv_area	Character*5	Radius of Tx service area
Rad_of_Rx_serv_area	Character*5	Radius of Rx service area only required if C_mode is 0 or positive
H_Tx_input	Character*4	Input value of Tx site height (if available)
H_Rx_input	Character*4	Input value of Rx site height (if available) only required if C_mode is 0 or positive
Max_power	Character*6	Maximum radiated power
Type_of_Tx_ant	Character*1	Type of Tx reference antenna (E / I)
Type_of_Rx_ant	Character*1	Type of Rx reference antenna (E / I) only required if C_mode is 0 or positive
Chan_occup	Character*1	Channel occupation (0 / 1)
PD	DoublePrecision	Point distance for the profile (default value = 0.1 km)
Perm_FS_input	Character*5	Input value of permissible field strength (if not filled in, the value is taken from the table in Annex 1)
Max_CBR_D_input	Character*3	Input value of maximum cross-border range (if not filled in, the value is taken from the table in Annex 1)
Sea_temperature	Character*1	Sea temperature (C / W)
Topo_path	Character*63	Path of the terrain height data (e.g. 'C:\TOPO')
Morpho_path	Character*63	Path of the morphological data (e.g. 'C:\MORPHO')
Border_path	Character*63	Path of the (border-) line data (e.g. 'C:\BORDER')
D_to_border	Integer*4	Distance to border line (for the selection of the type of line calculation, 0 = calculation on the borderline, negative value = calculation of the maximum cross-border range, positive value x = calculation on the x-km line only required if C_mode is 0 or positive
Land_from	Character*3	Country of Tx or country to calculate from



Land_to	Character*3	Country of Rx or country to calculate to
Rx_ant_gain	Character*4	Gain of Rx antenna only required if C_mode is 0 or positive
Depol_loss	Character*4	Depolarization loss only required if C_mode is 0 or positive
Cor_fact_frequ_diff	Character*4	Correction factor according to frequency Difference, only required if C_mode is 0 or positive, if missing, this value is calculated
Azi_Tx_input	Character*5	Tx azimuth
Azi_Rx_input	Character*5	Rx azimuth only required if C_mode is 0 or positive
Ele_Tx_input	Character*5	Tx elevation
Ele_Rx_input	Character*5	Rx elevation only required if C_mode is 0 or positive
D_sea_input	Character*5	Input value of distance over sea; if filled, the calculation of distance over sea is switched off
Ant_typ_H_Tx	Character*7	Horizontal antenna type of Tx
Ant_typ_V_Tx	Character*7	Vertical antenna type of Tx
Ant_typ_H_Rx	Character*7	Horizontal antenna type of Rx only required if C_mode is 0 or positive
Ant_typ_V_Rx	Character*7	Vertical antenna type of Rx only required if C_mode is 0 or positive
Desig_of_Tx_emis	Character*9	Designation of emission of Tx
Desig_of_Rx_emis	Character*9	Designation of emission of Rx only required If C_mode is 0 or positive

## Description of all output data

Name	Data type	Description
HCM_error	Integer*4	Error value, see list of HCM_error values
Info(i)	Logical*4(20)	List of Info(i) values, see list of Info values
Calculated_FS	Real	Calculated field strength
Perm_FS	Real	Permissible field strength
Prot_margin	Real	Protection margin
Free_space_FS	Real	Free space field strength
Distance	DoublePrecision	Distance between Tx position and Rx position
D_sea_calculated	DoublePrecision	Calculated distance over sea (or from input value)
Dir_Tx_Rx	DoublePrecision	Horizontal direction from Tx to Rx
Dir_Rx_Tx	DoublePrecision	Horizontal direction from Rx to Tx
V_angle_Tx_Rx	DoublePrecision	Vertical direction from Tx to Rx
V_angle_Rx_Tx	DoublePrecision	Vertical direction from Rx to Tx
H_diff_angle_Tx_Rx	DoublePrecision	Horizontal difference angle from Tx to Rx
H_diff_angle_Rx_Tx	DoublePrecision	Horizontal difference angle from Rx to Tx
V_diff_angle_Tx_Rx	DoublePrecision	Vertical difference angle from Tx to Rx
V_diff_angle_Rx_Tx	DoublePrecision	Vertical difference angle from Rx to Tx
Delta_frequency	DoublePrecision	Frequency difference in kHz
Heff_Tx	Real	Tx effective antenna height
Heff_Rx	Real	Rx effective antenna height
Heff	Real	Total effective antenna height
Dh	Real	Terrain irregularity
Dh_corr	Real	Correction factor according to the terrain irregularity
Tx_TCA	Real	Tx clearance angle
Rx_TCA	Real	Rx clearance angle
Tx_TCA_corr	Real	Correction factor according to the Tx clearance angle
Rx_TCA_corr	Real	Correction factor according to the Rx clearance angle
ERP_ref_Tx	Real	ERP of the reference transmitter
Land_FS	Real	Land field strength
Sea_FS	Real	Sea field strength
Tx_ant_corr	Real	Correction factor according to the Tx antenna type (horizontal and vertical)
Rx_ant_corr	Real	Correction factor according to the Rx antenna type (horizontal and vertical)
Tx_ant_type_corr	Real	Correction factor according to the Tx reference antenna type (E / I)
Rx_ant_type_corr	Real	Correction factor according to the Rx reference antenna type (E / I)
Perm_FS_from_table	Real	Permissible field strength from the Table in Annex 1
Corr_delta_f	Real	Correction factor according to the Frequency difference (calculated or from the input value)
Channel_sp_Tx	Real	Channel spacing of Tx in kHz
Channel_sp_Rx	Real	Channel spacing of Rx in kHz
Power_to_Rx	Real	Power in direction of Rx
CBR_D	Real	Maximum cross-border range in km (from input value or from table in Annex 1)
Version	Character*5	HCMMS_V7 version number
Coo_Tx_new	Character*15	Calculated Tx co-ordinates

Coo_Rx_new	Character*15	Calculated Rx co-ordinates (or line co-ordinates)
T_Prof(i)	Integer*2 (10002)	Terrain height profile
M_Prof(i)	Integer*2 (10002)	Morphological profile
PN	Integer*2	Number of profile points
H_Datab_Tx	Integer*2	Tx site height from terrain database
H_Datab_Rx	Integer*2	Rx site height from terrain database

---

## CHAPTER 10: INTERFACE TO THE HCMMS\_V7.DLL

---

To simplify the use of the HCM software, the HCM team build a dynamic link library HCMMS\_V7.DLL.

A lot of programming languages (e.g. Visual Basic, C++) are able to work with DLL's.

An example in Visual Basic is available from the HCM team.

There are some restrictions in some programming languages: Boolean - and String variables are difficult to handle and the number of arguments is limited. Therefore some modifications in the interface are required:

- No Boolean variables are passed,
- Only one String is passed.

The Interface to (and from) the DLL is:

There is one Subroutine called HCMMS\_V7\_DLL.

The arguments are:

### *C\_mode*

An input value; the mode of calculation; a 4 byte INTEGER (LONG in VB); permissible values = -11 to +12; Values -2 to -8 and 2 to 10 are deprecated.

### *Bor\_dis*

An input value; the distance to the borderline; a 4 byte INTEGER (LONG in VB); Values: 0 = calculations are performed on the borderline, positive value x = calculations are performed at the x-km line, any negative value = calculation of the maximum cross-border range.

### *PD*

An input value; the distance between two profile points (grid size) in km; an 8 byte DOUBLE PRECISION (DOUBLE in VB); this value has to be set to 0.1 (default value).

### *Distance*

An output value; the distance between transmitter and the receiving point in km; an 8 byte DOUBLE PRECISION (DOUBLE in VB).

### *H\_Datab\_Tx*

An output value; a 2 byte INTEGER (INTEGER in VB); the height of the transmitter site above sea level from the terrain database in m.

### *H\_Datab\_Rx*

An output value; a 2 byte INTEGER (INTEGER in VB); the height of the receiver site above sea level from the terrain database in m. This value is only valid if CMODE is positive!

### *HCM\_error*

An output value; a 4 byte INTEGER (LONG in VB); the error code. Error codes are listed in the HCM documentation. An additional error code is generated by the DLL software:

Number: 3000

Description: The string variable passed to the DLL is too short (less than 432 characters).

#### *Heff*

An output value; a 4 byte REAL (SINGLE in VB), the effective antenna height in m used for the calculations according the ITU\_R method.

#### *Dh*

An output value; a 4 byte REAL (SINGLE in VB), the terrain irregularity in m used for the calculations according the ITU\_R method.

#### *Dh\_corr*

An output value; a 4 byte REAL (SINGLE in VB), the correction factor according to the terrain irregularity in dB used for the calculations according the ITU\_R method.

#### *Power\_to\_Rx*

An output value; a 4 byte REAL (SINGLE in VB), the power in the direction of the receiver in dBW.

#### *Free\_space\_FS*

An output value; a 4 byte REAL (SINGLE in VB), the free space field strength in dB $\mu$ V/m.

#### *Land\_FS*

An output value; a 4 byte REAL (SINGLE in VB), the land field strength in dB $\mu$ V/m.

#### *Sea\_FS*

An output value; a 4 byte REAL (SINGLE in VB), the sea field strength in dB $\mu$ V/m.

#### *Tx\_ant\_corr*

An output value; a 4 byte REAL (SINGLE in VB), the correction factor in dB according to the transmitter antenna type (horizontal and vertical).

#### *Tx\_ant\_type\_corr*

An output value; a 4 byte REAL (SINGLE in VB), the correction factor in dB according to the transmitter antenna type ('E' or 'I').

#### *Dir\_Tx\_Rx*

An output value; a DOUBLE PRECISION (DOUBLE in VB), the horizontal direction from the transmitter to the receiver in degrees.

#### *V\_angle\_Tx\_Rx*

An output value; a DOUBLE PRECISION (DOUBLE in VB), the vertical direction from the transmitter to the receiver in degrees.

#### *Tx\_TCA*

An output value; a 4 byte REAL (SINGLE in VB), the transmitter clearance angle in degrees.

#### *Rx\_TCA*

An output value; a 4 byte REAL (SINGLE in VB), the receiver clearance angle in degrees.

#### *Tx\_TCA\_corr*

An output value; a 4 byte REAL (SINGLE in VB), the correction factor according to the transmitter clearance angle in dB.

#### *Rx\_TCA\_corr*

An output value; a 4 byte REAL (SINGLE in VB), the correction factor according to the receiver clearance angle in dB.

#### *D\_sea\_calculated*

An output value; a DOUBLE PRECISION (DOUBLE in VB), the distance over sea in km taken into account during the calculations (either input value or calculated value).

#### *Rx\_ant\_corr*

An output value; a 4 byte REAL (SINGLE in VB), the correction factor in dB according to the receiver antenna type (horizontal and vertical).

#### *Rx\_ant\_type\_corr*

An output value; a 4 byte REAL (SINGLE in VB), the correction factor in dB according to the receiver antenna type ('E' or 'I').

#### *Delta\_frequency*

An output value; a DOUBLE PRECISION (DOUBLE in VB), the frequency difference between transmitter - and receiver frequency in kHz.

#### *Corr\_delta\_f*

An output value; a 4 byte REAL (SINGLE in VB), the correction factor according to the frequency difference between transmitter - and receiver frequency in dB.

#### *Calculated\_FS*

An output value; a 4 byte REAL (SINGLE in VB), the calculated field strength in dB $\mu$ V/m.

#### *Perm\_FS*

An output value; a 4 byte REAL (SINGLE in VB), the permissible field strength in dB $\mu$ V/m (input value or calculated value).

#### *CBR\_D*

An output value; a 4 byte REAL (SINGLE in VB), the maximum cross border range in km (input value or from Agreement).

#### *ERP\_ref\_Tx*

An output value; a 4 byte REAL (SINGLE in VB), the power of the reference transmitter in dBW.

#### *Prot\_margin*

An output value; a 4 byte REAL (SINGLE in VB), the protection margin in dB (difference of calculated field strength and permissible field strength).

#### *I\_str*

An input / output value (a part of this string is only input, an other part is only output). A CHARACTER (STRING in VB) variable with at least 432 characters. The content of this string is:

Start pos.	Stop pos.	Description	In-/Output	Format
1	15	Tx co-ordinates	Input	009E223350N4422
16	30	Rx co-ordinates	Input	009E223350N4422
31	34	Height of Tx site	Input	9999 or four blanks
35	38	Height of Rx site	Input	9999 or four blanks
39	45	Tx hor. ant. Type	Input	000ND00
46	52	Tx vert. ant. type	Input	000ND00
53	57	Tx Azimuth	Input	999.9
58	62	Tx Elevation	Input	+99.9
63	66	Tx Antenna height	Input	9999
67	70	Rx Antenna Height	Input	9999

71	71	Tx Type of antenna	Input	'E' or 'I'
72	77	Max. radiated power	Input	+999.9
78	89	Tx Frequency	Input	99999.99999M
90	90	Channel occupation	Input	'0' or '1'
91	91	Sea temperature	Input	'C' or 'W'
92	96	Tx Service area	Input	99999
97	101	Rx Service area	Input	99999
102	106	Distance over sea	Input	99999 or 5 blanks
107	118	Rx Frequency	Input	99999.99999M
119	127	Rx Design. of emis.	Input	14k0.....
128	136	Tx Design. of emis.	Input	14k0.....
137	143	Rx hor. ant. Type	Input	000ND00
144	150	Rx vert. ant. Type	Input	000ND00
151	155	Rx Azimuth	Input	999.9
156	160	Rx Elevation	Input	+99.9
161	161	Rx Type of ant.	Input	'E' or 'I'
162	165	Rx Antenna gain	Input	99.9
166	169	Depolarization loss	Input	99.9
170	174	Perm. Field strength	Input	+99.9 or 5 blanks
175	178	Corr. fact. acc. freq. diff.	Input	99.9 or four blanks
179	181	County code to calcul. to	Input	XXX
182	184	County code of Tx	Input	XXX
185	187	Max. cross border range Input	Input	999 or 3 blanks
188	250	Path of terrain data	Input	C:\TOPO..... (63 C)
251	313	Path of borderline data	Input	C:\BORDER (63 C)
314	376	Path of morphological data	Input	C:\MORPHO (63 C)
377	382	Version number	Output	
383	402	Info values (field, 20 values)	Output	'T' true or 'F' false
403	417	Tx co-ordinates calculated	Output	008E213651N2137
418	432	Rx co-ordinates calculated	Output	007E413672N4127
433	???	Debug directory (optional)	Input	e.g. 'C:\Temp'

## Note

If more than 432 characters are passed, the remaining part (> 432) must be a valid directory name.

If a valid directory is given, the HCMMS\_V7\_DLL will write a text-file into this directory: DEBUG.TXT every time it is called.

DEBUG.TXT contains all input values passed to the DLL and stores all output values coming from the DLL.

## Additional files

To run the DLL on your system, you need some additional run-time files.

The self-installing executable kit 'VFRUN66CI.exe' installs the run-time components needed to run Visual Fortran applications on systems, which do not have Visual Fortran, installed.

The components installed by this kit are:

- DFORRT.DLL - Visual Fortran non-threaded run-time support
- DFORMD.DLL - Visual Fortran threaded run-time support
- FQWIN.HLP - QuickWin run-time help file
- MSVCRT.DLL - Microsoft Visual C run-time support
- OLEAUT32.DLL - Microsoft OLE Automation
- OLEPRO32.DLL - Microsoft OLE Automation
- STDOLE32.TLB - Microsoft OLE Automation

Files will be installed only if newer versions do not exist on the system. The files in this kit can be used with earlier versions of Visual Fortran.

The file 'VFRUN66CI.exe' is available on the server.



## ANNEX: LIST OF ERROR CODES AND INFO(I) VALUES

### HCM\_error values

0	No error
36	Error opening terrain- or morphological data file (data not available)
200	Error in longitude (in 'Point_height' or 'Point_type' subroutine)
210	Error in latitude (in 'Point_height' or 'Point_type' subroutine)
220	Error reading record (in 'Point_height' or 'Point_type' subroutine)
300	Latitude is not in range of 0.0 - 90.0 (in 'Point_height' or 'Point_type' subroutine)
400	Height is missing (-9999) (in 'Point_height' subroutine)
1000	Distance between Tx and Rx = 0. Calculations not possible
1001	Error in geographical coordinates (Tx longitude, degrees)
1002	Error in geographical coordinates (Tx longitude, minutes)
1003	Error in geographical coordinates (Tx longitude, seconds)
1004	Error in geographical coordinates (Tx longitude, E/W)
1005	Error in geographical coordinates (Tx latitude, degrees)
1006	Error in geographical coordinates (Tx latitude, minutes)
1007	Error in geographical coordinates (Tx latitude, seconds)
1008	Error in geographical coordinates (Tx latitude, N/S)
1009	Error in Tx antenna height
1010	Error in transmitting frequency value
1011	Error in transmitting frequency unit
1012	Error in radius of service area of Tx
1013	Error in input value height of Tx site above sea level
1014	Error in geographical coordinates (Rx longitude, degrees)
1015	Error in geographical coordinates (Rx longitude, minutes)
1016	Error in geographical coordinates (Rx longitude, seconds)
1017	Error in geographical coordinates (Rx longitude, E/W)
1018	Error in geographical coordinates (Rx latitude, degrees)
1019	Error in geographical coordinates (Rx latitude, minutes)
1020	Error in geographical coordinates (Rx latitude, seconds)
1021	Error in geographical coordinates (Rx latitude, N/S)
1022	Error in Rx antenna height
1023	Error in reception frequency value
1024	Error in reception frequency unit
1025	C_mode is out of range
1026	Error in input value of permissible field strength
1027	Error in input value of maximum cross border range
1028	The distance is greater than 1000 km. Calculations not possible
1029	Error in radius of Rx service area
1030	Error in input value Rx site height above sea level
1031	Error in Tx elevation
1032	Error in Tx azimuth
1033	Error in type of Tx antenna (E/I)
1034	Error in power
1035	Error in input value of distance over sea
1036	The 'xxx.ALL' borderline file for Tx is missing
1037	The 'xxx.ALL' borderline file for Rx is missing

1038	Error in type of antenna
1039	Error in input data of correction factor according frequency difference
1040	Channel spacing outside definition range (Rx)
1041	Channel spacing outside definition range (Tx)
1042	Error in Rx elevation
1043	Error in Rx azimuth
1044	Error in Rx type of antenna ("E" or "I")
1045	Error in gain of Rx antenna
1046	Error in input data of depolarization loss
1047	Distance to borderline is too long
1048	Selected line data not available
1049	Error in line data
1050	No HCM Agreement frequency and important technical data missing e.g. CBR, max. perm. FS
2000	wrong Figure_frequency (from Get_figure_FS_value)
2001	wrong Time_percentage (from Get_figure_FS_value)
2002	wrong Sea_temperature (from Get_figure_FS_value)
2003	wrong Figure_Heff (from Get_figure_FS_value)
2004	wrong Figure_distance (from Get_figure_FS_value)
3000	DLL inputstring to short

### Info(i) values

1	No height of Tx site is given or Tx is mobile; height is taken from the terrain database
2	Height of Tx site differs from height of terrain database
3	Height of Tx site differs more than 10%, calculated values may be (extremely) wrong!
4	Frequency out of range of table in Annex 1
5	Input value of permissible field strength is used
6	Input value of maximum cross border range is used
7	Distance between Tx and Rx is less than both service area radiuses; field strength is set to 999.9
8	No height of Rx site is given or Rx is mobile/line, height is from the terrain database
9	Height of Rx site differs from height of terrain data
10	Rx site height differs more than 10%, calculated values may be (extremely) wrong!
11	Free space field strength used because distance < 1 km
12	Free space field strength is used, because 1st Fresnel zone is free
13	Distance over sea is greater than total distance. Distance between Tx and Rx is used!
14	Input value of correction factor according frequency difference is used
15	Frequency difference outside definition range; 82 dB is used
16	Calculated distance over sea is set to 0 because of missing morphological data.
17	Tx channel spacing outside definition range, 25 kHz is used!
18	Correction factors for the band 380 - 400 MHz are used.