Site Investigation based on Return Flow in Horizontal Directional Drilling
Marsdiep Project

Tom Hijnekamp
Site Investigation based on Return Flow in Horizontal Directional Drilling

Marsdiep Project

Master Thesis

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday April 18, 2016 at 11:00 AM.

Student: T.A.J. (Tom) Hijnekamp
Student number: 1508466
E-mail: tomhijnekamp@gmail.com

Supervisor: Dr. Ir. D.J.M. Ngan-Tillard, TU Delft
Thesis committee: Prof. Dr. Ir. J.D. Jansen, TU Delft
Dr. Ir. W. Broere, TU Delft
D.A. den Hamer, MSc., TU Delft
Ir. J.J. van Meerten, Deltares
Dr. H.M.G. Kruse, Deltares

This thesis is confidential and cannot be made public until September 1, 2016.

An electronic version of this thesis is available at http://repository.tudelft.nl/.
Preface

This thesis is the conclusion of the MSc track of Geo-Engineering at the faculty of Civil Engineering and Geo-sciences of Delft University of Technology. The thesis is supported by and conducted at Deltares.

When I started the context of my thesis was roughly known, i.e. within the Marsdiep project at Deltares there were different subjects suitably for a master thesis. The Marsdiep project was the investigation into the feasibility of constructing a pipeline between Texel and Den Helder using Horizontal Directional Drilling (HDD).

At the start in March I wrote a research proposal, in which I was not only going to relate the samples to the soil by studying the influences on and the processes going on in the drilling fluid but I was going to correct the initial estimation of the formation parameters as well. Unfortunately for me it turned out to be as ambitious as unrealistic given the scope of the data acquisition (on which I had very limited influence due to contractual and financial requirements) and the time restraints. As soon as this became clear I decided to change the focus of this research.

Since my background is Applied Earth Sciences I decided to focus more on the geology. Increasing the scale and limiting the scope to a more reasonable level, allowed me to take more distance from the fluid and transport processes going on in the borehole.

In order to improve the geological cross-section I still required the source locations of the samples. The source locations are calculated by rewriting, expanding and adapting a method I used on site in Den Helder to calculate the diameter of the borehole during a flow test. After some trial and error and a significant amount of time spend on data processing, the results can be found in this thesis.

Of course I could not have done this without support and input from others. First I would like to thank my graduation committee for their support. Without detracting from the input and efforts of the other members of the graduation committee, I would especially like to thank Dianne Den Hamer, from TU Delft, who was working for Deltares on site in both Texel and Den Helder, and was very helpful in getting this research to the point where it is now. From André Pietjouw from Episcope, Jonno Pouw and Ton Bennen, both from PWN and all present on site in both locations, I learned a lot about the HDD process as well as about regular practice in the construction industry, for which I am thankful.

Furthermore I would like to thank the following persons that were working on the Marsdiep project and related projects and were always available for questions: Wim Post, David Nugroho and Mike Woning, all working at Deltares. I would also like to thank Dick Mastbergen and Arno Talmon, both working at Deltares, for their help with questions I had with regard to fluid properties and processes. From Deltares I would also like to thank Jonathan Nuttall for his help and patience with regards to programming related issues. In addition I want to thank Freek Busschers and Jeroen Schokker, both working at TNO, for identifying the samples and helping me understand the geology of the region.

Last but not least I would like to thank my friends and family. By asking questions and showing interest they forced me to take a look from different perspectives which led to new insights and improvements of this research.

Tom Hijnekamp
Delft, April 2016
The focus of this research is on samples taken from the return flow. With these samples and within the context of the Marsdiep project it is investigated if Horizontal Directional Drilling (HDD) can be used as a Site Investigation (SI) technique. In the Marsdiep project the feasibility of a pipeline constructed using HDD between the island Texel and Den Helder is investigated. Marsdiep is the name of the water and tidal gully between Texel and Den Helder. Regular SI techniques are not feasible, due to among others; the length of 4.5km, the depth that is required for a HDD of this length and the tidal gully.

The geology of the area is unknown so SI is required. For this purpose pilot drillings, the first part of a regular HDD project, are made at NAP-65m and NAP-85m from both Texel and Den Helder. These pilot drillings are adapted so that additional data are acquired.

The rheologic properties are determined for samples taken from the return flow and the formation of the samples is determined after sieving.

A range of possible source locations of the samples are calculated, using different scenarios based on the uncertainty with regards to the borehole diameter and the loss of drilling fluid. The results of the scenarios are compared to the results of the initial SIs, which consists of a seismic survey, onshore, vertical drillings and Cone Penetration Tests (CPTs). It is found that the source location of the samples can be calculated as one of the scenarios gave accurate results (scenario 5, drilling fluid loss 7% and a borehole diameter of 0.4m and 0.45m for respectively Texel and Den Helder).

A geological cross-section is constructed based on the results of scenario 5 and initial SIs.

In addition to the geology the suitability of HDD as a SI technique is investigated. It is found that with minor adaptations to a regular HDD it can be used in combination with other techniques to determine the geology of an area. The other SIs are required to compare the results to and to determine the best scenario.

In order for HDD to be of use as a separate SI technique more extensive adaptations are required to minimize the uncertainties that required the use of scenarios in this research.
De nadruk van dit onderzoek ligt op samples die genomen zijn van de return flow. Met deze samples en in het kader van het Marsdiep project is onderzocht of het mogelijk is horizontaal gestuurd boren te gebruiken als grondonderzoek techniek. In het Marsdiep project wordt de haalbaarheid onderzocht van het maken van een pijplijn tussen Texel en Den Helder gebruik makend van gestuurd boren. Marsdiep is de naam van het water en de getijden geul tussen Texel en Den Helder. Reguliere grondonderzoek technieken zijn niet wenselijk wegens onder andere de lengte van 4.5 km, de benodigde diepte voor een gestuurde boring van deze lengte en de getijden geul.

De geologie in dit gebied is onbekend dus grondonderzoek is nodig. Met dit doel zijn er pilot boringen, de eerste fase van een regulier gestuurd boren project, uitgevoerd op NAP-65m en NAP-85m van zowel Texel als Den Helder. Deze pilot boringen zijn aangepast zodat er extra data kunnen worden verzameld.

De vloeistof eigenschappen van de samples uit de return flow is bepaald en de formatie van de samples is bepaald na zeving.

Een bereik aan mogelijke oorsprong locaties is berekend, gebruik makend van verschillende scenarios gebaseerd op de onzekerheden met betrekking tot de diameter en het verlies van boorvloeistof. De resultaten van de verschillende scenarios zijn vergeleken met de resultaten van een initieel grondonderzoek, wat bestaat uit seismisch onderzoek, verticale boringen op Texel en in Den Helder en meerdere Cone Penetration Tests (CPT’s). Uit dit onderzoek blijkt dat het mogelijk is om de oorsprong te berekenen aangezien de resultaten van een van de scenarios goed kloppen wat al bekend was (scenario 5, 7% verlies van boorvloeistof en een boorgat diameter van 0.4m en 0.45m voor respectievelijk Texel en Den Helder.).

Een geologische doorsnede is gemaakt op basis van de resultaten van scenario 5 en van de initieele grondonderzoeken.

Ook is de geschiktheid van gestuurd boren als grondonderzoek techniek onderzocht. Het blijkt dat met kleine aanpassing ten opzichte van een reguliere boring het gebruikt kan worden in combinatie met andere grondonderzoek technieken om de geologie van het gebied te bepalen. De andere grondonderzoek technieken zijn nodig om de resultaten mee te kunnen vergelijken en zo het meest waarschijnlijke scenario te bepalen.

Om gestuurd boren als enige techniek te gebruiken zijn meer en uitgebreidere aanpassingen nodig om de onzekerheden te minimaliseren die het gebruik van scenarios vereisten in dit onderzoek.
Contents

Preface iii
Abstract iv
Samenvatting v
Abbreviations xi
Definitions xiii
Symbols xv

1 Introduction 1
1.1 Goal 1
1.2 Thesis structure 1

2 Literature Review 3
2.1 Geology 3
2.1.1 Geological History 6
2.2 Risks 6
2.2.1 Project Risks 6
2.2.2 General Risks 7
2.3 HDD 8
2.3.1 History of HDD (research) 8
2.3.2 The HDD Process 8
2.4 Comparison of HDD with Directional Drilling (DD) 12

3 Marsdiep Project 13
3.1 Locations and Dimensions 13
3.1.1 Initial Cross-section 13
3.1.2 Site Investigations 14
3.2 Calculations 16
3.3 Alignment 16

4 Pilot Drilling Descriptions 17
4.1 Site Setup and Data Acquisition 17
4.2 Description of the Pilot Drillings 19
4.2.1 Tests 19
4.2.2 Texel 20
4.2.3 Den Helder 21
4.3 Sampling Method 21
4.4 Suitability of HDD for SI 22

5 Processes 23
5.1 Literature Overview 23
5.2 Different Fluid Models 23
5.2.1 Determination of Bingham Parameters 24
5.2.2 Determination of Herschel-Bulkley Parameters 24
5.2.3 Comparison Fluid Models and Determination Methods 25
5.3 Flow of Drilling Fluids 27
5.3.1 Infiltration Loss 29
5.4 Transport 30
5.4.1 Transport Parameter 33
5.5 Lag Time or Source Location 33
6 Methodology ............................................ 35
   6.1 Equipment and Tests .............................. 35
   6.2 Sample Source Program ......................... 36
      6.2.1 Calculation of the Source Location (Intersection Method) 36
      6.2.2 Scenarios .................................. 37
   6.3 Marsdiep Visual Program ....................... 41
   6.4 Fixes Program .................................. 43

7 Measurements .................................. 45
   7.1 Supplied Data ................................ 45
      7.1.1 ProData System ........................... 45
      7.1.2 Survey Data (Drilling Paths) .......... 46
      7.1.3 Driller Log ................................ 50
      7.1.4 Drill Company Data ..................... 50
   7.2 Acquired Data ................................ 51
      7.2.1 On site ................................... 51
      7.2.2 Deltares, Delft ......................... 53

8 Interpretation .................................. 55
   8.1 Drilling Fluid ................................ 55
      8.1.1 Fluid Model .............................. 55
      8.1.2 Time Dependent Behavior ............ 59
   8.2 Transport and Flow Processes ............... 60
      8.2.1 Flow Type ................................ 60
      8.2.2 Infiltration Loss ....................... 60
      8.2.3 Transport ................................ 60
   8.3 Sample Identification ......................... 61
   8.4 Source Location ................................ 64
      8.4.1 Lag Diagram .............................. 66
   8.5 Cross-section ................................ 68
   8.6 Improved Cross-section ....................... 77
   8.7 Summary ...................................... 80

9 Conclusions ...................................... 81
   9.1 Drilling Fluid ................................ 81
      9.1.1 Fluid Model .............................. 81
      9.1.2 Time Dependent Behavior ............ 81
      9.1.3 Transport and Flow Parameters ...... 81
   9.2 Source Location ................................ 82
      9.2.1 Lag Diagram .............................. 82
   9.3 Cross-section ................................ 82
   9.4 Suitability of HDD for SI ...................... 82

10 Recommendations ............................. 85
   10.1 Drilling Fluid ................................ 85
      10.1.1 Rheological Data Acquisition ........ 85
      10.1.2 Fluid Model .............................. 85
      10.1.3 Processes ................................ 85
   10.2 Source Location ................................ 85
   10.3 Cross-section ................................ 86
   10.4 Suitability of HDD for SI ...................... 86

Bibliography ........................................ 89
Appendices ........................................ 93
   A Geology Background .............................. 96
   B Initial Site Investigations, Proposals and Calculations 103
   C Equipment and Tests ........................... 113
D  Additional Results  117
E  Code of the SampleSource Program  133
F  Code of the MarsdiepVisual Program  206
Abbreviations

AP
Appelscha Formation

API
American Petroleum Institute

BH
Bingham

BTL
Boren van Tunnels en Leidingen. Translated into English; drilling tunnels and pipelines.

BX
Boxtel Formation

CPT
Cone Penetration Test

DD
Directional Drilling

DR
Drente Formation

DS
Drillstring

EC
Electro-conductivity

EE
Eem Formation

EE-URTY
Eem Formation, Urk Tynje section

EE-BB
Eem Formation, Bruine Bank section

GBB
Gezamelijk Basisonderzoek Boortechnieken. Translated into English it is; joint fundamental research drilling techniques.

HB
Herschel-Bulkley
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDD</td>
<td>Horizontal Directional Drilling</td>
</tr>
<tr>
<td>KR</td>
<td>Kreftenheye Formation</td>
</tr>
<tr>
<td>NA</td>
<td>Naaldwijk Formation</td>
</tr>
<tr>
<td>NAP</td>
<td>Normaal Amsterdams Peil, Translated into English; Amsterdam Ordnance Datum.</td>
</tr>
<tr>
<td>PE</td>
<td>Peelo Formation</td>
</tr>
<tr>
<td>PENI</td>
<td>Peelo Formation, Nieuwolda section</td>
</tr>
<tr>
<td>PPI</td>
<td>Plastics Pipe Institute</td>
</tr>
<tr>
<td>PZ</td>
<td>Peize Formation</td>
</tr>
<tr>
<td>ROP</td>
<td>Rate Of Penetration</td>
</tr>
<tr>
<td>RPM</td>
<td>Rotations Per Minute</td>
</tr>
<tr>
<td>SI</td>
<td>Site Investigation</td>
</tr>
<tr>
<td>TBM</td>
<td>Tunnel Boring Machine</td>
</tr>
<tr>
<td>URVE</td>
<td>Urk Formation, Veenhuizen section</td>
</tr>
</tbody>
</table>
Definitions

**A.Hak-Drillcon**
Drillcon is the department of the firm A.Hak that is concerned with horizontal drilling. It was the main contractor in Texel.

**Backflow**
Pressure driven flow, that comes out of the borehole while the pumps are off.

**Carriage**
The part of the rig that is connected to the drillstring, the drill fluid flows through it and it delivers the pushing force on the drillstring.

**Coupling Time**
The time required to attach or detach a pipe from the Drillstring (DS), during which the pumps are off.

**Critical flow**
The minimum annular average fluid velocity that would prevent stationary accumulation of cuttings in a bed (Pilehvari et al., 1999).

**Deltares**
A Major Technological Research Institute in the Netherlands which is concerned with research in the areas of water, soil and infrastructure.

**DGeoPipeline**
Deltares program for calculations on pipelines.

**DINO**
A database maintained by TNO, containing most of the Site Investigations (SIs) executed in the Netherlands. DINO stands for 'Data en Informatie van de Nederlandse Ondergrond', translated into English it is; Data and information of the Dutch subsurface.

**Directional drilling**
This term is used to make a distinction between the use of directional drilling in the petroleum industry, often referred to as directional drilling, and construction industry, where the drilling method is called horizontal directional drilling.

**Drilling fluid or Mud**
A fluid, often a mixture of water and bentonite, which has different tasks. Amongst others; suspending and removing cuttings, cleaning and lubricating the drill bit and pipe and minimizing hole erosion (Ariaratnam et al., 2007).

**Drillstring**
A set of pipes, also called joints, connecting the drill bit to the drill rig, transports the drilling fluid to the drill front and can contain a cable for data transfer between the drill and the surface.

**GeoTOP**
A model of the dutch soil that currently spans parts of the Netherlands, with a grid size of 100x100m and vertical resolution of 0.5 meter.
**Horizontal directional drilling**

See directional drilling.

**Joint**

A pipe of the drillstring, the joint number is the number of pipes in the borehole.

**LMR**

LMR is the part of a larger group of companies called Ludwig Freytag that is concerned with horizontal drilling. It was the main contractor in Den Helder.

**Mud cake**

Plastering takes place in the borehole; other names include: filter cake, external plaster.

**Mud crust**

Plastering takes place in the formation; other names include: mud spurt, internal plaster.

**Mud pulse**

A system that sends pressure pulses through the drilling fluid in the drillstring in order to transfer data from the drill head to the surface.

**PWN**

Drinking water company, responsible for the environmental management of dunes along the coast of North Holland and the transport infrastructure in this region. The client for the Marsdiep project.

**RD Coordinates or Rijksdriehoekscoördinaten**

The national coordinate system of the Netherlands.

**Reamer**

A drill head used in increasing the diameter of an existing borehole.

**Return flow**

The drilling fluid coming out of the borehole.

**Soil parameters**

The parameters of the soil, with regard to the strength or permeability for example.

**TNO**

A Major Technological Research Institute in the Netherlands which has as goal: "TNO connects people and knowledge to create innovations that boost the sustainable competitive strength of industry and well-being of society." It is active in a number of different work areas.

**Tripping**

There are two types of tripping. Tripping back: pulling back the DS. Tripping in; moving forward in an existing borehole. For example after changing the jet bit to a mud motor.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>A factor used in calculating infiltration loss, value = 8/75</td>
<td>—</td>
</tr>
<tr>
<td>α_H</td>
<td>Angle of inclination, between the boring and the horizontal axis</td>
<td>°</td>
</tr>
<tr>
<td>α_V</td>
<td>Angle of inclination, between the boring and the vertical axis</td>
<td>°</td>
</tr>
<tr>
<td>c_v</td>
<td>Volumetric concentration of the grains in the drilling fluid</td>
<td>—</td>
</tr>
<tr>
<td>K</td>
<td>Consistency Index as used in the Herschel Bulkley and Power Law fluid models</td>
<td>Pa.s^n</td>
</tr>
<tr>
<td>CF1</td>
<td>Used to convert [°] to [mPa]</td>
<td>—</td>
</tr>
<tr>
<td>CF2</td>
<td>Used to convert [RPM] to [1/s]</td>
<td>—</td>
</tr>
<tr>
<td>V_T(Hor.)</td>
<td>The cuttings velocity on the horizontal part of the boring</td>
<td>m/s</td>
</tr>
<tr>
<td>V_T(Incl.)</td>
<td>The cuttings velocity on the inclined part of the boring</td>
<td>m/s</td>
</tr>
<tr>
<td>ρ_f</td>
<td>Density of the fluid</td>
<td>kg/m^3</td>
</tr>
<tr>
<td>ρ_s</td>
<td>Density of the grains</td>
<td>kg/m^3</td>
</tr>
<tr>
<td>Δ</td>
<td>Deviation of the calculated shear stress from the measured shear stress</td>
<td>—</td>
</tr>
<tr>
<td>D_15</td>
<td>A value for the diameter of grains at which 15% of the grains are smaller</td>
<td>m</td>
</tr>
<tr>
<td>D_90</td>
<td>A value for the diameter of grains at which 90% of the grains are smaller</td>
<td>m</td>
</tr>
<tr>
<td>D_BH</td>
<td>Diameter of the borehole</td>
<td>m</td>
</tr>
<tr>
<td>d</td>
<td>Diameter of the grains</td>
<td>m</td>
</tr>
<tr>
<td>δ</td>
<td>Distance between center of borehole en center of DS</td>
<td>mm</td>
</tr>
<tr>
<td>D_p</td>
<td>The pore diameter, assumed to be 4/9 * D_15</td>
<td>m</td>
</tr>
<tr>
<td>ε_factor</td>
<td>Eccentricity, defined as the distance between center of DS and borehole divided by the borehole radius minus the DS radius, all in [mm]</td>
<td>—</td>
</tr>
<tr>
<td>f</td>
<td>Fannings-friction coefficient</td>
<td>—</td>
</tr>
<tr>
<td>n</td>
<td>Flow Index as used in the Herschel Bulkley and Power Law fluid models</td>
<td>—</td>
</tr>
<tr>
<td>g</td>
<td>Gravitational constant</td>
<td>m/s^2</td>
</tr>
<tr>
<td>He</td>
<td>Hedstrom number</td>
<td>—</td>
</tr>
<tr>
<td>κ</td>
<td>A blockage factor used in calculating the infiltration loss, the value depends on the soil; 2 or 3 is common for sand</td>
<td>—</td>
</tr>
<tr>
<td>L</td>
<td>Length of the borehole</td>
<td>m</td>
</tr>
<tr>
<td>L_H</td>
<td>Length of the horizontal part of the borehole</td>
<td>m</td>
</tr>
<tr>
<td>L_I</td>
<td>Length of the inclined part of the borehole</td>
<td>m</td>
</tr>
<tr>
<td>m</td>
<td>local exponential factor, required to calculate the generalized Reynolds number</td>
<td>—</td>
</tr>
<tr>
<td>—</td>
<td>Millions of years</td>
<td>MA</td>
</tr>
<tr>
<td>n_0</td>
<td>Porosity</td>
<td>—</td>
</tr>
<tr>
<td>ψ_T</td>
<td>Normalized shear stress</td>
<td>—</td>
</tr>
<tr>
<td>OD_DS</td>
<td>outer diameter of the DS</td>
<td>m</td>
</tr>
<tr>
<td>ΔP_ph</td>
<td>Pressure difference between the borehole and the surrounding groundwater</td>
<td>Pa</td>
</tr>
<tr>
<td>PV</td>
<td>Plastic viscosity, used in the Bingham model</td>
<td>mPa.s</td>
</tr>
<tr>
<td>S</td>
<td>Pressure drop, in meters of water column, 1 mwk/m is about 10 kPa (9.80665 kPa)</td>
<td>(mwk/m)^6</td>
</tr>
<tr>
<td>dP/dL</td>
<td>Pressure drop gradient</td>
<td>kPa/m</td>
</tr>
<tr>
<td>P_output</td>
<td>Pump rate</td>
<td>L/min</td>
</tr>
<tr>
<td>ΔQ_il</td>
<td>The volume that is lost due to infiltration into the soil</td>
<td>m^3/s</td>
</tr>
<tr>
<td>R_h</td>
<td>Hydraulic radius, defined as A/P, where A is the cross-section and P is the wetted perimeter</td>
<td>m</td>
</tr>
<tr>
<td>r_p</td>
<td>Plug flow radius</td>
<td>m</td>
</tr>
<tr>
<td>RO_300RPM</td>
<td>Readout of the viscometer at 300 Rotations Per Minute (RPM)</td>
<td>°</td>
</tr>
<tr>
<td>RO_600RPM</td>
<td>Readout of the viscometer at 600 RPM</td>
<td>°</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Unit</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>$Re$</td>
<td>Reynolds number</td>
<td>—</td>
</tr>
<tr>
<td>$Re_{gen,full}$</td>
<td>Reynolds number, generalized</td>
<td>—</td>
</tr>
<tr>
<td>$R_{fac}$</td>
<td>A factor for the hydraulic diameter, 1.5 for an annulus</td>
<td>—</td>
</tr>
<tr>
<td>$R_i$</td>
<td>Radius of the drillstring</td>
<td>m</td>
</tr>
<tr>
<td>$R_o$</td>
<td>Radius of the borehole</td>
<td>m</td>
</tr>
<tr>
<td>$ROP$</td>
<td>Rate of Penetration</td>
<td>m/s</td>
</tr>
<tr>
<td>$\bar{ROP}$</td>
<td>The average Rate of Penetration</td>
<td>m/s</td>
</tr>
<tr>
<td>$\psi$</td>
<td>Scaling factor used in correcting the yield point for a more accurate result in the Herschel Bulkley model</td>
<td>—</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Shear rate</td>
<td>1/s</td>
</tr>
<tr>
<td>$\dot{\gamma}_i$</td>
<td>Shear rate at setting $i$ of the viscometer</td>
<td>1/s</td>
</tr>
<tr>
<td>$\gamma_{rf}$</td>
<td>The shear rate of the return flow</td>
<td>1/s</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Shear stress</td>
<td>Pa</td>
</tr>
<tr>
<td>$S_{DX(\text{Hor.})}$</td>
<td>The $x$-coordinate of the source of the cuttings, on the horizontal part of the boring</td>
<td>m</td>
</tr>
<tr>
<td>$S_{DX(\text{Incl.})}$</td>
<td>The $x$-coordinate of the source of the cuttings, on the inclined part of the boring</td>
<td>m</td>
</tr>
<tr>
<td>$S_{DY(\text{Hor.})}$</td>
<td>The $y$-coordinate of the source of the cuttings, on the horizontal part of the boring</td>
<td>m</td>
</tr>
<tr>
<td>$S_{DY(\text{Incl.})}$</td>
<td>The $y$-coordinate of the source of the cuttings, on the inclined part of the boring</td>
<td>m</td>
</tr>
<tr>
<td>$\tau_{i,\text{calc}}$</td>
<td>Calculated shear stress at setting $i$ of the viscometer</td>
<td>Pa</td>
</tr>
<tr>
<td>$\tau_{i,\text{data}}$</td>
<td>Measured shear stress at setting $i$ of the viscometer</td>
<td>Pa</td>
</tr>
<tr>
<td>$\tau_w$</td>
<td>Shear stress due to friction</td>
<td>Pa</td>
</tr>
<tr>
<td>$\tau_y$</td>
<td>Yield stress of the fluid</td>
<td>Pa</td>
</tr>
<tr>
<td>$t_s$</td>
<td>Average time that a grain is present in the fluid in the borehole</td>
<td>s</td>
</tr>
<tr>
<td>$t_r$</td>
<td>Relative settle time</td>
<td>s</td>
</tr>
<tr>
<td>$t_k$</td>
<td>Average time that is required for a grain to settle</td>
<td>s</td>
</tr>
<tr>
<td>$Y_g$</td>
<td>A transport parameter found in BTL 42</td>
<td>—</td>
</tr>
<tr>
<td>$\Psi_{TB}$</td>
<td>Dimensionless transport parameter</td>
<td>—</td>
</tr>
<tr>
<td>$\Phi_{TB}$</td>
<td>Transport capacity for bed flow</td>
<td>—</td>
</tr>
<tr>
<td>$\Phi_{TP}$</td>
<td>Transport capacity for plug flow</td>
<td>—</td>
</tr>
<tr>
<td>$v_{\text{Drill}}$</td>
<td>The average drill velocity</td>
<td>m/s</td>
</tr>
<tr>
<td>$v_{rf}$</td>
<td>Flow velocity of the mixture containing drilling fluid and soil</td>
<td>m/s</td>
</tr>
<tr>
<td>$v_{\text{settle}}$</td>
<td>Settle velocity grain</td>
<td>m/s</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Viscosity of the drilling fluid</td>
<td>Pa.s</td>
</tr>
<tr>
<td>$\mu_\infty$</td>
<td>Constant viscosity at very high shear rates</td>
<td>Pa.s</td>
</tr>
<tr>
<td>$w_i$</td>
<td>Weighting factor, given by $\tau_i - \tau_y$</td>
<td>—</td>
</tr>
<tr>
<td>$y_a$</td>
<td>Infiltration depth</td>
<td>m</td>
</tr>
<tr>
<td>$y_c$</td>
<td>Cake thickness</td>
<td>m</td>
</tr>
<tr>
<td>$YP$</td>
<td>Yield point as used in the Bingham Model</td>
<td>Pa</td>
</tr>
</tbody>
</table>
Introduction

The drinking water of Texel, an island in the northern part of the Netherlands, is supplied from the mainland through two pipes in shallow trenches in the seabed of the Marsdiep. In 2013 the cover above one of these pipes eroded. The pipe floated up and started to scour along the seabed under influence of the tidal currents. This movement slowly increased the free floating length and caused an increase in stresses in the pipe. The scouring wore the pipe wall down, where it left the seabed, to such an extent that the material failed to cope with the stresses and started to leak before breaking (Mesman et al., 2014). During a heatwave in the summer from that year, additional water had to be transported to Texel to meet the demand.

In order to prevent this from happening again an investigation is started in order to determine whether a pipeline at considerable depth could be constructed using Horizontal Directional Drilling (HDD). This investigation is known as the Marsdiep project, named after the waterway between Texel and the mainland.

However, the geology between Texel and Den Helder is unknown and due to the location standard Site Investigations (SIs) techniques as required for more conventional HDD projects, ie. crossing a river or road, would be too expensive. Therefore it is relevant to know if HDD can be used as an alternative to regular SIs. Initial SIs have taken place and consist of one onshore vertical drilling and four Cone Penetration Tests (CPTs) on both sides and a seismic survey of the Marsdiep.

With a length of more than 4 kilometer this is one of, if not the longest HDD project to date. This means that the forces and required pressures are considerably higher than during a regular project. Due to the pressure required to transport the soil in the borehole to the surface the required depth is considerable. To lower the required pressure ‘meeting in the middle’ is proposed, this halves the distance the soil needs to be transported. ‘Meeting in the middle’ entails drilling from both sides and when one drilling is positioned above the borehole of the other drilling it is gradually lowered till there is one borehole.

In order to investigate the soil pilot drillings are planned. A pilot drilling is the first phase of a HDD project. A preliminary design is constructed based on the available data. This design will be checked and improved based on results of the pilot drillings.

Although HDD is used in the Netherlands since the seventies of the previous century, numerous aspects are less than well understood. In order to address these uncertainties additional data are gathered on site.

1.1. Goal
The goals of this research must be seen within the context of the following broader research question: is HDD suitable as a SI technique? The focus of this research is on the samples taken from the return flow. The goal of this research is to answer the following questions:

- Is it achievable to connect samples from the return flow to a source location in the subsurface?
- Is it possible to create a cross-section with the results of the samples from the return flow?
- Is HDD suitable as a SI technique and can it be improved, based on experiences in the field?

Furthermore the drilling fluid is an important item in this thesis. An investigation of the drilling fluid is required in order to investigate and answer the first question.

1.2. Thesis structure
Chapter 2 presents the results of a literature study. In this chapter sections detailing the geology of the Marsdiep area, risks, the HDD method and a section comparing HDD with Directional Drilling (DD) as practiced in the petroleum industry can be found.

Chapter 3 briefly presents the Marsdiep project and the initial SIs.
Chapter 4 presents the execution of the pilot drillings.

Chapter 5 presents a theoretical approach to processes happening in the borehole.

Chapter 6 presents the programs and method created in order to reach the goals of this thesis.

Chapter 7 presents characteristic measurements of the pilot drillings. These include measurements taken in the laboratory as well as taken from the drilling rig.

Chapter 8 presents the interpretation of measurements as well as the results of the programs, eg. the fluid model and source locations for the samples.

Chapter 9 presents the conclusions and chapter 10 contains the recommendations for further research.

The references and appendices can be found at the end.
2

Literature Review

In this chapter an overview is presented of what is known about the geology of the Marsdiep area, making use of both public available knowledge as well as the initial Site Investigations (SIs) performed for this project. Risks related to the geology are investigated as well. The Horizontal Directional Drilling (HDD) method is explained and a comparison is made between HDD and Directional Drilling (DD) as practiced in the petroleum industry.

2.1. Geology

In order to know which processes occur in the borehole during HDD the geology needs to be known. Based on the description in the literature, databases and on the preliminary survey consisting of boreholes, CPTs and seismic data, the formations are identified and very briefly described in (Nugroho and van Meerten, 2015). The English description of the formations can be found in table A.2, appendix A. Table A.1, also in appendix A, contains the grain size definitions.

Table 2.1 contains the formations encountered in the Marsdiep project. Within the aim of the thesis it is very important to know more about these formations, in order to be able to correlate the spoils of the return flow to the formations encountered. This is done by evaluating what is already known, collected and investigated by TNO, a dutch institute which maintains the DINO database, which contains the results of most SIs performed in the Netherlands. Figure 2.1 shows all of the formations found in the Netherlands that have been deposited during the Quaternary. The descriptions from TNO are general but they are extensive with respect of what is possible and give insights into the origin of the formations. On the differences in origin, a distinction might be made when investigating soil samples from the return flow. The differences can be used to determine if the sample is mixed, ie. contains characteristics of different formations, due to flow and transport processes in the borehole.

The investigation of the geology is divided in two parts; first the geologic history of the area and second the formation descriptions.
Table 2.1: Summary of formations encountered during the Marsdiep project.

<table>
<thead>
<tr>
<th>Fm.</th>
<th>Age</th>
<th>Local Occurrence</th>
<th>Den Helder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naaldwijk (NA)</td>
<td>0.01 MA - Present</td>
<td>Brown to gray, moderately fine, weak to moderately silty sand, which can contain traces of iron oxide, traces of micas, shells, small clay layers and traces of detritus at the bottom of the formation.</td>
<td>Brown to gray, moderately fine, well sorted, weak silty sand, which can contain traces of organic material, traces of micas, shells and clay as well as some peat and gyttja layers at the bottom of the formation.</td>
</tr>
<tr>
<td>Boxtel (BX)</td>
<td>0.66 MA - Present</td>
<td>Gray, very to moderately fine, moderately to very silty sand, which can contain traces of organic material and small layers of peat and clay.</td>
<td>Light brownish gray, very fine, moderately sorted, slightly silty sand.</td>
</tr>
<tr>
<td>Kreftenhaye (KR)*</td>
<td>0.24 MA - 0.01 MA</td>
<td>Brownish, colored, coarse to medium grained sand. Can contain peat, shell fragments and organic debris.</td>
<td></td>
</tr>
<tr>
<td>Eem (EE)</td>
<td>0.13 MA - 0.12 MA</td>
<td>Gray to dark gray, well sorted sand, which can contain a few detritus layers at the top of the formation, traces of colored material and clay layers.</td>
<td>Dark yellow to brownish gray, very sandy clay with micas or a brownish gray, very to moderately fine, well sorted, slightly silty sand with traces of micas.</td>
</tr>
<tr>
<td>Urk - Tynje (URTY)</td>
<td>0.85 MA - 0.13 MA</td>
<td>Gray, moderately fine to very coarse, slightly to very silty sand, with traces of colored material, shells, white quartz and organic material and a variation in sorting.</td>
<td>Light gray to gray, slightly silty moderately coarse sand, with traces of organic material and shells and badly to moderately well sorted.</td>
</tr>
<tr>
<td>Urk - Veenhuizen (URVE)</td>
<td>0.85 MA - 0.13 MA</td>
<td>Gray, weak silty moderately fine to moderately coarse sand, with traces of colored material, pink grains, few black grains and well sorted.</td>
<td></td>
</tr>
<tr>
<td>Peelo - Nieuwolda (PENI) - 'Pot Clay'</td>
<td>0.47 MA - 0.42 MA</td>
<td>Dark gray, very strong clay containing some gravel.</td>
<td>Gray to dark gray, moderately to badly sorted, moderately coarse to very coarse sand, which can contain silt, gravel and traces of white quartz.</td>
</tr>
<tr>
<td>Peelo (PE)</td>
<td>0.47 MA - 0.42 MA</td>
<td></td>
<td>Gray, moderately to very coarse, weak silty sand with some shells and white quartz grains.</td>
</tr>
<tr>
<td>Appelscha (AP)</td>
<td>1.07 MA - 0.47 MA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Not separately identified.
Figure 2.1: Overview of the formations found in the Netherlands, deposited during the Quaternary Period, with their age and origin (TNO, 2013b).
2.1.1. Geological History

The oldest formation that is expected at the depths considered for the project is from the Pliocene epoch in the Neogene period. This is taken as the starting point for the geological history study.

During the end of the Reuverian, the last period of the Pliocene, the temperature drops, in such an extent that all flora characteristic for the Neogene disappears. This enables other flora types to dominate the forests. From these new types an average temperature of about thirteen or fourteen degrees Celsius has been determined. This change in temperature, visible in the flora, marks the end of the Neogene and the start of the Quaternary (de Mulder et al., 2003).

The Quaternary is a dynamic period for the Netherlands. The North Sea Basin originating from the early Neogene continues to sink which results in almost 1000 meter of sediment in the center of the basin. In the southern part of the Netherlands the London-Brabant Massif and the Rhines Massif are continuing to rise, while volcanism in the Eiffel continues into the Pleistocene. The movements of the soil can be divided into three mechanisms:

- Compaction of the layers present; the speed of the compaction increases towards the center of the basin, to an average of 5 mm per century.
- Isostatic movement; due to loading caused by filling of the basin with sediment, growing and melting glaciers and a changing sea level, the isostatic movement centers on the basin as well and causes a subsidence rate of on average 15 mm per century during this period.
- Local tectonic movement; faults in the Roer Valley graben cause a subsidence of on average 6 mm per century in the Roer Valley graben and the western part of the Netherlands while the southern and eastern parts of the Netherlands rise with an average of 2 mm per century.

These three processes combined, caused a complex combination of marine, fluvial and coastal depositions of sediment (de Mulder et al., 2003).

The height of the sea level has significant effects on the position of the coastline. However it is far from the only determining factor, since tectonic movement, change in volume or location of rivers and changes in the sediment brought by these rivers can affect the process as well. The effect of different factors is hard to identify. Which process is dominant depends on the time scale. When considering a timescale of tens of millions of years the dominant factor determining the size of the oceans is plate tectonics, causing a change in the order of 300 - 500 meter (Haq et al., 1987), while on a scale of 100.000 years the amount of land ice is the more dominant factor. Besides these two, the mechanisms mentioned earlier have an influence as well (compaction, isostatic movement and local tectonic movement). Therefore it is more common to use the relative change in sea level, which can be used to mean land or sea changes.

One of the most renowned characteristic of the Quaternary is the relative rapid succession of glacial and interglacial periods. During the glacial periods the rivers transport relatively coarse materials and the wind moves large amounts of fine sand due to a lack of vegetation. The subsurface is also influenced by permafrost which partially thaws in the summer and freezes in winter, causing cryoturbation. During the interglacial periods the climate is similar to the current climate. Since a glacial period lasts for about 80.000 to 90.000 years and an interglacial periods lasts for about 10.000 to 20.000 years, it is more often cold than warm in the Netherlands during the Quaternary.

To summarize; a change in any of the above described factors influenced the deposition processes and circumstances causing a complex mixture of marine, fluvial and coastal sediments in the North-Western part of the Netherlands.

2.2. Risks

A brief overview of the most important risks related to HDD projects are presented here. A complete overview would be too extensive to present here. The risks are divided into specific to the project and risks related to HDD.

2.2.1. Project Risks

The risks presented in table 2.2 are a selection from the risk identified in the initial research (Nugroho and van Meerten, 2015), this selection is elaborated upon for this research. One of the risks is further elaborated on after the table.
2.2. Risks

Table 2.2: An expanded selection of the risks identified in the initial research (Nugroho and van Meerten, 2015). The depths are two of the three originally proposed depths, the other being NAP-45m. The pilot drillings are executed at NAP-65m and NAP-85m.

<table>
<thead>
<tr>
<th>Depth</th>
<th>Risk</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAP-65m</td>
<td>Part of the drilling path goes through the valley fill, the composition of which is unknown. There is a risk of encountering gravel and larger rocks. At high concentrations of gravel and cobbles the drilling could get stuck or the pulling through of the product pipe could be prevented or encounter additional problems. Even a single boulder could prevent the drilling from succeeding.</td>
<td>At higher concentrations gravel or cobbles the drilling should be relocated to another path. The chance of encountering high concentrations is difficult to estimate.</td>
</tr>
<tr>
<td>NAP-100m</td>
<td>At larger depths more formations of coarse sand exist. Due to the possibility of confined gravel layers in coarse sand there is a risk of loss of return flow and a risk of deterioration of the drilling fluid. Both processes can lead to borehole instability, which can lead to failure.</td>
<td>If coarse sand and gravel layers are encountered over large lengths the risk of having to abort the drilling is great. Due to the length and the limited soil investigation the risks are high.</td>
</tr>
<tr>
<td>Encountering Pot Clay</td>
<td>Pot clay is encountered in Texel in the onshore SIs but it is not identified in the seismic survey of the Marsdiep. When encountered the Pot Clay can cause problems such as; additional friction to getting the Drill-string (DS) stuck, both due to the swelling capacity. It can also cause problem with regards to the progress due to its strength is may take a lot of time but it can also cause the drill bit to deviate from the proposed path leading to additional friction on the DS and making the installation of the pipeline more difficult.</td>
<td>The risk of encountering thick pot clay layers is not large. For small lengths the Pot Clay can be drilled through but if the Pot Clay is encountered at the horizontal depth another depth free of Pot Clay has to be selected.</td>
</tr>
</tbody>
</table>

Pot Clay
Pot Clay refers to the Nieuwolda section of the Peelo Formation. The formation is deposited during an ice age and compressed by subsequent ice age. This formation is often present in deep incised valleys which can reach a depth of hundreds of meters. The Pot Clay encountered in Texel is assumed to be the edge of such an incised valley with orientation west-east and the valley further to the north. The clay of the Nieuwolda section is very compact and is strong to very strong. It has a light to dark gray, brownish black to black color and can contain chalk, micas and can be weak to moderately silty or sandy. The clay is glaciolacustrine deposit, which means that it is deposited by meltwater at the edges of glaciers. The strength of the clay stems from subsidence of the water-rich clay and compaction by a glacier during the Saale Glaciation.

2.2.2. General Risks
Two of the main risks related to HDD in general are elaborated on in this section, these are borehole instability and seepage.

Borehole Instability
Borehole instability can influence a project in multiple ways depending both on the length of the instability and the severity of the instability, ie. the amount of additional soil that enters the borehole. The severity ranges from ‘raining in’ to total collapse of the borehole which results in damaged or loss of equipment. ‘Raining in’ is when well sorted soils are insufficiently supported by the drilling fluid and no interlocking occurs causing a slow
‘raining in’ of grains into the drilling fluid. If the amount of additional soil does not exceed the transport capacity this won’t lead to total hole collapse. The following causes for borehole instability are identified (Kruse, 2009):

- Loosely packed granular soils.
- The presence of very permeable gravel layers, no mud crust and or mud cake is formed and thus the walls are not stabilized.
- High groundwater pressure, for example in an aquifer.
- Groundwater flow.
- Extreme low strength of the layers.
- Chemical reaction of the drilling fluid with the groundwater, causing for example the drilling fluid to flocculate.
- Chemical reaction of the drilling fluid with the soil.

Seepage
This term covers multiple processes and risks, it can be used to describes events ranging from blow outs to the surface to leakage of groundwater into the borehole. The following causes for seepage are identified in (Kruse, 2009):

- A high hydraulic head in an aquifer, which exceeds the static downward pressure of the drilling fluid.
- Reduction of the density of the drilling fluid during the drilling phases.
- Excavations at the entry or exit point for the tie-in weld of the pipeline.
- Chemical reaction of drilling fluid with ground water.
- External caused of high groundwater pressure such as pipeline damage.

2.3. HDD
This section is subdivided into the History of HDD (research), and the HDD Process, with a focus on the Pilot Phase, since that phase is where this thesis focuses on. In appendix C.2 a brief, general section can be found on equipment used in HDD.

2.3.1. History of HDD (research)
Since its appearance in the Netherlands, HDD and other trenchless technologies have proved their usefulness and advantages over traditional methods that make use of a trench. This led to the creation of a number of initiatives to improve the knowledge about these techniques in the Netherlands, funded by the industry alone or with (partial) funding from the government. This is partially due to the fact that there is a knowledge gap with respect to applicability in weak soils, which are characteristic for the Netherlands (Arends, 1998).

The two main research programs conducted in the Netherlands, both consisting of a multitude of separate research projects and lasting multiple years, are;

- Boren van Tunnels en Leidingen. Translated into English; drilling tunnels and pipelines. (BTL), which translates to Drilling Tunnels and Pipelines. Conducted during the 1990s.
- Gezamelijk Basisonderzoek Boortechnieken. Translated into English it is; joint fundamental research drilling techniques. (GBB), which translates to ‘Collective fundamental research drilling techniques’. It is partially based upon the BTL reports as well as a continuation of the BTL research. It is conducted in the first few years of the 2000s.

Afterwards no fundamental research, on this scale, has taken place in the Netherlands. HDD is a growing market outside of the Netherlands as well. For example in Canada and the USA its use is increasing as well. Canada and the USA are facing a growing problem in rehabilitating and replacing their decaying underground utility systems causing an increase in the demand for technologies such as HDD (Zayed and Mahmoud, 2013).

2.3.2. The HDD Process
The general, idealized drilling process can be described as follows; pressure and pressurized drilling fluid are delivered to the drill bit from the drilling rig at the surface. The pressurized drilling fluid flows through the drill bit, infiltrating and excavating the soil. The pressure from the drilling rig causes the drill bit to move forward. The drilling fluid forms a mud crust (located in the formation) and a mud cake (located in the borehole). This also happens in front of the drill bit so that the groundwater is partially replaced by drilling fluid in the soil soon to be excavated. The crust and cake prevent intrusion of significant volumes of groundwater in the drilling fluid.
2.3. HDD

The crust and the cake also strengthen the wall of the borehole, increasing the borehole stability. The drilling fluid transports the drilled soil to the surface where the fluid is recycled and pumped down again. In reality the process is less simple due to a number of reasons and processes, for example the flow is not continuous due to connection of new drill pipes at the surface.

Phases
The HDD process can be divided into different phases: pilot, reaming and pull in phase (Kruse, 2009; Denekamp et al., 2000).

Pilot Phase
During this phase an initial hole is drilled from location A (entry point in figure 2.2a) to location B (exit point in figure 2.2a). Behind the drill bit a steering tool is located in a non-magnetic part of the drill string, which monitors the location and progress of the drill bit. In some cases, at or near the surface, a casing can be used. A casing can be used to prevent blow out, erosion of the borehole and to prevent deviation from the planned drill path through weak soils. The drill fluid flows back to the surface, between the drill string and the soil.

In some cases when the length of the drilling becomes too long the pilot boring is started from both sides, this is called the intersect or meeting in the middle method. What is considered too long, depends on the project. But in the Marsdiep project, where the intersect method is going to be applied, the reason is that the drilling fluid pressure that is required to ensure proper borehole cleaning, would exceed the limit put on the drilling fluid pressure with regard to blowout prevention.

Reaming Phase
During this phase the diameter of the borehole is increased in one or more steps to fit the final product pipe or conduit, see figure 2.2b. To increase the borehole a reamer is pulled from location B to location A. During this phase the drill fluid can flow to either or both sides of the borehole.

Pull In Phase
During this phase the pipe or conduit is pulled through from location B to location A. A reamer is pulled through followed by a the pipe with a swivel in between to prevent torque, from the rotation of the drill string, on the pipe or conduit, see figure 2.2c.
Drill Mechanisms and Techniques
HDD makes use of two drilling mechanisms. One is a jet bit which cuts using the drilling fluids flowing through nozzles, which is mainly used in soft soils. The other one is a downhole mud motor which is used in stronger soils (Royal et al., 2010). The mud motor is powered by the flow of the drilling fluid. The introduction of a drill bit that can rotate separately from the drill string has led to two drilling techniques; drilling (or rotary drilling) and sliding (or slide drilling). Where rotary drilling enables drilling straight and sliding enables the construction of turns, see Figure 2.3, (Royal et al., 2010). Both techniques can be used with a jet bit or a mud motor.
Position determination

Different methods can be used to determine the position of the drill head, depending on the local situation and project. For example, a commonly used method is the Walkover method. This uses a sonde, after the drill bit, that sends an electro-magnetic signal to a hand held receiver on surface. This method is only suitable at limited depth (<10 m) and the surface has to be free of steel structures. In both Texel and Den Helder a Gyro Steering Tool is used. Advantages of the gyro method are (Brownline B.V., 2010):

- No disturbances due to magnetic interference.
- No access on surface required.
- No restriction in built-up areas.
- No limits on drill depth.
- Continuous measurements.
- Real time mud pressure in borehole.

To get the data on the surface initially a mudpulse is used in Texel. This device sends pressure pulses through the drilling fluid to the surface which can be translated to data. Later the data transfer is switched to a cable within the Drillstring (DS), see section 4.2.2 for the reasons behind this decision. Due to the cable being present in the flow inside the DS, it can be damaged by the fluid. This caused delays in Texel since the equipment has to be pulled back to replace the cable. In Den Helder data are transferred through a wire in the DS which in this case caused no delays due to pulling the DS back for repairs.

Drilling Fluids

The drilling fluid often consists of a mixture of water with bentonite and/or polymers. Bentonite is a natural occurring clay mineral (mostly montmorillonite), which, when mixed with water, can be used as a carrier fluid. Different polymers can be added to create a fluid with exact specifications. Mixing drilling fluid becomes increasingly difficult as different polymers are added since the polymers affect more than one property of the drilling fluid. It is crucial to have a drilling fluid suitable for the project because of the following functions (Ariaratnam, 2001; Ariaratnam et al., 2007):

1. Transporting the drilled soil to the surface by suspending and carrying it in the flow.
2. Stabilizing the borehole, by creating a non- or low permeable mud cake or crust, the drilling fluid also exerts a hydrostatic pressure on the surroundings. The combination prevents leakage into the borehole of groundwater and prevents borehole collapse.
3. Keeping the drilling bit clean as well as cooling the equipment.
4. Lubricating the equipment and reducing the friction between the drill string and the wall of the borehole.
5. When a mud motor is used the fluid flow is powering the mud motor.

The most important factors in deciding what drilling fluid is to be used are the specific conditions at the project site, so this has to be investigated properly. As is presented above point 3, 4 and 5 are site independent and almost any fluid will fulfill these functions. Site conditions covers both the soil present as well as the groundwater present, for example pH and salt content, since salt can cause the bentonite to flocculate and lead to a change in the properties of the drilling fluid. The material which is used as a basis in Texel is an OCMA (a quality mark) bentonite. In Den Helder, Teqgel (a mixture of bentonite and polymers in order to guarantee certain properties) is used.

Advantages

HDD is a trenchless technique, which has advantages over more traditional trench technology, especially with regards to the social costs, i.e. disturbances of any kind, since it can be constructed more quickly, requires less working space and can be used without disruption to the surface activities (traffic and pedestrian areas) (Ariaratnam, 2001). In densely populated areas, such as the Netherlands, these advantages become extremely significant.

HDD has some advantages when comparing it with other trenchless technologies as well. Examples are:

- No vertical shafts required, HDD operates from surface to surface.
- Relative short setup and installation time.
- Flexibility in the borehole alignment and thus the ability to evade obstacles. For example existing conduits and pipelines.
- A longer installation length than achievable with other techniques in one drive (Allouche et al., 2003) as cited in (Zayed and Mahmoud, 2013).
2.4. Comparison of HDD with Directional Drilling (DD)

The differences between HDD and Directional Drilling (DD) in the petroleum industry make a comparison difficult but since the amount of research conducted in the petroleum industry with regards to DD dwarfs the amount of research done within the HDD industry (Baumert et al., 2005), it is worthwhile to look into it, to see whether it applies or not.

For example the flow velocity is a factor which is significantly different; in HDD it is a low velocity laminar flow which causes little or no erosion to the borehole walls. In DD the flow velocity is much higher and heavier fluids are used, resulting in a turbulent flow. This is possible due to the difference in geology, such speeds and volume would erode weak and loosely packed soil and cause instability in the borehole but in rock or within a casing this is feasible. Due to the depth the pressures and temperatures are different as well, resulting in different downhole behavior of the drilling fluid. The differences extend beyond the physics and dimensions. For example, there is a difference in definitions as well. In the Marsdiep project the maximum angle of inclination of the borehole with the horizontal is about 12 degrees whereas a major part in directional drilling is much steeper. This results in a different definition for the angle of inclination; the angle between the boring and the horizontal in HDD and the angle with the vertical in DD. This varies between companies as well, the angle of inclination is differently defined by the surveyors used in Texel and Den Helder.
Marsdiep Project

The current pipeline is constructed in a shallow trench in the seabed and crosses a tidal gully. It is subject to scour and less erosive processes that all result in considerable maintenance costs. This is one of the reasons the proposed construction method for the new pipeline(s) is Horizontal Directional Drilling (HDD). If HDD is used for this project it will be one of, if not, the longest pipeline built using this technique to date.

Deltares has been hired to investigate the feasibility of the project and give advice on the most feasible depth and diameter of the pipe. This includes soil investigations but also being present when the pilot drillings are executed, taking samples and evaluating the results afterwards.

In this chapter the characteristics of the Marsdiep project are presented, both the results of the preliminary design calculations as well as the Site Investigations (SIs) conducted (Mesdag et al., 2014; Nugroho and van Meerten, 2015; Post, 2015).

3.1. Locations and Dimensions

In this section the locations and dimension of the proposed alignments as well as the executed initial SIs are presented.

3.1.1. Initial Cross-section

The initial cross-section is presented in figure 3.1. This cross-section is based on the results of the initial SIs.

Figure 3.1: Cross-section generated in the program DGEOpipeline, based on the results of Cone Penetration Tests (CPTs) and a seismic survey (Nugroho and van Meerten, 2015), with Texel on the left and Den Helder on the right side. The ‘_t’ and ‘_d’ in the figure correspond to identified in respectively Texel and Den Helder. In table 2.1 and table A.2 the abbreviations are elaborated on.
3.1.2. **Site Investigations**
In both Texel and Den Helder onshore SIs are conducted. The initial SI consists of boreholes and CPTs. The positions of which is presented in figure 3.2 and the coordinates and more information with regards to the initial SI can be found in appendix B.1. Figure 3.3 shows the results derived from the seismic offshore survey and figure 3.1 shows the resulting cross-section based on the seismic survey as well as the CPTs. This cross-section is also used in the calculations. The soil parameters used for the calculations are derived using the CPT and borehole data as well as the NEN 997-1 table 2b. These parameters can be found in appendix ??.

![Figure 3.2: The locations of the conducted onshore SIs, Den Helder (left) and Texel (right) (Nugroho and van Meerten, 2015). The locations marked with a ‘B’ are the boreholes and those marked with ‘DKMP’ are CPT data points.](image)
Figure 3.3: The cross-section as a result of the seismic survey (Mesdag et al., 2014).
3.2. Calculations
Making use of, amongst others, the program DGeoPipeline and the cross-section presented in figure 3.1 a number of preliminary calculations are made during the initial research into the feasibility of the project. The calculations range from required pressures and limits with regards to fracturing and blowout to required push and pull forces. Only the results of those calculations relevant for the pilot drilling are presented in appendix B.

3.3. Alignment
In figure 3.4 the alignment of the planned pilot drillings is presented. The pilot drilling alignment is close to but does not cross the proposed alignments for the pipeline. This reduces the risk of interaction between the new and old boreholes, which can cause unpredictable problems but the encountered soil during the pilot drillings is similar to the soil during the construction of the pipeline.

Figure 3.4: The alignment of the Marsdiep project (Post, 2015).
4

Pilot Drilling Descriptions

In this chapter the details of the pilot drillings are discussed. The source of the information is, in most cases a report about the setup of the pilot drillings (Post, 2015). Information regarding the execution of the plan is gathered on site during the pilot drillings.

The primary goal of the pilot drillings is to determine the feasibility of construction of a pipeline between Texel and Den Helder using Horizontal Directional Drilling (HDD), despite the lack of specific data about the local geology and properties of the layers. A secondary goal is to gain knowledge and experience that will help with the design and execution of the realization of the pipeline through the use of the HDD method. The information required for the definitive design is gathered efficiently at location with the pilot drillings. The final design will implement necessary measures to reduce the risks identified during the tests. The main contractors for Texel and Den Helder are respectively A.Hak-Drillcon, a Dutch company and LMR, a German company.

4.1. Site Setup and Data Acquisition

In figure 4.1a and figure 4.1b the site setup of respectively Texel and Den Helder are depicted in a schematic way. Included are the in- and output of the different parts, which can vary in time. For example the input at the mixing unit is highly variable depending on the soil, groundwater, efficiency of the recycling unit etc.

An attempt is made to construct a fitting mass balance but this is hindered by numerous processes and missing information. The data available for this purpose are manual read outs of flow meters on the pipes of the in- and outflow. The outflow is measured on a pipe coming from the mud pit. The fluid level in the mud pit below the casing varied. This caused a difference that is in the order of or larger than the losses occurring. This causes the calculated loss to be negative at times. A negative loss would mean leakage from the surrounding groundwater into the borehole. No evidence of this is found in the properties of the mud coming out of the borehole, neither loss of properties due to dilution or abnormal increase in salt content, measured indirectly through the electro conductivity. A small constant increase is present due to salt groundwater in the drilled soil, that is not filtered out during recycling so the salinity of the inflow slowly increases.

An overview of the supplied data is inconvenienced by the amount of different sources, parameters, definitions and formats used. Coupled with the different measuring frequencies and a difference in time between the sources of data this can cause apparent contradictions and uncertainties. For both locations the supplied data can be divided into 4 sources:

- The ProData system; All data are measured at the surface and most of it on the hydraulics of the rig. Measurements are automated and this resulted in data, with a frequency of once every 3 seconds, concerning carriage speed and position (the part of the drilling rig that pushes on the Drillstring (DS)), pull and push, torque in both directions, rotation speed, pump pressure and pump rate.
- Survey data; acquired by multiple different companies and in one drilling by the drill company. The supplied data sets vary by company but the majority is the same and consists of data, once every forward going pipe of the DS, about the location of the drill head and the number of pipes (joint number).
- Driller log data; delivered by the driller. These data also contain the joint number and it is a continuous data set, so when pulling back, the joint number is recorded as well. Furthermore for every joint the push and pull, torque in both directions, drill time, rotation speed, pump rate and pump pressure are recorded. The main difference with the Survey data is that tripping back is included in the driller data sets. The main difference with the ProData system is that the data are estimations by the driller.
- Drill Company data; delivered by the drill company it contains a wide range of data with a varying degree of usefulness, due to irregular or insufficient measuring frequency. It contains among others mud data, photos and technical information of the used (drilling) equipment and data on the usage of materials.
(a) A flowchart of the site setup during the drilling in Texel.

(b) A flowchart of the site setup during the drilling in Den Helder.

Figure 4.1: Site setups.
4.2. Description of the Pilot Drillings

An overview of the realized drilled paths can be found in figure 4.2 and in table 4.1 the planned and realized paths is presented. Below the different alignments are explained as executed, the proposals can be found in appendix B. The horizontal alignment is presented in figure 3.4 and is exemplified in section 3.3.

![Figure 4.2: Cross-section with the realized drill paths, with Texel of the left and Den Helder on the right side.](image)

<table>
<thead>
<tr>
<th>Planned</th>
<th>Realized</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Location</strong></td>
<td><strong>Code</strong></td>
</tr>
<tr>
<td>Texel</td>
<td>PB1T-B</td>
</tr>
<tr>
<td></td>
<td>PB1T-C1</td>
</tr>
<tr>
<td>Den Helder</td>
<td>PB1DH-A1</td>
</tr>
<tr>
<td></td>
<td>PB1DH-A3</td>
</tr>
</tbody>
</table>

4.2.1. Tests

Two types of tests are used to determine different aspects of the drilling processes. To determine the gel-strength of the drilling fluid and the stability of the borehole a stop test is done and to determine the average diameter of the borehole a pill test is done.

**Stop Test**

The stop test is performed once in Texel and twice in Den Helder. It is executed after reaching the agreed end of the drilling. The drilling is paused for a number of days and afterwards the conditions in the borehole are tested to see what the effect of a stop is on borehole stability. When starting after the stop, the pumps are started first to check if the return flow would start and to test the gel-strength of the drilling fluid. If the pressure generated by the start of the pumps is insufficient to break the gel-strength of the fluid there are no returns. If this happens the Drillstring (DS) is rotated to break the gel-strength of the drilling fluid. The order of starting is the opposite of normal practice after a break. The start of the pumps before rotation of the DS causes increased annular pressure with an increased risk of fracturing or blowouts. The stability of the borehole is determined while pulling the drill back to the surface from the build-up of the required pulling force.
Pill Test
This test is performed twice in Den Helder. After a certain volume, a pill of a more viscous fluid is pumped down, regular drilling fluid is pumped again. The time till the more viscous fluid reaches the surface is monitored so the average diameter of the borehole can be calculated. The regular fluid is pumped down to prevent distortion of the results of the stop test (since it would deviate from the reality), which is done after the pill test.

4.2.2. Texel
This project is different than regular projects in the HDD industry and this allowed the contractor in Texel to take a different approach as well. A mud pulse is initially used to send the data from the drill bit to the surface, see appendix C.2.1 for more information. After a few days filled with startup issues related to this and other systems provided by a subcontractor, the company is replaced. The company responsible for the mud pulse system had a gamma ray device installed which is to be read out at the end as well.

The PB1T-B drilling is executed first at a depth of between NAP-65m and NAP-70m, followed by the PB1T-C1 Drilling at a depth of NAP-85m. The drillings are executed in shifts, 24 hours a day.

The drilling fluid is created using OCMA-grade Bentonite.

PB1T-B
Due to difficulties in steering in the layer with a jet bit, the first attempt build up a lot of friction and had to be aborted at about 837m. A second attempt is made at another depth. In order to accomplish this without additional friction the DS is pulled back to the corner before continuing forward while letting the jet bit fall down until a more stable layer is reached. This creates one corner with a different radius than proposed, see figure 4.2 but did not cause a significant increase of the friction on the DS. The new depth is around NAP-70m and the drilling is aborted after about 1700m, 178 joints, of the proposed 2000m have been drilled. Drilling further is not possible according to the contractor (van Meerten and Nugroho, 2015). In table 4.2 the observations from the drillers is presented around that length. Due to the aborted end a stop test is not executed.

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Joint</th>
<th>Activities and Observations</th>
</tr>
</thead>
</table>
| 26-Jul | 21:08 | 176 | Loss, visual estimate 50%.
|       | 23:30 | 176 | Tripping back in order to try and restore return flow. |
|       | 03:13 | 166 | High torque and pull during tripping back. When rotating and pushing an increase in return flow is observed. |
|       | 07:00 | 176 | Back at face. |
| 27-Jul | 10:16 | 167 | 1 mm diameter sand on sieve, inflow is too thin. |
|       | 10:35 | 169 | Back flow observed. |
|       | 11:35 | 176 | Back flow stops, less progress afterwards. |
|       | 13:44 | 177 | Disturbances in data transfer, tripping back to joint 139 (bleeder sub). |
|       | 15:41 | 140 | Tripping in. |
|       | 19:00 | 176 | On face, driller requests Barazan to counter the loss of fluid. |
|       | 00:00 | 176 | Inflow 1200 L/min, for 2 hours. |
| 28-Jul | 02:30 | 178 | No progress, loss, visual estima 75%. |
|       | 04:00 | 178 | Bentonite supply exhausted, loss of communication. |
|       | 05:00 | 178 | Tripping back to surface. |

PB1T-C1
This boring is executed according to plan. The drilling encountered a strong clay layer for a length of about 10 drill pipes while still going down. At a length of 1501m and a depth of NAP-85m a stop test is executed, with a duration of two days.
4.2.3. Den Helder

In Den Helder both drillings are executed successfully. The PB1DH-A2 drilling is executed first at a depth of NAP-85m, followed by the PB1DH-A1 drilling at a depth of NAP-65m. Work takes place in 12 hour shift when drilling with a jet bit and continuously when the mud motor is used. The data from this drilling are reliable and complete almost as soon as the drilling finished, due to daily delivery of almost all of the data gathered that day.

The drilling fluid created using Teqgel, which is a pre-mix of bentonite and certain additives to enhance the properties of the mixture.

PB1DH-A2

In Den Helder a bottoms up travel time, the time it takes for fluid from the drill head to reach the surface, is determined in a pill test at approximately 700m borehole length. The test is conducted while tripping back the DS, in order to change from a jet bit to the mud motor. The pill test is however not as easy or accurate as expected in this case, since during the tripping of the DS, large losses of drilling fluid are occurring of which the exact quantity and location are unknown. At the same time backflow occurred making calculation even more uncertain. This makes it very difficult or impossible to gain an accurate idea of the travel time due to the inability to accurately calculate the diameter of the borehole due to the uncertainties with regards to the volume of the drilling fluid. Another pill test is conducted after the end is reached with the mud motor. During this pill test the pumping continued unhindered by any other activity and thus the result is much more reliable.

A stop test of 2 days is done after reaching the end at 2000m length and at a depth of NAP -85m. The stop test went as expected. During tripping back the pull force required is higher than expected. The reason for this is expected to be a minor hole instability caused by the abnormal high mud pressures compared to normal practice.

PB1DH-A1

This drilling is executed with just a jet bit. A pill test is conducted at the end before the stop test. The stop test is executed with a duration of approximately 3.5 days after reaching the end at 1000m and a depth of NAP -65m. The stop test caused no problems at startup afterwards.

4.3. Sampling Method

Samples are taken once every hour and analyzed once every two hours. The samples that are not tested on site are put in a cooling container and transported to Delft to be tested afterwards if required. The method of sample taking is with a bucket on a stick through a hole in the casing in Texel, see figure 4.3a and at the end of the casing in Den Helder, see figure 4.3b. Both methods however resulted in mixing of the drilling fluid at the end. If a moving bed flow is present in the borehole, no evidence of this is seen at the surface. Although no significant difference is visible in the flow, samples are taken by gathering fluid from the whole width of the flow.

(a) Sample location in Texel.

(b) Sample location in Den Helder.

Figure 4.3: The sample locations in Texel and Den Helder.
4.4. Suitability of HDD for SI

The drillings in both locations resulted in a collection of data. The data from the second location is more useful and complete due to the lessons learned from the first drilling. However improvements are possible. One of the main requirements for reliable Site Investigations (SIs) is knowing where the samples are coming from. This can be calculated using basic formulas and a number of assumptions but tests, such as the pill test, can and should be used to correct the calculations.

For both companies the pilot drilling is unlike a regular one and the difference in how they handled this is remarkable. In both cases the amount of data gathered is significantly more than the companies are used to as well as that the data are studied more closely than the companies are used to. From the results of a regular HDD project the source location of the samples, as calculated in this research, could be calculated. For example the pump rates and pump time, as measured by the Prodata system, could be reconstructed using the driller logs, which are part of a regular HDD project, but it is less accurate that way. The resulting locations would be less accurate as well and it requires a number of assumptions, for example the diameter of the borehole, for which an indication is acquired in Den Helder using the pill test, and the loss of drilling fluid.

The difference between theory and practice is substantial. Numerous processes are known to occur but no information about the significance or the magnitude of these processes is present or used on site. In this line of work, these data are apparently not always required to successfully complete projects.
Processes

In order to be able to calculate the sample source location it is important to know what can happen in the borehole. First in this chapter is a brief literature overview, followed by suitable fluid models and it ends with a brief overview of the most important flow and transport processes related to Horizontal Directional Drilling (HDD) projects. A complete overview would be too extensive so a selection is made that focuses on processes influencing the flow and the transport capacity in the pilot phase.

5.1. Literature Overview

Presenting a brief overview of what processes take place in the borehole is complex, since there are multiple approaches to find the answer to what happens in the borehole and why: some papers describe two fluid phases (Espinosa-Paredes et al., 2007; Ofei et al., 2014; Sun et al., 2014), some include three fluid phases (Kelessidis and Bandelis, 2004; Duan et al., 2008; Ozbayoglu et al., 2009). There are approaches taking the angle of the borehole into account and one such approach identifies 7 different flow and transport mechanisms (Ramadan et al., 2005) and there is an approach that divides the borehole trajectory in 3 in addition to 3 fluids in the borehole (Cheng and Wang, 2008).

What rheologic model is best suited to describe the fluid behavior is a question with no definitive answer yet. Many drilling fluids exhibit both yield stress and shear-thinning behavior. To reproduce these effects a three-parameter rheology model is required. The three parameters are: yield stress, consistency index and flow index. An evaluation is made between models that are used in the drilling industry: Robertson-Stiff, Collins-Graves, Herschel-Bulkley (HB) and a range of other models by (Weir and Bailey, 1996; Bailey and Weir, 1998) as cited in (Gjerstad and Time, 2014). In more recent times however the HB model is more used and described by among others (Talmon and Huisman, 2005) but it is also recommended by the American Petroleum Institute (API, 2010) as quoted in (Gjerstad and Time, 2014). But more complex models are used as well, for example (Zhou et al., 2013) proposes a 4 parameter model, depending on yield stress, 2 viscosity indexes and a flow index.

The amount of research conducted shows that a solution is still not found, while the amount of research conducted in the petroleum industry with regards to Directional Drilling (DD) dwarfs the amount of research conducted within the HDD industry (Baumert et al., 2005). What is making a comparison difficult is that the definitions (ie. inclination or depth) and specifics (ie. temperature or pressure) are often so different that it is not applicable in the other research field. The specifics are not always as well reported as is preferable and therefore it remains unclear whether certain research can be used.

5.2. Different Fluid Models

One of the main factors that determine whether a HDD project succeeds, is the selection of a proper drilling fluid and its behavior in the borehole and related to that, the transport capacity of the fluid. In order to identify the behavior of the fluid a description model has to be selected first.

A model often used to describe the rheology of the drilling fluid in HDD is the Bingham Plastic Model (Baumert et al., 2005), which can be written in the following way:

\[ \tau = \tau_y + \mu \cdot \gamma \]

where:

- \( \tau \) = the shear stress in [Pa]
- \( \tau_y \) = the yield stress in [Pa], which for a Bingham (BH) fluid is equal to the yield point, YP
- \( \mu \) = the plastic viscosity in [Pa.s], for BH fluids also written as PV
- \( \gamma \) = the shear rate in [1/s]
The Power Law rheological model assumes no yield stress and as such the shear stress is defined by:
\[ \tau = K \gamma^n \]  
(5.2)
Where \( K \) is the flow consistency index [Pa.s] and \( n \) is the flow index [-]. But while the model is sufficient with high shear rates it is not as accurate for low shear rates (Hemphill et al., 1993) as cited in (Baumert et al., 2005).

The Herschel-Bulkley or yield-power law model, is a modified version of the Power Law rheological model and is defined by:
\[ \tau = \tau_y + K \gamma^n \]  
(5.3)

Most drilling fluids show behavior that cannot be accurately described by the Bingham model for fluids. While the model is sufficiently accurate in the higher shear rate range (300-600 rpm) it is often inaccurate in the lower shear rate ranges, see figure 5.1, which are of interest for laminar, annular flow behavior (Baumert et al., 2005).

The actual shear rate can be a criterion on which a selection is made between the mentioned fluid models. The average shear rate in an annulus can be calculated using the following formula from (Denekamp et al., 2000).
\[ \gamma_{rf} = R_{fac} \frac{8 \times v_{rf}}{D_{BH}} \]  
(5.4)
where:
- \( \gamma_{rf} \) = the average shear rate of the return flow in [1/s]
- \( R_{fac} \) = a factor for the hydraulic radius, which is 1.5 for an annulus
- \( v_{rf} \) = the average velocity in [m/s]
- \( D_{BH} \) = the diameter of the borehole in [m]

Another formula for the average shear rate in an annulus is (Mendes and Naccache, 2005):
\[ \gamma_{rf} = \frac{v_{rf}}{(R_o - R_i)} \]  
(5.5)
Where \( R_o \) is the radius of the borehole in [m] and \( R_i \) is the radius of the Drillstring (DS) in [m].

### 5.2.1. Determination of Bingham Parameters

The values for the Bingham parameters are derived from the viscometer results as follows:
\[ \mu = \frac{(RO_{600RPM} - RO_{300RPM}) \times CF1}{(600 - 300) \times CF2} \]  
(5.6)
where:
- \( RO_{600RPM} \) = the value of the viscometer at 600 Rotations Per Minute (RPM) [°]
- \( RO_{300RPM} \) = the value of the viscometer at 300 RPM [°]
- \( CF1 \) = a conversion factor used for converting the readout value from [°] to [mPa]
- \( CF2 \) = a conversion factor used for converting the setting of the device to shear rate, [-] to [1/s]

\( CF1 \) and \( (600 - 300) \times CF2 \) are roughly the same value, respectively 511 and 510, so just \( RO_{600RPM} - RO_{300RPM} \) could be used without causing a large deviation. The \( YP \) is calculated as follows:
\[ YP = RO_{300RPM} - ((RO_{600RPM} - RO_{300RPM}) \times CF1) \]  
(5.7)
Only if the plastic viscosity is a constant the yield point, \( YP \), is equal to the yield stress, \( \tau_y \).

### 5.2.2. Determination of Herschel-Bulkley Parameters

Multiple methods to determine the parameters of the HB model exist. Two methods are discussed.

The Klotz and Brigham Method (Klotz and Brigham, 1998) that uses \( w_i \) as a weighting factor to account for the fact that the data gained at high speeds is more precise and reliable than the low-speed data. This method is meant to be used on data acquired with a 6-speed viscometer. The speeds are 3, 6, 100, 200, 300 and 600 RPM, which have to be multiplied by 1.703 to get the corresponding shear rate.
5.2. Different Fluid Models

\[
n = \left\{ \sum_{i=1}^{6} w_i \log(\dot{\gamma}_i) \right\} \sum_{i=1}^{6} \left[ w_i \log(\tau_{i,\text{data}} - \tau_y) \right] - \sum_{i=1}^{6} w_i \left( \sum_{i=1}^{6} w_i \log(\dot{\gamma}_i) \log(\tau_{i,\text{data}} - \tau_y) \right) + \left( \sum_{i=1}^{6} w_i \log(\dot{\gamma}_i) \right)^2 - \sum_{i=1}^{6} w_i \left( \sum_{i=1}^{6} w_i \log(\dot{\gamma}_i) \right)^2 \right\} \div \left( \sum_{i=1}^{6} w_i \log(\dot{\gamma}_i) \right)^2 - \sum_{i=1}^{6} w_i \left( \sum_{i=1}^{6} w_i \log(\dot{\gamma}_i) \right) (5.8)
\]

And

\[
\log(K) = \frac{\sum_{i=1}^{6} \left[ w_i \log(\tau_{i,\text{data}} - \tau_y) \right] - n \sum_{i=1}^{6} \left[ w_i \log(\dot{\gamma}_i) \right]}{\sum_{i=1}^{6} (w_i)} (5.9)
\]

where:
- \( w_i \) = a weighting factor (\( \tau_i - \tau_y \))
- \( \dot{\gamma}_i \) = the shear rate [1/s]
- \( \tau_{i,\text{data}} \) = the shear stress at the shear rate [Pa]

\( \tau_y \) is, in this case an initial estimation of the yield stress which through iteration is changed till an acceptable value is found for \( \Delta \) which is used as a measurement of error and is given by:

\[
\Delta = \sum_{i=1}^{6} (\tau_{i,\text{data}} - \tau_{i,\text{calc}})^2
\]

(5.10)

Where \( \tau_{i,\text{calc}} \) is the calculated shear stress at a given shear rate.

The Log Method is a more fundamental method, which is based on rewriting the basic HB model formula, (Denekamp et al., 2000; Huisman, 1999). When assuming \( \tau_y = \psi \tau_{3,\text{rpm}} \), where \( \psi \) is a scaling factor, formula 5.3 can be rewritten to:

\[
\tau - \tau_y = K(\gamma)^n = \log(\tau - \tau_y) = \log(K) + n \log(\gamma)
\]

(5.11)

Plotting \( \log(\tau - \tau_y) \) versus \( \log(\gamma) \) results in a linear graph in which \( \log(K) \) is given by the intersection between the graph and the y-axis and \( n \) is the slope or gradient of the graph. By varying the \( \psi \) -value the parameters can be fine tuned to find the solution with the smallest deviation from the measured data. Based on experience the value for \( \psi \) often is between 0.8 - 0.9 or 80 - 90% of the \( \tau_{3,\text{rpm}} \) (Denekamp et al., 2000).

5.2.3. Comparison Fluid Models and Determination Methods

An example from (Klotz and Brigham, 1998) is used in table 5.1 to compare the different methods and models. The Klotz and Brigham method, Log Method and Log Method, Scaled are all methods for the HB model. The Power Law is included since it is mentioned in section 5.2. The deviation, \( \Delta \), in the table is calculated using formula 5.10.

In this case 80% of the stress measured at 3 RPM proved the most accurate (\( \psi=0.8 \)). The same results are plotted in figure 5.1 for a clear overview. In the figure it can also be seen where the deviations are the largest. In the case of the power law for example it is sufficiently accurate at low shear but deviates significantly at higher shear, the Bingham approach shows the exact opposite. The method for determining the deviation fails to give an indication of where the method is the most accurate.

Since the scaled variant of the Log Method (Denekamp et al., 2000), is both less complex and more accurate it is the preferred method in this research.
Table 5.1: A table comparing the results and accuracy of different fluid models as well as two methods of acquiring the HB parameters, the data from mud model 3 from (Klotz and Brigham, 1998) is used.

<table>
<thead>
<tr>
<th>RPM</th>
<th>Dial Reading</th>
<th>Klotz and Brigham Method</th>
<th>Log Method</th>
<th>Log Method, (ψ=0.8)</th>
<th>Scaled</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>14</td>
<td>$\tau_y = 6.46$</td>
<td>$\tau_y = \tau_{3rpm}$</td>
<td>7.15</td>
<td>$\tau_y = \psi \cdot \tau_{3rpm}$</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>n = 0.552</td>
<td>n = 0.686</td>
<td>n = 0.522</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>K = 0.534</td>
<td>K = 0.217</td>
<td>K = 0.661</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>38</td>
<td>$\Delta = 1.173$</td>
<td>$\Delta = 4.278$</td>
<td></td>
<td>$\Delta = 0.356$</td>
</tr>
<tr>
<td>300</td>
<td>44</td>
<td>Bingham</td>
<td>Power Law</td>
<td></td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>60</td>
<td>$YP = 14.305$</td>
<td>n = 0.261</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$PV = 0.016$</td>
<td>K = 4.454</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\Delta = 94.997$</td>
<td>$\Delta = 105.670$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.1: Graph based on table 5.1. Showing different models to describe fluid behavior.
5.3. Flow of Drilling Fluids

The average velocity can be calculated with the following formula, (Baroid, 1997) as cited in (Baumert et al., 2005):

\[ v_{rf} = \frac{21.22 \times P_{Output}}{(D_{BH} - OD_{DS})} \]  

(5.12)

where:  
- \( v_{rf} \) = average fluid velocity in the borehole [m/s]  
- \( P_{Output} \) = the pump rate [L/min]  
- \( OD_{DS} \) = the outer diameter of the DS in [mm]

The pressure loss for Bingham fluids can be calculated using the following formulas, where the \( \text{efactor} \) corrects for possible eccentricity of the DS, see figure 5.2:

\[ \frac{dP}{dL} = \left[ \frac{47.88(PV \times v_{rf})}{D_{BH} - OD_{DS}^2} + \frac{6 \times YP}{(D_{BH} - OD_{DS})} \right] \times \text{efactor} \]  

(5.13)

where:
- \( dP/dL \) = pressure drop gradient [kPa/m]  
- \( \text{efactor} \) = eccentricity [-]  
- \( \delta \) = distance between center of borehole en center of DS [mm]

The following parameters are defined as characteristic for the flow behavior for a Bingham fluid (Bisschop, 1995):

**The Reynolds number** for Bingham fluids, which is defined as follows:

\[ Re = \frac{\rho_{rf} D_{BH} v_{rf}}{\mu} \]  

(5.14)

Where Re is the Reynolds number and \( \rho_{rf} \) is the fluid density. In general if \( Re \leq 2000 \) the flow is laminar. A generalized formula for the Reynolds number (Madlener et al., 2009), which can be applied to different fluid models is defined as follows:

\[ Re_{gen.full} = \frac{\rho_{rf} v_{r f}^2 n * D_{BH}^n}{\frac{1}{8} \times \left( \frac{D_{BH}}{v_{rf}} \right)^n + K \times \left( \frac{3 n + 1}{4 m} \right)^n \times 8^{-n-1} + \mu_{\infty} \frac{3 n + 1}{4 m} \times \left( \frac{D_{BH}}{v_{rf}} \right)^{n-1}} \]  

(5.15)

\[ m = \frac{n \times K \times (8 \times v_{r f}/D_{BH})^n + \mu_{\infty} \times (8 \times v_{r f}/D_{BH})}{\frac{1}{8} \times (v_{r f}/D_{BH})^n + \mu_{\infty} \times (v_{r f}/D_{BH}) \times \left( \frac{3 n + 1}{4 m} \right)^n} \]
This is based on the viscosity law for an extended version of the HB model, that adds an $\mu_\infty$ as a term for the constant viscosity at very high shear rates. The viscosity law for the BH model and the law for the HB model are different and the formula can be reduced accordingly. The viscosity law for the BH model is:

$$K = 0, n = 1 \rightarrow \mu = \frac{\tau_y}{\gamma}$$  \hspace{1cm} (5.16)

Where $\mu_\infty$ is constant in the BH model and is equal to the plastic viscosity, PV. The viscosity law for the HB model is:

$$\mu_\infty = 0 \rightarrow \mu = \frac{\tau_y}{\gamma} + K\gamma^{n-1}$$  \hspace{1cm} (5.17)

**The Hedstrom number**, which is defined as follows:

$$He = \frac{\rho_{rf}\tau_y D^2_{BH}}{\mu^2}$$  \hspace{1cm} (5.18)

**The settling velocity**, can be determined in a number of ways. In order for settling to occur, the grain has to surpass the yield stress. The maximum grain diameter, $d$, before the yield stress is exceeded can be found using multiple formulas, according to (Wan and Wang, 1994) as cited in (Bisschop, 1995):

$$d = \frac{16.5\tau_y}{g(\rho_s - \rho_{rf})}$$  \hspace{1cm} (5.19)

And according to (Slater, 1977) as cited in (Bisschop, 1995):

$$d = \frac{15.4\tau_y}{g(\rho_s - \rho_{rf})}$$  \hspace{1cm} (5.20)

Where $\rho_s$ is the grain density. Both equation 5.19 and equation 5.20 can be rewritten and are examples of a criterion originating in the plasticity theory:

$$\tau_y \leq a_{cr}(\rho_s - \rho_{rf})gd$$  \hspace{1cm} (5.21)

Where $a_{cr}$ is in a range between 0.048 - 0.200 according to the results in a literature study by (Chhabra, 1993) as cited in (Talmon and Huisman, 2005). For Bingham fluids the following formula can be used to calculate the settling velocity (Wan, 1985) as cited in (Bisschop, 1995):

$$v_{\text{settle}} = \frac{(\rho_s - \rho_{rf})gd^2}{18\mu} - \frac{7d\tau_y}{24\mu}$$  \hspace{1cm} (5.22)

This formula is similar to the Stokes equation only contains an additional term. Other formulas found for the settling speed in a Bingham fluid are often variations on the Stokes formula for laminar flow, for example the formula for moderate inertial effects presented by (Prashant and Derksen, 2011):

$$v_{\text{settle}} = \frac{(\rho_s - \rho_{rf})gd^2}{18\mu(1 + \sqrt{Re/9.06})^2}$$  \hspace{1cm} (5.23)

which is based on:

$$C_d = \frac{24}{Re} \left(1 + \frac{\sqrt{Re}}{9.06} \right)^2$$  \hspace{1cm} (5.24)

$$Re = \frac{\rho_{rf} U_c L_c}{\mu}$$

The settling velocity as found in (Talmon and Huisman, 2005) has a slightly different form:

$$v_{\text{settle}} = \frac{\alpha}{18} \frac{(\rho_s - \rho_{rf})gd^2}{\mu}$$  \hspace{1cm} (5.25)
where:
\[
\mu = \frac{\tau_y}{\dot{\gamma}} + K(\dot{\gamma})^{n-1}
\] (5.26)

And \(\alpha\) is an empirical coefficient to be determined from laboratory tests, in the paper \(\alpha = 0.5\) is used. The similarities with earlier formulas are remarkable since (Talmon and Huisman, 2005) consider a Herschel-Bulkley model instead of Bingham Plastic.

**The ratio between yield stress and shear stress due to friction,** which is defined as follows:
\[
\xi = \frac{\tau_y}{\tau_w}
\] (5.27)

Where \(\tau_w\) is the shear stress due to friction, which is calculated using:
\[
\tau_w = f \frac{\rho}{2} \rho f v_r^2
\] (5.28)

Where \(f\) is the Fannings-friction coefficient. Equation 5.27 can be used, for laminar flow, to determine the size of the plug, for turbulent flow it is assumed that no plug flow occurs.

**The Plasticity number,** defined as the ratio between Hedstrom and Reynolds:
\[
P_B = \frac{He}{Re} = \frac{8\xi}{1 - \frac{3}{4}\zeta + \frac{1}{4}\xi}
\] (5.29)

The relative settling time:
\[
t = \frac{t_k}{t_s}
\]
\[
t_k = \frac{D}{2u_{settle}}
\]
\[
t_s = \frac{L}{2u_{rf}}
\] (5.30)

where:
- \(t\) = relative settling time [-]
- \(t_k\) = average time a grain requires to settle
- \(t_s\) = average time a grain is in the borehole
- \(L\) = borehole length [m]

If \(t < 1\) settling has to be taken into account.

### 5.3.1. Infiltration Loss

Infiltration is a complex process but it is necessary to quantify in order to know the quantity of the return flow. In BTL-34 as cited in (Mastbergen et al., 1998; Denekamp et al., 2000; Mastbergen and Aanen, 2003) the following formulas can be found:
\[
y_a = \frac{\Delta P_{bh} \ast \alpha \ast D_p}{k \ast \tau_y}
\]
\[
\Delta Q_{it} = \pi \ast v_{drill} \ast (n_0 \ast (D_{BH} \ast y_a + y_a^2) + y_c \ast (D_{BH} - y_c))
\] (5.31)

where:
- \(y_a\) = the infiltration depth
- \(\Delta P_{bh}\) = the pressure difference between the borehole and the surrounding groundwater
- \(\alpha\) = a factor (value 8/75)
- \(D_p\) = the pore diameter (4/9 * \(D_{15}\))
- \(k\) = a blockage factor and is 1 for coarse sand/ fine gravel, 2 or 3 for fine sand
- \(\Delta Q_{it}\) = The volume that is lost due to infiltration
- \(v_{drill}\) = the drill velocity
- \(n_0\) = the porosity
- \(y_c\) = cake thickness
5.4. Transport

Different approaches exist to model the transport processes and the amount of transport processes identified. An extensive one by (Peden et al., 1990) presents a summary of earlier research:

- **Homogeneous Suspension**: sand is transported in suspension and distributed uniformly over the annular space.
- **Heterogeneous Suspension**: sand is transported in suspension but there is a concentration gradient across the annulus with more sand in the lower half of the annular space.
- **Suspension/Saltation or Saltation/Suspension**: sand is still transported in suspension but it is densely populated near the low-side wall so that it is virtually transported by jumping forward or saltating on the surface of the low-side wall. If suspension is dominant, this category is termed "Suspension/ Saltation" and vice versa.
- **Sand Clusters**: sand is transported in suspension, but is transported in clusters and all of the sand within each cluster is traveling with roughly the same velocity.
- **Separated Moving Beds (Dunes)**: separated sand beds are formed on the low-side wall of the annulus. The sand on the surface travels forward while the sand on the inside of the bed remains stationary, so that the beds look like they are rolling or sliding forward as a whole.
- **Continuous Moving Bed**: a thin, continuous sand bed is formed on the low-side wall of the annulus with the sand near the low side wall rolling or sliding forwards at a lower velocity than that above the bed.
- **Stationary Bed**: a continuous sand bed is formed on the low-side wall of the annulus with the sand on the surface of the bed rolling or sliding forward while the sand inside the bed is stationary.

The modeling of cuttings transport is extremely complex and no simple correlations exist for HB fluids (Bern et al., 2007). Methods recommended to the API with regard to transport in vertical and high-angle wells are based on Bingham fluids (Bern et al., 2007). For Bingham fluids 4 transport processes are identified (Bisschop, 1995):

- **‘No’ transport**: grains are too big or the flow velocity is too low thus no transport occurs, this is referred to as ‘Flow with stationary bed’ in Figure 5.3a.
- **Transport through occurrence of plug-flow**.
- **Bed transport**: transport along the bottom, also called saltation, see Figure 5.3b. The grains have a lower velocity than the drilling fluid.
- **Suspension transport**: grains are small enough or the flow velocity and the viscosity are high enough, the grains are transported as fast as the flow and the grains are evenly divided in the vertical direction, this is referred to as ‘Flow as homogeneous suspension’ in Figure 5.3a.
5.4. Transport

(a) Different transport processes, (Vanoni, 1977) as cited in (Bisschop, 1995).

(b) The saltation process, (van Rijn, 1987) as cited in (Bisschop, 1995).

Figure 5.3: Transport processes

Plugflow
In a moving fluid, with a yield stress, plug flow is assumed to occur. In figure 5.4a the geometry of plug flow is shown. An equation is given in (Kelessidis et al., 2006) for the $\tau_y$ as depicted in the figure. This formula can be rewritten to a function for the plug flow radius as follows:

$$\tau_y = \frac{r_p}{2} \frac{dP}{dL}$$

$$r_p = 2\tau_y \frac{dL}{dP}$$

(5.32)

Where $r_p$ is the radius of the plug flow.

For plugflow the transport capacity is relatively easy to calculate, with the following formula (Bisschop, 1995):

$$\Phi_{Tp} = \xi^2 c_v$$

(5.33)

where:
- $\Phi_{Tp} = $ the transport capacity
- $\xi = $ the part where plug flow occurs
- $c_v = $ the grain concentration

(a) The geometry of plug flow, in a pipe (Kelessidis et al., 2006).

(b) The geometry of plug flow, in a concentric annulus (Kelessidis et al., 2006).

Figure 5.4: Plug flow geometries
Bed transport
For the transport in bed transport the formula’s by (Graf, 1984) as cited in (Bisschop, 1995) can be used:

\[ \Phi_{TB} = 10.39 \psi_{T}^{-2.52} \]  

(5.34)

where:

\[ \psi_{T} = \frac{(\rho_{s} - \rho_{rf})d}{\rho_{rf}SR_{h}} \]  

(5.35)

The concentration which can be transported is:

\[ c_{v} = \frac{\Phi_{TB} \sqrt{(\rho_{s} - \rho_{rf}) \rho_{rf}gd}}{v_{rf}R_{h}} \]  

(5.36)

where:

\( \Phi_{TB} \) = transport capacity bed flow  
\( \psi_{T} \) = normalized shear stress  
\( S \) = pressure drop  
\( R_{h} \) = hydraulic radius

Bed transport to Suspension transport transition
Whether a grain is transported by the flow can be described by the following formulas (van Rijn, 1993) as cited in (Bisschop, 1995):

\[ \theta_{cr} = \begin{cases} 0.24d_{s}^{-1} & 1 < d_{s} \leq 4 \\ 0.14d_{s}^{-0.64} & 4 < d_{s} \leq 10 \\ 0.04d_{s}^{-0.1} & 10 < d_{s} \leq 20 \\ 0.013d_{s}^{0.29} & 20 < d_{s} \leq 150 \\ 0.055 & d_{s} > 150 \end{cases} \]  

(5.37)

where:

\[ \theta_{cr} = \frac{\tau_{w}}{(\rho_{s} - \rho_{rf})g} \]  

(5.38)

and:

\[ d_{s} = \left( \frac{\rho_{s}(\rho_{s} - \rho_{rf})g}{\mu^{2}} \right)^{1/3} \]  

(5.39)

However the resulting mode of transportation is bed transport, which has a high chance of sand accumulation. The ideal mode is suspension transport. The following formula, derived in (Hanks, 1980) as cited in (Bisschop, 1995) can be used to calculate the velocity required for suspension transport:

\[ v_{cr} = 1.32c_{v}^{0.186} \left[ 2gD \left( \frac{\rho_{s} - \rho_{rf}}{\rho_{rf}} \right) \right]^{1/2} \left( \frac{d}{D} \right)^{0.23} \]  

(5.40)

An upper limit with regards to concentration, \( c_{v} \), is not presented, although 0.15 - 0.2 is assumed as a safe value. Since \( v_{cr} \) is the lower boundary the left part could be rewritten to \( v \geq \ldots \). Rewriting that equation results in a formula for the maximum diameter of a grain for suspension transport based on the fluid velocity.

\[ d \leq D_{BH} \left( \frac{v_{rf} + \left[ 2gD_{BH} \left( \frac{\rho_{s} - \rho_{rf}}{\rho_{rf}} \right) \right]^{-1/2}}{1.32c_{v}^{0.186}} \right)^{1/0.23} \]  

(5.41)

Suspension transport
All of the dug soil is transported if the condition of equation 5.40 is fulfilled.
5.4.1. Transport Parameter

Another approach to see whether transport will take place can be found in BTL-42 as cited in (Denekamp et al., 2000). It is based on the grain size and the rheology of the drilling fluid.

\[ Y_g = \frac{3 \tau_y}{(\rho_s - \rho_{rf}) * g * D_{90}} \]  

(5.42)

Where \( Y_g \) is the transport parameter and \( D_{90} \) is the grain size at which 90% of the grains are smaller. Accretion of sand will take place for \( Y_g < 0.1 \). This is the theoretical value for which a grain in a non-Newtonian fluid will no longer settle and the settling speed is 0. There are however other factors that influence the settling speed that is why a \( Y_g > 2 \) to 4 is recommended. The formula can be rewritten to:

\[ D_{90} = \frac{3 \tau_y}{(\rho_s - \rho_{rf}) * g * Y_g} \]  

(5.43)

which can be used to calculate the maximum diameter that the fluid can transport based on a certain value for the transport parameter.

Within the scope of this research and the data gathered it is decided not to go into more detail with regards to the transport processes.

5.5. Lag Time or Source Location

The results of the previous sections can be combined to calculate the lag time or source location. This is necessary in order to connect the spoils of the return flow to a location in the subsurface. One such approach focuses on the creation of a Lag Diagram which plots the source location of the cuttings against the current position of the drill bit and an example is presented in figure 5.5. This approach uses the following assumptions (Garcia-Hernandez et al., 2008):

- Steady state condition in both flow rate and rate of penetration.
- No changes in the rheological properties of the fluid along the well.
- Circular wellbore geometry.
- No mass transfer between the combined phases and their surroundings.
- No heat transfer between the combined phases and their surroundings.
- No interruption of the drilling process (no connection time).
- Constant cuttings velocity in each segment of the well.
- Constant physical characteristics of cuttings (size, distribution and shape).

The approach also divides the geometry into three parts: vertical, inclined and horizontal. The vertical part is not discussed in this research since it is not applicable to this project. In equation 5.44 the formulas for the inclined part are presented.

\[ S_{DX(Incl.)} = \frac{V_{T(Incl.)} L_I \sin(\alpha_V)}{V_{T(Incl.)} + ROP} \]
\[ S_{DY(Incl.)} = \frac{V_{T(Incl.)} L_I \cos(\alpha_V)}{V_{T(Incl.)} + ROP} \]  

(5.44)

where:
- \( S_{DX(Incl.)} \) = is the x-coordinate on the inclined part
- \( V_{T(Incl.)} \) = is the cuttings traveling speed
- \( L_I \) = is the length of the inclined part
- \( \alpha_V \) = is the angle between the inclined part and the vertical
- \( ROP \) = is the average Rate Of Penetration (ROP)
- \( S_{DY(Incl.)} \) = is the y-coordinate on the inclined part
In equation 5.45 the formulas for the horizontal part can be found.

\[ S_{DX(Hor.)} = \frac{V_{T(Hor.)}L_H}{V_{T(Hor.)} + ROP} \]  

(5.45)

where:
- \( S_{DX(Hor.)} \) is the x-coordinate on the horizontal part of the boring
- \( V_{T(Hor.)} \) is the cuttings traveling speed
- \( L_H \) is the length of the horizontal part
- \( S_{DY(Hor.)} \) is constant

Figure 5.5: An example of a lag diagram, relating the current location of the drill head to the location of the source of the cuttings, (Garcia-Hernandez et al., 2008). ROP is the drill velocity, \( V_T \) is velocity of the cuttings and \( V_L \) is the fluid velocity.
In this chapter the methodology of the performed tests is elaborated on. As briefly mentioned in chapter 1 a goal of this thesis is to investigate if it is possible to connect the spoils of the return flow to a location in the ground. This is divided into two parts. The methodology of the actual data acquisition and tests performed on samples from the return fluid is presented first in a section on Equipment and tests. Secondly the programs that connect the samples to a source location are presented. These programs are explained further on in separate sections in more detail:

- SampleSource Program, main program, does the calculations.
- MarsdiepVisual Program, creates graphs for visualization of the measurements and the results of the main program.
- Fixes Program, deals with inconsistencies in the data and makes figures for the report that are not depended on samples and scenarios, eg. figure 6.1a.

6.1. Equipment and Tests

In this section the equipment and the performed tests are mentioned. For more details see appendix C, which contains photos and brief explanations of the equipment that is used. Not all of the described equipment is used in both Texel and Den Helder as well as the laboratory of Deltares.

<table>
<thead>
<tr>
<th>Equipment</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marsh Funnel</td>
<td>To get an indication of the viscosity.</td>
</tr>
<tr>
<td>Viscometer or Rheometer</td>
<td>A six-speed viscometer, to determine the rheological properties.</td>
</tr>
<tr>
<td>Mud Balance</td>
<td>Determine the density of both inflow and return flow.</td>
</tr>
<tr>
<td>Electrical Conductivity Meter</td>
<td>To get the electrical conductivity, for example an increase in salt content influences the rheological properties of the drilling fluid.</td>
</tr>
<tr>
<td>Sand Content Kit</td>
<td>Determine the sand content of the return flow.</td>
</tr>
<tr>
<td>Sieve Test (On Site)</td>
<td>To get an indication of the returns and to get clean fluid for the rheology test.</td>
</tr>
<tr>
<td>Sieve Test (Laboratory)</td>
<td>To determine the coarseness of the returns.</td>
</tr>
<tr>
<td>pH meter</td>
<td>To get the pH, for example a decrease in pH is often an indication for the presence of peat.</td>
</tr>
</tbody>
</table>

An adapted sieve test is performed on site, the test is adapted to the limits of the location and the available time. The test conducted to check visually for changes in the soil and the sieved fluid, is used in determining the rheology. Sieved fluid is required since the presence of grains can cause additional friction, distorting the results of the test and cause damage the viscometer. A standardized sieve test is conducted in the laboratory in Delft to determine the grain size distribution and to identify the formation of the samples.
6. Methodology

6.2. SampleSource Program

This program does the main calculations. For both drilling sites it can calculate for every sample, amongst others, the following:

- The location of the samples, based on different scenarios.
- The fluid properties for the Bingham and Herschel-Bulkley model.
- Which of those two models is the best fit to the experimental data.

A concise flowchart of the program can be found in figure 6.2. The full code can be found in appendix E.

6.2.1. Calculation of the Source Location (Intersection Method)

The Intersection Method, created for this research, is used to calculate the source location of the sample. The method is a fundamental one and appears easy to apply. It is foremost based on the pumped volume and the location of the drill head. For both sites the pump rate [l/min] and pump activity are known at time intervals of 3 seconds. With this and an estimated borehole diameter a length can be calculated, this length is the amount of space the pumped volume takes in the bored hole. Below the method is elaborated with a simplified example, the parameters are only used in this example.

**Example Calculation**

In figure 6.1a a schematic, simplified version is depicted of the Intersection Method with two different diameters but the same amount of pumped fluids. In figure 6.1b the drill head and fluid head locations are plotted.

At $T-0$ the sample is taken and the location of the drill head and the location of the sample are known. At $T-1$ the drill head is one pipe less far in the ground. The amount of fluid that is pumped down between $T-0$ and $T-1$ is calculated and divided by a cross-sectional area of the bored hole. The resulting length is the distance that the fluid head traveled in this time. Continuing to do this results for the case of Diameter 1 in an exact match at $T-2$. At this point the fluid and the drill head are at the same location, which is expected to be the source location of the sample.

Since diameter 2 is larger and the fluid rate is the same the distance the fluid head moves within the same time interval is less. This means, in this case, that the transport time is increased and the distance from the surface to the source location is decreased. It is also not an exact match.

In table 6.2 the values of the variables that are used in the figure 6.1a is presented.

<table>
<thead>
<tr>
<th>Diameter 1</th>
<th>Diameter 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-Sectional Area</td>
<td>1</td>
</tr>
<tr>
<td>Length Borehole</td>
<td>8</td>
</tr>
<tr>
<td>Drill Rate</td>
<td>1</td>
</tr>
<tr>
<td>Pump Rate</td>
<td>3</td>
</tr>
<tr>
<td>Fluid Head Rate</td>
<td>3</td>
</tr>
</tbody>
</table>

As is presented at $T-2$ the fluid head for diameter 1 and the drill head are located at the same location. Logically this is the source location of the sample taken. For diameter 2 there is not an exact match. After interpretation it can be found that at around 5.8m the locations are the same.
Complications
There are a number of factors that complicate the method as it is described above. The most important ones are listed below with a description and how they are handled.

Diameter
The diameter of the bored hole is unknown and although it can be estimated, it is likely highly variable due to amongst others the variation in the soil, pump rates or progress rate. Due to the scope of this thesis different diameters are looked at but it is assumed to be the same over the whole length of the borehole, see the next section. It is however recommended that for further research the influence of diameter on transport processes is taken into account.

Fluid loss
Unfortunately the fluid loss is also unknown. Despite the measurements taken on site the uncertainties remain significant. To handle this problem 3 different losses are considered. In reality this parameter will also be location dependent and it is recommended that it is further investigated to determine its influence. Furthermore, in Den Helder there is a loss of about 5 m³ in the casing, every night which means that probably the mud cake of a certain layer relatively close to the surface is not impermeable. For Texel the level of the drilling fluid in the casing is not checked and reported on, so it is unknown whether a loss occurred during breaks.

Geometry and Flow and Transport Processes
This factor is a combination of interacting processes and uncertainties. The borehole is assumed to be cylindrical in shape and the location of the Drillstring (DS) will have an influence on the behavior of the fluid down hole but not much more is definitely known. About the location of the DS downhole there are multiple ideas, ranging from corkscrew shape, while drilling, to below the borehole.

A another factor is the possibility of a bed forming in the borehole. When assuming a three phase system (from bottom to top; stable bed, moving bed and fluid) the cross-sectional area through which the fluid can freely flow, becomes severely limited as well as complex since the fluid might flow through the bed as well as at velocities significantly different from the flow through the top of the borehole. The velocity of the cuttings is in most cases not equal to the velocity of the fluid, the exception being suspension transport.

For now the cross-section of the borehole is assumed to be circular and the DS is located within the borehole, i.e. the cross-sectional area of the DS is subtracted from the cross-sectional area of the borehole. The other factors are not taken into account.

The Data or Execution in Practice
This last part relates to both the format of the acquired data as well as to the events that have taken place. The events range from repairs, blocks of missing data or just errors in the data, pulling back a few pipes when the system short circuit due to a damaged cable to other events that are not foreseeable but do influence the drilling process and add to the complexity of the data sets gathered on site. The time scale of the data related to the pumped volume, once every 3 seconds, is different from the time scale of the data related to the position of the drill bit, once every DS pipe, approx 10 min. There is not only a difference in frequency but also a difference of minutes is possible.

Exact matches are rare between the two sets of data. This is solved by taking a deviation in consideration and looking for the closest approximation. If it is not found the deviation increases and the match is sought again until a match is found.

6.2.2. Scenarios
In order to deal with the uncertainties with regards to the source location stemming from the uncertainties with regards to the method of transportation and the losses, different scenarios are taken into account. For two variables three options are investigated to get a range of values and thus to get an idea of the magnitude of uncertainty:

- Loss of drilling fluid; the values are the same for both sites.
  - 0%; no loss this is a theoretical upper limit for the volume of the return flow.
  - 7%; this value is based on the in- and outflow measurements taken on site. This is an average value of the overall difference between the cumulative in- and outflow, which are only available for Den Helder. An attempt is made to reconstruct it for Texel but due to different sources the resulting loss is negative and thus unrealistic.
• 30%; this is an assumption taken from the contractors and is used as an initial conservative value, this is used as the lower limit for the volume of the return flow.

• Borehole Diameter; the values are different for both sites. The borehole diameter is related to transport time since when the diameter goes up the velocity goes down and thus the transport time goes up as well, which means that the source location becomes closer to the surface.
  – $1.1 \times D_{\text{Drill}}$, this is an assumption based on the minimum a drill head removes when passing through the ground. When it rotates or pauses for moment the diameter will already be bigger. This is a conservative value and as such the velocities in the borehole are likely to be lower and thus the transport time longer.
  – 0.45m, this is the result from tests conducted with a thicker fluid, a pill, in Den Helder. Such a test is only done in Den Helder so for Texel 0.4m is assumed since the equipment used in Texel is smaller than in Den Helder.
  – 0.8m, for Den Helder and 0.55m, for Texel. These diameters are based on calculations of the diameter derived from the measured sand content (Nugroho et al., 2016).

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Diameter Texel</th>
<th>Diameter Den Helder</th>
<th>Loss [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>$1.1 \times$ Diameter of the Drill</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.40</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.45</td>
<td>0.45</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0.55</td>
<td>0.80</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td>30</td>
</tr>
</tbody>
</table>
6.2. Sample Source Program

(a) A simplified version of the Intersection Method.

(b) A graphical representation of the head locations.

Figure 6.1: The Simplified Intersection Method.
Flowchart of the Sample Source program

- **Initial program setup**

- **At the start options are available. Depending on the what is required, a selection can be made of the scenarios and locations.**

- **Initialize Scenarios**
  - Open files and read files, assign values.

- **Preparations**
  - This calculates all that is not sample dependent, such as the geometry of the borehole.

- **Main Loop**
  - For every sample is checked if the rheology was determined and if this is the case both the Bingham and Herschel-Bulkley parameters are calculated.
  - Calculates the sample origin.
  - Write the results to files and or screen and plot the relevant graphs.

- **End Main Loop**
  - Plotting the graphs that depend on multiple samples. Writing the output files.

- **Progress Check**

- **Finishing the Program**

Figure 6.2: The flowchart for the SourceSample program.
6.3. MarsdiepVisual Program

This program is designed to create results not related to the scenarios and samples or depended on calculated data, i.e. measurements and sieve results. This program consists of multiple parts. A flowchart of the program is presented in figure 6.3. The full code can be found in appendix F. Parts 1, 2, 4 and 6 visualize results, initial or created in the SampleSource program. Part 3 deals with the data related to the sieve tests conducted on samples taken from the initial (vertical) boreholes and those on the samples acquired from the return flow in Texel and Den Helder during the (horizontal) pilot drillings. Part 5 merges different parameters from all scenarios into 1 file for easier evaluation.
Figure 6.3: The flowchart for the MarsdiepVisual program.
### 6.4. Fixes Program

This program consists of multiple parts as well but these are all small programs that only had to run once, when correctly written.

- **ProData file fix**: in Texel multiple pumps are present and for a short while a pump is attached that had the sensors but those are not calibrated or connected so a gap in the mudflow data exists. The gap is filled using data with a lower accuracy found in the driller logs.
- **Gefs**: this part reads the desired variables from all 69 gef files containing the sieve results and puts them in a .csv file. Thus collecting all required information in one file for easier continuation and removing the unnecessary variables.
- **Intersection Method figure**: this creates a figure that is used to clarify the Intersection Method, figure 6.1a.
- **Fluid Model figure**: this creates a figure, figure 5.1, that is used to compare the different fluid models and the different methods of determining the parameters.
Measurements

In this chapter characteristic results of measurements conducted on the project site and in the laboratory tests are presented.

7.1. Supplied Data

In this section examples of the supplied data are presented. A description of the different types can be found in section 4.1.

7.1.1. ProData System

In figure 7.1 a screen capture is presented of the resulting data from the ProData system. The system is, besides calibration, essentially the same in both Texel and Den Helder. From right to left the columns, in figure 7.1 are:

- TS (time stamp) in [days month year hours:minutes:seconds].
- BIT-DPT, this feature is not implemented.
- HOL-DPT, this feature is not implemented.
- CAR-POS (carriage position) in [mm].
- CAR-SPD (carriage speed) in [cm/min].
- PULL (pulling force) in [KN].
- PUSH (pushing force) in [KN].
- TORQ+ (the torque in clockwise direction) in [N.m].
- ROT-SPD (rotation speed) in [Rotations Per Minute (RPM)].
- M-PRESS (pump pressure) in [Bar].
- MudFlw IH (pump rate) in [L/min].
- TORQ- (the torque in anti-clockwise direction) in [N.m].
- PRESS-1, this feature is not implemented.
- PRESS-2, this feature is not implemented.
- PRESS-3, this feature is not implemented.
- HS-D, this feature is not implemented.
7. Measurements

7.1. Survey Data (Drilling Paths)

In figure 7.2 and figure 7.3 screen captures is presented of the survey data from respectively Texel and Den Helder.

The survey data are, in this research, mainly used to determine the drill location. In figure 7.4 and figure 7.5, the actual drilled paths in respectively Texel and Den Helder are depicted. With on the axis; Elevation in \([mNAP]\), Away, the horizontal distance, in \([m]\) and \(R_{\text{Calculated}}\) which is the horizontal deviation in \([m]\), with the positive value being a deviation to the right from the proposed path. The jet bit part of the NAP-85m drilling in Texel is missing so the beginning of the NAP-85m mud motor bit is used to recreate the drill location for the jet bit part. The data set used for the graph is used for the calculations but since it is the same, the mud motor line (green) covers the jet bit line (yellow) in figure 7.4.

The NAP-65m jet bit part in Den Helder is deliberately moved 2 meters to the right in order to create more distance between the boreholes at different depths. This as a safety measure against fracturing or blow-out to the other borehole and the loss of circulation that can cause. The horizontal deviation in Texel is about thrice the deviation in Den Helder.
### 7.1. Supplied Data

#### Figure 7.2: A screen capture of the survey data from Texel.

#### Figure 7.3: A screen capture of the survey data from Den Helder.
Figure 7.4: The drilled paths in Texel.
Figure 7.5: The drilled paths in Den Helder.
7.1.3. Driller Log

In figure 7.6 and figure 7.7 screen captures of the driller logs from respectively Texel and Den Helder is presented. In this research the data are used to check on the other data sets.

Figure 7.6: A screen capture of the driller log from Texel.

Figure 7.7: A screen capture of the driller log from Den Helder.

7.1.4. Drill Company Data

The drill companies also provided information about the equipment and the tests performed on fluid samples by them. Due to the frequency, uncertainties about the specifics and the difference in what is tested, the results of those tests are not taken into consideration for this research.
7.2. Acquired Data
This section presents the results of the laboratory tests conducted on the samples acquired during the pilot borings as well as the measurements taken in the laboratory of Deltares in Delft. An overview of the test can be found in section 6.1 and the tests are elaborated in appendix C.

7.2.1. On site
Here the results are presented of the tests conducted in the on site laboratory in Texel and Den Helder. This will mainly consist of the results of the rheological tests. A screen capture of a part of the data is shown in figure 7.9. Results of the sieve test performed on site is presented in figure 7.8. A description of this sample can be found in table 7.1. Additional observations are that the 0.6mm sieve contained less than the previous sample and that the 0.212mm and 0.063mm contained less silt-sized grains that had to be washed away after sieving since the silt grains blocked the flow through the 0.212mm and 0.063mm sieves.

Table 7.1: Descriptions of the sieve results depicted in figure 7.8. The quantity is based on how much of the sieve, area wise, could be covered with the quantity that is present on the sieve. The sample in the table is PB1DH-A3-85-69.

<table>
<thead>
<tr>
<th>Sieve Size [mm]</th>
<th>Quantity [%]</th>
<th>What</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.6</td>
<td>-</td>
<td>Pieces of wood</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>Pieces of wood</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>Pieces of wood, few sand grains</td>
</tr>
<tr>
<td>0.6</td>
<td>&lt;10</td>
<td>Sand (clear)</td>
</tr>
<tr>
<td>0.355</td>
<td>&lt;50</td>
<td>Sand (gray)</td>
</tr>
<tr>
<td>0.212</td>
<td>&lt;25</td>
<td>Sand (gray)</td>
</tr>
<tr>
<td>0.063</td>
<td>&lt;25</td>
<td>Sand (gray)</td>
</tr>
</tbody>
</table>

Figure 7.8: An example of the results from the sieve test conducted on site. The sample in the figure is PB1DH-A3-85-69.
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tijd monstername</td>
<td>datum en tijd</td>
<td>8-09-15 9:25</td>
<td>8-09-15 10:55</td>
<td>8-09-15 12:04</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Omschrijving boorproces</td>
<td></td>
<td>Caspo</td>
<td>1 boorbuiging voor de casing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Samenstelling boorvloeistof</td>
<td></td>
<td>Texgel Speciel 2,60m/l 68m3 tapv 3d (43.3) lkg/m3, Seda A.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Bemonstering</td>
<td>tvb</td>
<td>opslag</td>
<td>analyse</td>
<td>opslag</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>aantal stangen in het [no.]</td>
<td>0</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>monsternummer</td>
<td>analyse on site</td>
<td>PB1DH-A3-85-1</td>
<td>PB1DH-A3-85-2</td>
<td>PB1DH-A3-85-3</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Temperatuur [oC]</td>
<td>21.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Elektrische [mS/m]</td>
<td>1.97</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Chloride gehalte [mg/l]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>pH [-]</td>
<td>6.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Marsh-funnel viscositeit [sec/100]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Soortelijk gewicht [kg/m3]</td>
<td>12.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Zandgehalte [%]</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>gewicht boorvloeistof [gram]</td>
<td>5.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>2 [mm]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1 [mm]</td>
<td>vegetable</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0.6 [mm]</td>
<td>vegetable</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0.55 [mm]</td>
<td>vegetable</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0.212 [mm]</td>
<td>vegetable</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0.063 [mm]</td>
<td>vegetable</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>600 r.p.m.</td>
<td>92.30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>300 r.p.m.</td>
<td>59.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>200 r.p.m.</td>
<td>38.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>100 r.p.m.</td>
<td>26.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>50 r.p.m.</td>
<td>17.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>30 r.p.m.</td>
<td>14.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>3 r.p.m.</td>
<td>10.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>GS-10</td>
<td>11.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>GS-10m</td>
<td>24.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>grundbeschrijving en/of bijzonderheden</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>monsternummer</td>
<td></td>
<td>PB1DH-A3-85-1</td>
<td>PB1DH-A3-85-2</td>
<td>PB1DH-A3-85-3</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>tijd notite</td>
<td>Limiet afstand</td>
<td>08-09-15 10:55</td>
<td>08-09-15 10:55</td>
<td>08-09-15 12:04</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>Soortelijk gewicht [kg/m3]</td>
<td>11.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>MF</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>AV [mPa*s]</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>PV [mPa*s]</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>r-pv yield [Pa]</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>CS [Pa]</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>GS-10 [Pa]</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>GS-10m [Pa]</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Zandgehalte [bulk]</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 7.9: A screen capture of data gathered in Texel and Den Helder. This example comes from the Den Helder NAP-65m drilling. The unit of the viscometer values is degrees.
7.2.2. Deltares, Delft
A selection is made of the samples taken on site. This selection is made based on head location at sampling time, to get an even spread along the drilled paths. In the Deltares laboratory in Delft the samples are sieved. The sieve results are put in a database, which resulted in a standard report and a .gef file per sample. An example of the standard report is presented in figure 7.10.
Figure 7.10: An example of a report constructed for every sample. The sample in the report is PB1DH-A3-85-69.
In this chapter the interpretation and correlation of the measurements and data as described in the previous chapters as well as the results of the programs are presented.

8.1. Drilling Fluid

In this section the interpretation of results and measurements with relation to the fluid behavior and model are presented. All samples discussed here are taken from the return flow and a selection is made based on the drill bit location at sampling time to get an even spread of the fluid samples along the borehole.

8.1.1. Fluid Model

Although the Bingham (BH) model is widely used, it is not always the best model to describe the rheology of the samples from the return flow. As an alternative the Herschel-Bulkley (HB) model is used. Which of the fluid models describes the fluid behavior the best is not only dependent on the used base materials for the bentonite slurry but also on the chemistry of the soil. For example peat, encountered in Den Helder, often lowers the pH, which lowers the viscosity of the fluid. Additional materials can be added to counteract the decrease in pH, for example soda ash and fresh bentonite but those materials have an influence on other properties of the fluid for example the viscosity. Table 8.1 shows the results of determining which fluid model is the best fit to the experimental data, using formula 5.10.

<table>
<thead>
<tr>
<th>Model</th>
<th>Texel No. of Samples</th>
<th>Percentage</th>
<th>Den Helder No. of Samples</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bingham (BH)</td>
<td>21</td>
<td>81 %</td>
<td>11</td>
<td>46 %</td>
</tr>
<tr>
<td>Herschel-Bulkley (HB)</td>
<td>5</td>
<td>19 %</td>
<td>13</td>
<td>54 %</td>
</tr>
<tr>
<td>No fluid data</td>
<td>12</td>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td><strong>Total No. of Samples</strong></td>
<td><strong>38</strong></td>
<td></td>
<td><strong>31</strong></td>
<td></td>
</tr>
</tbody>
</table>

For Texel BH is the best fit for about 81% of the cases. The use of one model and that model being BH for all samples from Texel may be a logical choice. For Den Helder, where a different drilling fluid is used (see subsection 4.2.2 and subsection 4.2.3) BH is in about 46% of the cases the best fit, against 54% for the HB model. Which model is the best fit for the fluid samples from Den Helder, is unclear.

The variable behavior of the fluid in both sites makes it difficult to incorporate the behavior of the fluid in the calculations and makes most of the preliminary calculations less reliable. For example the value for the yield stress, for which a minimum value is prescribed, is higher when calculated according to the BH model than when calculated according to the HB model. An extreme example is presented in figure 8.1a. For most samples the results are similar to those displayed in figure 8.1b. For samples taken in Texel the HB model’s yield stress is between 48 and 97 % of the BH model’s yield point. For samples taken in Den Helder the range is larger. The HB value is between 11 and 90 % of the BH value. In figure 8.2 the variations of the yield point are presented. The line in both figure 8.2a and 8.2b is based on the yield stress of the models with the best fit.

Predictions or prescribed requirements may end up being invalid or unattainable since the ratios between different properties vary from fluid model to fluid model. For example the ratio between viscosity and yield
point is different for the drilling fluid in Texel and the drilling fluid in Den Helder. The same carrying capacity requires less annular pressure in Den Helder, so the risk of fracturing the soil or causing a blow out is lower.

![Fluid Model Comparison PB1DH-A3-85-02](image1)

(a) Fluid model data for sample PB1DH-A3-85-02.

![Fluid Model Comparison PB1DH-A3-85-75](image2)

(b) Fluid model data for sample PB1DH-A3-85-75.

Figure 8.1: Different examples of the fluid model data of a sample.
8.1. Drilling Fluid

(a) A plot of the yield stress for both the Bingham and Herschel-Bulkley model in Texel. The orange line indicates the yield stress of the model that is closest to the measured data.

(b) A plot of the yield stress for both the Bingham and Herschel-Bulkley model in Den Helder. The orange line indicates the yield stress of the model that is closest to the measured data.

Figure 8.2: The yield stresses of the fluid model with the best fit.
Weighting Factor

The formula used in calculating the fit (equation 5.10) has one disadvantage of particular interest for the purposes of this research. It fails in providing information regarding the fit in the area that is of concern. That is why, in this research, a weighting factor is added that is based on a calculated value for the shear rate. This is done to account for the accuracy of the Bingham model at high shear rates, this can cause a wrong result of which fluid model is most accurate at the shear rate regime in the borehole. In table 8.2 the result of shear rate calculations is presented. The results are averaged per scenario.

Table 8.2: Shear rates in the annulus, averaged for every scenario. Method 1 is referring to equation 5.4 and method 2 to equation 5.5. Both are used to calculate the shear rate in the borehole but the outcome is different.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Shear Rates, ( \gamma_{rf} ) [1/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Texel</td>
<td>Den Helder</td>
</tr>
<tr>
<td>1</td>
<td>7.34 2.83 8.87 2.91</td>
</tr>
<tr>
<td>2</td>
<td>6.83 2.63 8.24 2.70</td>
</tr>
<tr>
<td>3</td>
<td>5.10 1.96 6.19 2.03</td>
</tr>
<tr>
<td>4</td>
<td>4.56 1.47 3.88 1.03</td>
</tr>
<tr>
<td>5</td>
<td>4.24 1.37 3.61 0.96</td>
</tr>
<tr>
<td>6</td>
<td>3.19 1.03 2.70 0.65</td>
</tr>
<tr>
<td>7</td>
<td>1.75 0.45 0.68 0.14</td>
</tr>
<tr>
<td>8</td>
<td>1.62 0.42 0.63 0.13</td>
</tr>
<tr>
<td>9</td>
<td>1.21 0.31 0.47 0.10</td>
</tr>
</tbody>
</table>

As is presented in table 8.2 the shear rates are all low compared to the high shear rates (510 - 1020 1/s) in which the Bingham model is most accurate. When adding the weighting factor to formula 5.10 the formula for calculation of the deviation between the calculated and measured shear stress becomes:

\[
\Delta = \frac{1}{8} \sum_i \left( \frac{(\tau_{i,data} - \tau_{i,calc})^2}{(\gamma_i - \gamma_{rf})^2} \right)
\]

Table 8.3: Fluid model fits for both sites, taking a weighting factor into account as defined in formula 8.1.

<table>
<thead>
<tr>
<th>Model</th>
<th>Texel No. of Samples</th>
<th>Texel Percentage</th>
<th>Den Helder No. of Samples</th>
<th>Den Helder Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bingham (BH)</td>
<td>17</td>
<td>65%</td>
<td>8</td>
<td>33%</td>
</tr>
<tr>
<td>Herschel-Bulkley (HB)</td>
<td>9</td>
<td>35%</td>
<td>16</td>
<td>67%</td>
</tr>
<tr>
<td>No fluid data</td>
<td>12</td>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Total No. of Samples</td>
<td>38</td>
<td></td>
<td>31</td>
<td></td>
</tr>
</tbody>
</table>

As taking a weighting factor into account for the samples from Texel the BH model is the best fit in about 65% of the cases. The use of only one model and that model being BH may not be such a logical choice. When taking a weighting factor into account for the samples from Den Helder the BH model is the best fit in about 33% of the cases. Choosing only one model may, in this case, not be a logical choice. This seems like a significant change compared to using no weighting factor but the amount of samples that changed is still small, therefore more research is recommended. The same graphs as depicted in figure 8.2 is created for both weighting factor 1 and weighting factor 2. The difference is minimal due to the amount of data points. The graphs can be found in appendix D.3.

It has to be mentioned that all the samples have been tested only once and it is not known how large the variation is which may occur as a result of the measurement equipment and the differences in the mixture itself. The measurement at 3 Rotations Per Minute (RPM) is the most uncertain due to the possibility of gel starting...
to form. Due to the low shear rate in the borehole the 3 RPM measurement becomes more significant when applying the weighting factor.

8.1.2. Time Dependent Behavior
More differences than mentioned above can be found between the drilling fluids. These differences may have an influence on the behavior and properties of the drilling fluid. A bleeding test is performed by saving a sample of the drilling fluid in Den Helder in a cylinder. Regular bleeding tests last 24 hours but since the stop test and possible standstill in final construction can have a longer duration the test is continued for a week. This test is performed to check the stability of the drilling fluid and to find out whether separation between solid and liquid phase will occur.

In figure 8.3 the results is presented. The first few days the fluid started to separate (right top picture). There is a small layer of water forming on top of the sample. The sample also increased in volume since the cylinder is filled to the 1000 ml mark at the start of the test. After 4.5 days the fluid is increased to such an extent that the water spilled over the top of the cylinder, no drilling fluid is lost. After a week the volume is decreased again though not as far back as the initial volume. For stops of limited time, for example during the night, these processes are not as significant as for stops of a longer duration.

These observations are made in static conditions, at atmospheric pressure and at the surface, major differences with the borehole are higher temperature and contact with air. The observed behavior might not occur or be different in magnitude or time scale in the borehole. However this means that a difference in fluid characteristics does not only depend on mixing and the soil and groundwater chemistry. It can also depend on time related processes that may occur in the drilling fluid. These processes might interact with the soil and groundwater. Additional research is required to determine the when these processes occur, the magnitude and subsequently the significance of the processes.

Figure 8.3: The long-time behavior of a sample from the drilling fluid used in Den Helder. This behavior includes swelling/shrinking and separation of phases. The red circle encloses the same crack in the fluid, containing gas and widening till day 4.5 and decreasing in size afterwards.
8.2. Transport and Flow Processes
The processes in this section are investigated in addition to the location determination using the Intersection Method.

8.2.1. Flow Type
The Reynolds number can be used to determine whether the flow is laminar or turbulent. Using equation 5.15 the Reynolds number for all samples, every scenario and both the BH and HB model are calculated. The Reynolds numbers for both fluid models are of the same order and laminar flow is suspected to have occurred since the Reynolds number does not cross the threshold value, 2000, for the transition to turbulent flow. In fact the maximum value remains beneath 5% of that value. The full results can be found in appendix D.2.

8.2.2. Infiltration Loss
The expected infiltration loss for every sample is calculated. Due to uncertainties with regard to the reliability of the results, for example the exact location of where the loss occurs is uncertain. Furthermore a loss of about 5 m³ for every 12 hour break, during the nights and after the stop test, remains unexplained. which is not taken into account by equation 5.31. The \( D_{15} \) that is used to determine the pore diameter in equation 5.31 requires the results of additional Site Investigation (SI) since the \( D_{15} \) of the sample from the return flow can be less than that of the soil at source location, which would lead to an underestimation of the loss, which would influence the calculated location of the loss and subsequently lead to a different sample source location for that sample and a number of earlier samples. Instead of adding the uncertainties described above to the results of the source location calculations, the loss of drilling fluid is a variable in the different scenarios (see section 6.2).

8.2.3. Transport
For the transport parameter, \( Y_g \), equation 5.42, 2 and 4 are recommended as values to prevent accretion of sand in the borehole. The formula can be rewritten to estimate the \( D_{90} \) the fluid can transport without accretion given the transport parameter is a certain value.

When the measured \( D_{90} \) of the sample is lower than the calculated value, \( D_{Calc} \) this means that the coarser material is not present at the source location otherwise it would have been transported. The transport is sufficient and this is most often the case.

In a few cases the \( D_{90} \) of the sample is larger than the \( D_{Calc} \) for transport. In these cases it is unknown whether the transport capacity is sufficient since coarser material than calculated left the borehole and thus the value for the transport parameter is actually higher than 2 or 4. But even coarser material could have been left in the borehole, forming a bed. These cases are counted as insufficient transport capacity.

In table 8.4 a summary is presented of the results. The results are presented in appendix D.1.

<table>
<thead>
<tr>
<th>Fluid Model</th>
<th>Texel</th>
<th>Den Helder</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y_g = 2 ) BH</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>( Y_g = 2 ) HB</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>( Y_g = 4 ) BH</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>( Y_g = 4 ) HB</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>Sufficient (( D_{90} &lt; D_{Calc} ))</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Insufficient (( D_{90} &gt; D_{Calc} ))</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>No Fluid Data</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Total No. of Samples</td>
<td>38</td>
<td>38</td>
</tr>
</tbody>
</table>

In table 8.5 the \( D_{90} \) of the samples taken from the vertical drillings during the initial SI are presented as well as the \( D_{90} \) from the samples taken from the return flow of the horizontal drilling. From this table it becomes clear that a static bed is formed in the borehole since the \( D_{90} \) of the vertical drilling is significantly higher than the \( D_{90} \) of the horizontal drillings for the same formation, which means that material remained in the borehole. For example for the Peelo Formation the difference is about a factor 2. For the Urk and Appelscha Formation the difference is significant as well. An exception is the \( D_{90} \) of the Eem Formation on the Texel side. The \( D_{90} \) of the sample from the return flow is significantly higher than the \( D_{90} \) of the samples from the vertical drilling. This is due to the samples that are located at the bottom of the filled valley, as indicated by the results of the seismic
8.3. Sample Identification

The samples taken from the return flow are identified by Freek Busschers and Jeroen Schokker, both geologists working for TNO. The samples are sorted into groups (figure 8.4) and the groups are identified and defined based on the samples and the results of the initial SIs (table 8.6). The inclusion of the results of the initial SI in the determination led to the inclusion of group 6 which is not identified in the samples from the pilot drillings.

The formation to which the samples belong are identified based on multiple factors such as coarseness, presence of micas and colored sand. No conflicting indicators of formations are found within one sample and as such the samples are identifiable as coming from one formation. This implicates that no (significant) mixing has occurred in the borehole. This means that no separation based on grain size has taken place (besides settling of the coarsest grains) or at least not on a recognizable scale but also that the drilling fluid did not cause significant wall erosion. Another implication is that a moving bed reaching the surface is less likely than a static bed.

Combining this with the results of section 8.2 indicates that the fluid model is of significantly less influence in this specific case than previously assumed. Since no significant mixing occurs plug flow is the likely mode of transport. Depending on the size of the sheared portion of the fluid the velocity of the plug is close to the average fluid velocity in the borehole. For this research this means that the source can be determined with accuracy without taking a fluid model into account. The calculation related to the drilling fluid only prove that this is possible and are not required to calculate the difference between the cuttings velocity and the fluid velocity.

Knowing what soil is encountered also provides more information regarding the risks identified in section 2.2. None of the risks in table 2.2 are as significant as assumed. The valley fill does not consist of gravel and larger rocks but of slightly coarser sand from the Eem formation. At larger depths the soil is not so coarse that the return flow stop due to infiltration losses. The Pot Clay, although encountered in Texel, is briefly drilled through without significant problems. Drilling PB1T-B in Texel was aborted amidst conflicting observations around the border to valley fill 2, so what is encountered there remains unclear.

---

8.3. Sample Identification

The tables below provide the $D_{90}$ of the samples taken during the initial vertical SI and taken during the pilot drilling. When more samples are tested the average is given. The bold figures indicate the cases where the $D_{90}$ of the samples taken during the vertical drilling is larger than the $D_{90}$ of the samples taken during the pilot drilling.

<table>
<thead>
<tr>
<th>Fm.</th>
<th>Vertical Samples</th>
<th>Pilot Samples</th>
<th>Vertical Samples</th>
<th>Pilot Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$D_{90}$ [mm]</td>
<td>$D_{90}$ [mm]</td>
<td>$D_{90}$ [mm]</td>
<td>$D_{90}$ [mm]</td>
</tr>
<tr>
<td>(No. Samples)</td>
<td>(No. Samples)</td>
<td>(No. Samples)</td>
<td>(No. Samples)</td>
<td>(No. Samples)</td>
</tr>
<tr>
<td>Naaldwijk</td>
<td>0.312 (3)</td>
<td>0.218 (1)</td>
<td>0.256 (2)</td>
<td>0.332 (1)</td>
</tr>
<tr>
<td>Boxtel</td>
<td>0.322 (1)</td>
<td>0.310 (1)</td>
<td>0.242 (3)</td>
<td>0.269 (1)</td>
</tr>
<tr>
<td>Kreftenhaye</td>
<td>0.209 (3)</td>
<td>0.329 (1)</td>
<td>0.345 (3)</td>
<td>0.348 (1)</td>
</tr>
<tr>
<td>Eem</td>
<td>0.236 (3)</td>
<td>0.406 (5)</td>
<td>0.591 (4)</td>
<td>0.311 (3)</td>
</tr>
<tr>
<td>Urk - Tynje</td>
<td>0.468 (9)</td>
<td><strong>0.271 (13)</strong></td>
<td>0.937 (4)</td>
<td><strong>0.436 (6)</strong></td>
</tr>
<tr>
<td>Peeloo</td>
<td>0.377 (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Urk - Veenhuizen</td>
<td>0.426 (6)</td>
<td>0.838 (6)</td>
<td></td>
<td><strong>0.491 (12)</strong></td>
</tr>
<tr>
<td>Appelscha</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Texel Sample Code</td>
<td>Den Helder Group</td>
<td>Texel Sample Code</td>
<td>Den Helder Group</td>
<td>Formation</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------</td>
<td>------------------</td>
<td>------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>B-65-TXL-03</td>
<td>1</td>
<td>PB1DH-A3-85-02</td>
<td>1</td>
<td>Boxtel Formation</td>
</tr>
<tr>
<td>B-65-TXL-04</td>
<td>2</td>
<td>PB1DH-A3-85-05</td>
<td>2</td>
<td>Kreftenheye Formation</td>
</tr>
<tr>
<td>B-65-TXL-05</td>
<td>3</td>
<td>PB1DH-A3-85-11</td>
<td>4</td>
<td>Eem Formation</td>
</tr>
<tr>
<td>B-65-TXL-07</td>
<td>3</td>
<td>PB1DH-A3-85-15</td>
<td>5</td>
<td>Urk Formation - Tynje</td>
</tr>
<tr>
<td>B-65-TXL-09</td>
<td>4</td>
<td>PB1DH-A3-85-16</td>
<td>5</td>
<td>Peelo Formation</td>
</tr>
<tr>
<td>B-65-TXL-10</td>
<td>4</td>
<td>PB1DH-A3-85-22</td>
<td>7</td>
<td>Urk Formation - Veenhuizen</td>
</tr>
<tr>
<td>B-65-TXL-12</td>
<td>4</td>
<td>PB1DH-A3-85-23</td>
<td>7</td>
<td>Appelscha Formation</td>
</tr>
<tr>
<td>B-65-TXL-13</td>
<td>4</td>
<td>PB1DH-A3-85-43</td>
<td>7</td>
<td>Appelscha Formation - Weerdinghe</td>
</tr>
<tr>
<td>B-65-TXL-15</td>
<td>4</td>
<td>PB1DH-A3-85-46</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>B-65-TXL-16</td>
<td>4</td>
<td>PB1DH-A3-85-47</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>B-65-TXL-28</td>
<td>4</td>
<td>PB1DH-A3-85-51</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>B-65-TXL-30</td>
<td>4</td>
<td>PB1DH-A3-85-54</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>B-65-TXL-32</td>
<td>4</td>
<td>PB1DH-A3-85-58</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>B-65-TXL-35</td>
<td>4</td>
<td>PB1DH-A3-85-59</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>B-65-TXL-40</td>
<td>4</td>
<td>PB1DH-A3-85-62</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>B-65-TXL-48</td>
<td>3</td>
<td>PB1DH-A3-85-63</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>B-65-TXL-51</td>
<td>3</td>
<td>PB1DH-A3-85-66</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>B-65-TXL-55</td>
<td>3</td>
<td>PB1DH-A3-85-69</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>B-65-TXL-59</td>
<td>3</td>
<td>PB1DH-A3-85-71</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>B-65-TXL-63</td>
<td>3</td>
<td>PB1DH-A3-85-75</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>B-65-TXL-67</td>
<td>3</td>
<td>PB1DH-A1-65-02</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>B-65-TXL-71</td>
<td>3</td>
<td>PB1DH-A1-65-05</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>B-65-TXL-74</td>
<td>3</td>
<td>PB1DH-A1-65-07</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>B-65-TXL-80</td>
<td>3</td>
<td>PB1DH-A1-65-09</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>C-85-TXL-04</td>
<td>4</td>
<td>PB1DH-A1-65-14</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>C-85-TXL-07</td>
<td>4</td>
<td>PB1DH-A1-65-17</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>C-85-TXL-09</td>
<td>4</td>
<td>PB1DH-A1-65-20</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>C-85-TXL-22</td>
<td>8</td>
<td>PB1DH-A1-65-23</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>C-85-TXL-25</td>
<td>7</td>
<td>PB1DH-A1-65-26</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>C-85-TXL-28</td>
<td>7</td>
<td>PB1DH-A1-65-29</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>C-85-TXL-31</td>
<td>7</td>
<td>PB1DH-A1-65-30</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>C-85-TXL-34</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-85-TXL-36</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-85-TXL-43</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-85-TXL-45</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-85-TXL-47</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-85-TXL-51</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-85-TXL-54</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
8.3. Sample Identification

(a) All Samples, Texel at the far end, Den Helder at the front.

Figure 8.4: An overview of the sieve results. Differences in samples size are mainly caused by a difference in the ratio between the progress of the drill and the pump rate. In Texel this ratio was often much lower than in Den Helder and as such the amount of grains left after sieving is often less for the samples taken in Texel than for the samples taken in Den Helder.

(b) A small sample.

(c) A large sample.
8.4. Source Location

As mentioned before multiple scenarios are used to get an estimate of the source location. Scenario 1 has the smallest diameter and the smallest loss of drilling fluid. This results in a source location relatively close to the drill location at sampling time. Scenario 9 has the largest diameter and the largest loss of drilling fluid, this results in a source location relatively far from the drill location at sampling time. Some scenarios result in unrealistic source locations. For example for Den Helder in scenario 7, 8 and 9 it seems that the first few samples of the second drilling belong to the end of the first drilling which is physically impossible due to the drill location and thus the start of the return flow. This is presented in a number of ways: lag diagram, elaborated in subsection 8.4.1 and in plots of the samples locations, elaborated in section 8.5.

In table 8.7 the results of the first and last samples from the first drilling in both Texel and Den Helder are shown for scenario 1, 5 and 9. It illustrates that the difference between scenario 1 and 9 increases with the bore hole length, but the specifics appear to be highly influenced by progress rate, frequency of data points and pump rate. As a result for a few samples multiple scenarios lead to the same source location. Figure 8.5a and figure 8.5b graphically show the results of the Intersection Method for different scenarios on samples from the same drilling in Den Helder. Sample PB1DH-A3-85-02 is taken at the start of the drilling at NAP-85m depth and PB1DH-A3-85-75 is taken near the end of the drilling at NAP-85m depth.

Table 8.7: Results from different scenarios for different samples from the same drilling in both Texel and Den Helder.

<table>
<thead>
<tr>
<th>Location</th>
<th>Sample Code</th>
<th>Drill Location at Sampling Time [m]</th>
<th>Scenario</th>
<th>Source Location [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Texel</td>
<td>B-65-TXL-03</td>
<td>111.95</td>
<td>1</td>
<td>88.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>79.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td>60.3</td>
</tr>
<tr>
<td></td>
<td>B-65-TXL-80</td>
<td>1681.58</td>
<td>1</td>
<td>1657.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>1648.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td>1600.1</td>
</tr>
<tr>
<td>Den Helder</td>
<td>PB1DH-A3-85-02</td>
<td>63.15</td>
<td>1</td>
<td>48.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>39.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td>29.9</td>
</tr>
<tr>
<td></td>
<td>PB1DH-A3-85-75</td>
<td>1973.66</td>
<td>1</td>
<td>1871.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>1794.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td>1298.6</td>
</tr>
</tbody>
</table>
(a) Sample PB1DH-A3-85-02, taken at the start of the Den Helder drilling at NAP-85m.

(b) Sample PB1DH-A3-85-75, taken at the end of the Den Helder drilling at NAP-85m.

Figure 8.5: Plots of the results of the 9 different scenarios for two samples taken at the start and the end of the NAP-85m drilling in Den Helder.
8.4.1. Lag Diagram
In this research there are data gathered about the flow rate, rate of penetration and connection time, this in contrast to the original paper (Garcia-Hernandez et al., 2008). Furthermore the path of the drilling is known, which shows that even in the horizontal part the inclination varies between about \(-2^\circ\) to \(+2^\circ\) with the horizontal axis. In figure 8.6 and figure 8.7 the head location is plotted against the source location, as calculated in scenario 5. These two figures are based on measurements and calculation on data from this project. Which is in contrast to figure 5.5, that is based on experiment in a laboratory (Garcia-Hernandez et al., 2008). Furthermore the source location in this research is determined using corrected fluid velocity while cuttings velocity, determined in a laboratory, is used in the original paper (Garcia-Hernandez et al., 2008). No such tests are performed for this research and thus no data are available to determine this velocity. Lag diagrams for all scenarios can be found in appendix D.4. In this appendix it is visible that the gradient of the lines graph for the NAP-65m and the NAP-85m drilling becomes lower as the scenario number increases. Furthermore it is easy to spot the samples that have unrealistic origins as the graphs become erratic, see figure 8.8.

It is presented that the ratio for progress of the drill head to pumped length is different in both sites. In Texel the gradient of both the 'NAP-65m' and 'NAP-85m' line are about the same as the 'ROP=0' line. Which means that the pump rate is relatively high compared to the progress rate. In Den Helder the 'NAP-65m' line has a lower gradient and the 'NAP-85m' line seems slightly lower as well. A lower gradient in this case means either faster progress or a lower pump rate.

![Texel Lag Diagram Scenario 05](image)

Figure 8.6: The lag diagram, relating the current location of the drill head to the location of the source of the cuttings found at the surface, constructed from the data gathered in Texel.
Figure 8.7: The lag diagram, relating the current location of the drill head to the location of the source of the cuttings found at the surface, constructed from the data gathered in Den Helder.

Figure 8.8: The lag diagram for scenario 9 in Den Helder.
8.5. Cross-section

In this section the results of the calculations of the source location are plotted on the initial cross-section. Figure 8.9 to figure 8.17 show the outcomes for the different scenarios.

The identified samples enables both an improvement of the cross-section as well as a validation of the scenarios. The validation is possible due to geological features that are known, for example from the vertical drillings conducted onshore in the initial SIs or from the seismic survey.

Figure 8.9: The calculated source location of the samples, according to scenario 1, plotted on the initial cross-section. The color indicates the formation as identified by geologist from TNO. The diameter is $1.1 \times D_{Drill}$ and the loss is 0%.

In figure 8.9 the situation as a result of scenario 1 is presented. The parameters used in calculating scenario 1 make this scenario unlikely more importantly the result is not consistent with the results of the seismic survey. Therefore, scenario 1 is improbable.
In figure 8.10 the situation as a result of scenario 2 is presented. The situation is possible but the parameters used in calculating scenario 2 make it unlikely since the diameter of the borehole is probably larger. Furthermore the results of the calculations contradict the results of the seismic survey.

Therefore, scenario 2 is improbable.
Figure 8.11: The calculated source location of the samples, according to scenario 3, plotted on the initial cross-section. The color indicates the formation as identified by geologist from TNO. The diameter is $1.1 \times D_{\text{Drill}}$ and the loss is 30%.

In figure 8.11 the situation as a result of scenario 3 is presented. The situation is possible but the parameters used in calculating scenario 3 make it unlikely since the loss is high and the borehole diameter is probably larger. Furthermore the results of the calculations contradict the results of the seismic survey. Therefore, scenario 3 is improbable.
In figure 8.12 the situation as a result of scenario 4 is presented. This is a likely result as it is a close fit to the seismic profile but the infiltration loss is 0% for this scenario, which makes it an unrealistic scenario. When compared with the results of the previous scenarios it is presented that the results of the source calculations and the results of the seismic survey are fitting better together.

Scenario 4 is improbable because of the assumed 0% loss for this scenario.
In figure 8.13 the situation as a result of scenario 5 is presented. This is a close fit to the seismic profile and has plausible parameter values. Some samples are not fitting to what is known but in general the fit is good. Scenario 5 is the most probable out of the 9 scenarios tested in this research.

This is however based on the assumption that one scenario is the best fit to reality for both locations. For Texel the results can checked with both the results of the initial vertical drilling and the results of the seismic survey at significant horizontal distance away from the starting point of the drilling. For the Den Helder side there is no data to check the results at horizontal distance away from the start of the drilling. This means that although scenario 5 is used as the best fit for both locations it is uncertain which one scenario is the best fit to reality on the Den Helder side.

Figure 8.13: The calculated source location of the samples, according to scenario 5, plotted on the initial cross-section. The color indicates the formation as identified by geologist from TNO. The diameter is 0.4m for Texel, 0.45m for Den Helder and the loss is 7%.
In figure 8.14 the situation as a result of scenario 6 is presented. This is the first scenario in which at least one result is unlikely without taking additional information into account. The yellow circle indicates a sample that is close to another sample but belongs to a different formation. Horizontal distance between the two samples is about 25m. It is however possible in this location due to the fact that there are two boreholes around that depth on that location due to multiple attempts, as is presented in figure 4.2. The results of the seismic survey contradict the calculated location of the Eem Formation at that point.

Scenario 6 is improbable because of the reasons mentioned above.
In figure 8.15 it is presented that the results of scenario 7 are contradicting each other. Sample origins in between samples identified as other formations are indicated by a red circle. Sample origins at places where there has not been drilled are indicated by a blue circle. This results comes from a combination of factors, among others the frequency of the gathered data. A dark blue circle indicates where based on the order in which the samples are taken the origin should be. The situation in the yellow circle is possible due to two boreholes around that location due to multiple attempts, it is however not likely to be true due to the seismic profile made in the initial SIs.

Scenario 7 results in an unrealistic outcome for the samples from Texel and Den Helder and the 0% loss of drilling fluids is unlikely as well.
In figure 8.16 it is presented that the number of samples at unrealistic positions is only increased in scenario 8 compared to scenario 7. The blue circle highlights samples that should be in the dark blue circle. The situation indicated by the yellow circle is unlikely, due to the conducted seismic survey.

Scenario 8 results in an unrealistic outcome for the samples from Texel and Den Helder.
In figure 8.17 it is presented that scenario 9 results in an unrealistic situation. The red circle highlights a sample location between or beyond samples of another formation. In the blue circle samples are indicated that belong in the dark blue circle, based on the order in which the samples are taken. The sample in the yellow circle is unlikely to be from that location due to the information derived from the seismic profile.

When looking at the lag diagram of this scenario (figure 8.8) it becomes clear that the samples in the blue circle are incorrectly located there. The points corresponding to those samples have a source location that is more that the position of the drill head, this is physically impossible. The lag diagrams found in appendix D.4 show the same erratic results for scenario 7 and 8.

Scenario 9 results in an unrealistic outcome for the samples from Texel and Den Helder and the 30% loss of drilling fluids is unlikely as well.
8.6. Improved Cross-section

Based on the results of the Intersection Method and the results of the initial SIs an improved cross-section can be constructed.

Figure 8.18 depicts what is known from the initial SIs and the calculated source location of the samples. In figure 8.19 a cross-section is constructed based on the data from figure 8.18. The certainty of the constructed cross-section becomes less the further away from the drillings on the sides and the samples. This is indicated by the dashed lines.

As is presented not all sample origins are perfectly aligned with what is known. Although scenario 5 is the most probable scenario, there is still room for improvement. The outcome could be refined by adjusting the scenarios and through iteration an improved outcome could be reached. This is not done since it is beyond the scope of this research. The aim of this research is to see whether it is possible to connect the samples to a reliable origin location.

More substantial improvements can be made by refining the Intersection Method. This can be done by acquiring additional data, for example multiple data points for the drill location per joint. Refinement could also be done by taking flow and transport processes into account. For example the location is now based on the fluid head but it is likely that the grains had a lower velocity than the fluid. There are also indications that a bed has formed in the borehole, this would decrease the cross-sectional area available for fluid flow. The velocity of the flowing fluid would increase in the area free of the bed but would slow down or stop in the area where the bed is.

The cross-section in figure 8.19 is mainly based on what is known, since the sample origin calculations are an approximation. What could make the approximation more accurate would be, for example; location based data about the loss of drilling fluid and taking possible bed formation in the borehole into account would increase the accuracy as well. Bed formation would result in a smaller area for the fluid to flow through, this would place the sample origin closer to the drill head location at sampling time, ie. further from the surface, and could result in a more accurate source location and thus a more reliable cross-section.
Figure 8.18: A cross-section with all that is known and the most probable sample locations.
Figure 8.19: A cross-section based on what is known.
8.7. Summary

It is found that the behavior of the drilling fluid is difficult to describe with one fluid model. Which of the two tested models varies from sample to sample. The flow is likely laminar as indicated by a value for the Reynolds number of in all cases less than 50. The transport capacity is in most but not in all cases sufficient, this is indicated by calculations as well as a comparison between the $D_{90}$ of samples from the vertical drillings and the $D_{90}$ of samples from the return flow.

The cuttings velocity is assumed to be about the same as the average fluid velocity. This is indicated by the calculations related to the fluid model and that no traces of significant mixing are found.

The risk identified in section 2.2 and table 2.2 are not as significant as previously assumed. In the valley fill gravel and larger rocks layers are not found but instead a coarse sand from the Eem formation is found. At larger depths the soil is not so coarse that the return flow stop due to infiltration losses. The Pot Clay, although encountered in Texel, has been drilled through without significant problems despite its high strength.

In this research scenario 5 (Loss = 7% and the diameter of the borehole = 0.4 or 0.45, for respectively Texel and Den Helder) is the most suitable, see table 8.8. This is due to the fit of the results to what is known, i.e. the results of the seisim curvey and the results of the initial vertical drillings. The arguments for this are however mostly based on the samples from Texel. This is due to the seismic survey which intersects with the data from Texel. It is however possible that the scenario with the best fit for the samples from Den Helder is another scenario. Unfortunately there is nothing that the Den Helder samples, at distance, could be correlated with, so in this research the same scenario is used for both Texel and Den Helder.

The lag diagrams created for this research, figure 8.6 till figure 8.8 and found in appendix D.4, can be taken into account to study the reliability of the results. In the case that the samples are located out of order the graphs become erratic with samples where the sample source is further than the drill head, which is physically impossible. This is presented for scenario 7 and 8, both found in appendix D.4 and for scenario 9 the results are erratic. In section 8.5 this is indicated in the figures by a blue circle.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Inconsistent with previous results and unlikely scenario parameters.</td>
</tr>
<tr>
<td>2</td>
<td>Inconsistent with previous results and an unlikely scenario parameter.</td>
</tr>
<tr>
<td>3</td>
<td>Inconsistent with previous results and unlikely scenario parameters.</td>
</tr>
<tr>
<td>4</td>
<td>Consistent with previous results and an unlikely scenario parameter.</td>
</tr>
<tr>
<td>5</td>
<td>Most probable, consistent with previous results.</td>
</tr>
<tr>
<td>6</td>
<td>Inconsistent with previous results, improbable outcome and an unlikely scenario parameter.</td>
</tr>
<tr>
<td>7</td>
<td>Inconsistent with previous results, improbable and unrealistic outcomes and an unlikely scenario parameter.</td>
</tr>
<tr>
<td>8</td>
<td>Inconsistent with previous results, improbable and unrealistic outcomes.</td>
</tr>
<tr>
<td>9</td>
<td>Inconsistent with previous results, improbable and unrealistic outcomes and an unlikely scenario parameter.</td>
</tr>
</tbody>
</table>
Conclusions

This chapter presents the conclusions of this research. These are the main subjects:

- Drilling Fluid
- Source Location
- Cross-section
- Suitability of Horizontal Directional Drilling (HDD) for Site Investigation (SI).

Of the main subjects in this chapter; number 2 (Source Location), 3 (Cross-section) and 4 (Suitability of HDD for SI) correspond to the 3 principal questions of this research defined in section 1.1. Main subject 1 (Drilling Fluid) is required to investigate main subject 2 (Source Location) but it is not a question by itself.

9.1. Drilling Fluid
In this section all conclusions that are related to the different aspects of the drilling fluid can be found.

9.1.1. Fluid Model
The Bingham (BH) model is in Texel the best fit to the rheologic behaviour of the drilling fluid for about 80% of the experimental data. In Den Helder it is less than 50%, with the more suitable fluid model being the Herschel-Bulkley (HB) model. When using a weighting factor, increasing the importance of the deviation around the calculated shear of the return flow, the BH model becomes the best suited model in less cases. Only 65% in Texel and about 33% in Den Helder.

Other fluid models or another approach (to preliminary calculations) is worth investigating since the fluid model and behavior are inconsistent during the drilling. Predictions or prescribed requirements may end up being invalid or unreachable since the ratios between different properties vary from fluid model to fluid model. This is the case with the yield stress. The value in the HB model can be as low as 11% of the value for the yield stress in the BH model in Den Helder. The fluid model did not have to be taken into account in the calculation of the source location since the grains appear to have been transported to the surface at fluid velocity. This is implicated by the fact that no signs of mixing are found while identifying the samples.

9.1.2. Time Dependent Behavior
The properties of the drilling fluid used in Den Helder are changing in time. This in addition to changes caused by interaction with the soil, groundwater and changes in fluid mixture. There are not enough data available for Texel to investigate possible time dependency of the fluid properties for that location. In order for this change to be taken into account, more research is required to determine the cause and the scale and subsequently the significance of the processes.

9.1.3. Transport and Flow Parameters
The flow in the borehole is laminar and the build up of a bed is expected. This is indicated by the difference in $D_{90}$ between the samples taken from the vertical drilling and the samples taken from the return flow. In general the $D_{90}$ is lower for samples taken from the return flow compared to the $D_{90}$ of the samples taken from the vertical drillings that are made onshore in both Texel and Den Helder.

This difference can be significant. For example for the Peelo Formation in Den Helder it is about a factor 2, $D_{90} = 0.937 \, mm$ (based on 4 samples from the initial vertical drilling) versus $D_{90} = 0.436 \, mm$ (based on 6 samples from the return flow during the horizontal drilling). An exception is the Eem Formation from the Texel side for which the $D_{90} = 0.236 \, mm$ (based on 3 samples from the initial vertical drilling) versus $D_{90} = 0.406 \, mm$
(based on 5 samples from the return flow during the horizontal drilling). This is due to the fact that the horizontal drilling, drilled through the bottom of the filled valley where much coarser material is found.

The influence of a bed in the borehole on the return flow and on the calculated origin locations of the samples is not taken into account in this research.

It is found that the properties and the fluid model of the return flow are determined by the properties of the inflow, the soil, the groundwater and time dependent processes.

9.2. Source Location
The first goal of this research is to investigate whether it is possible to connect samples from the return flow to a source location in the subsurface. This is possible but such a location cannot be calculated at once due to the uncertainties with regards to the diameter of the borehole and the loss of drilling fluid. Determination of a range of possibilities in order to get an indication of what the subsurface might look like is possible. The Intersection Method is created in this research for that purpose. An optimum scenario can be found by calibration if additional SIs have taken place, such as in this project the vertical onshore drillings and the seismic survey.

In this research scenario 5 (loss of drilling fluid is 7% and the diameter of the borehole is 0.4m and 0.45m for respectively Texel and Den Helder) is the most probable option because of the plausible parameters and the small difference between the outcome of the scenario and the results of the initial SIs.

More data are required in order to calculate the source location solely based on data gathered during the pilot drilling. An increase in the frequency of data, for example outflow, as well as additional data, for example gamma ray, are required. These data would serve to decrease the uncertainties with regards to the losses and diameter of the borehole as well as increase the data available to determine additional parameters of the soil.

Most of the risk identified before the pilot drillings as found in section 2.2 are regarding possibilities. During the pilot drillings however most of these risks are found to be of less significance. None of the risks in table 2.2 are as significant as assumed. The valley fill does not consist of gravel and larger rocks but of slightly coarser sand from the Eem formation. At larger depths the soil is coarser but not so coarse that the return flow stops due to infiltration losses. The Pot Clay, although encountered in Texel, is briefly drilled through without significant problems.

9.2.1. Lag Diagram
The lag diagram presents the source location as a function of the head location (Garcia-Hernandez et al., 2008). In reality the HDD process is less predictable and assumptions made in the paper seem to be unrealistic. For example it is wrong to assume that no interruption of the drilling process takes place or to assume that steady state conditions occur for both flow rate and the rate of penetration. The lag diagrams shown in this research are based on gathered data and the graphs are produced after the project. To calculate the source location using the formulas given for the lag diagram, equation 5.44 and equation 5.45, the parameters in the formula would have to be location dependent. For example the $\alpha_h$, which is the inclination with respect to the horizontal axis, still varies between $-2^\circ$ and $+2^\circ$ during the horizontal part. An estimation could be made for the average Rate Of Penetration (ROP) and coupling time, based on data from multiple projects. Coupling time is the time required to connect the next pipe to the Drillstring (DS). The average ROP and coupling time should be included to make the approach more useful in the field if such data are not (yet) available from the project.

9.3. Cross-section
The second goal of this research is to investigate whether it is possible to construct a cross-section based on the results of the samples taken from the return flow. Construction of a cross-section is possible based on the samples taken from the return flow of a HDD project. The samples are still identifiable after sieving, which reduces the samples to grains of 0.063mm and larger. This means that no significant mixing of soil from different sources has occurred in the borehole during transport. Therefore it is concluded that the effect of wall erosion is limited, as this would add grains from the eroded formations along the borehole.

9.4. Suitability of HDD for SI
The third goal of this research is to investigate if HDD is suitable as a SI technique and if it can be improved, based on experiences in the field. With a few adjustments compared to a regular HDD project, the Marsdiep project gained a lot of additional data. That more can be acquired does not diminish the quality and quantity of the data gathered at the sites for this project.
What does diminish the suitability of HDD as a SI technique is that in this research it is not possible to connect the samples from the return flow to a source location in the subsurface without the results of additional SIs to correlate the results. In order to improve the suitability additional measurements are required. When the loss of drilling fluid and the diameter of the borehole are better known, HDD can be used separately as a SI to determine the geology.

Among the additional data gathered for the Marsdiep project are detailed pump rates and pump times, part of the data acquired by the Prodata system. Although these can be reconstructed from the driller logs, which are part of a regular HDD project, it is less accurate that way and it is preferable that the data are acquired in as much detail as possible.
10 Recommendations

This chapter presents recommendations for further research that needs to be conducted in order to improve the accuracy of the source location determination and thereby the suitability of Horizontal Directional Drilling (HDD) as a Site Investigation (SI) technique.

10.1. Drilling Fluid

More research considering other rheological models should be conducted. This is due to the uncertainty regarding which of the fluid models is the most suitable for certain types of drilling fluid. Research should be done with regards to the reliability of the results of the preliminary calculations. Research is also required to improve the usefulness of these results, since the variation in which fluid model represents the fluid most has implications for the fluid properties, in other words if the fluid changes to such an extent that the fluid model changes it is highly recommendable that the properties according to the new model still fulfill the requirements.

10.1.1. Rheological Data Acquisition

The effect of the sieve test on the fluid properties should be investigated. An effect is noted during the fieldwork but the extent of this is currently unknown. This could be investigated by making a drilling fluid that closely resembles return flow and by dividing it in 3 parts. Part 1 and 2 are put through a 0.063 mm sieve without vibration in order for the sand to be removed and not having it influence the equipment. Part 1 is tested in a viscometer, part 2 is first mixed with a high shear mixer before testing the rheology in the viscometer and part 3 is put through a set of sieves on a vibrating table for about 10 min before testing the rheology in the viscometer. The viscometer test could be done at different time intervals after sieving to check for time dependency.

10.1.2. Fluid Model

In this research a weighting factor is introduced to get a better idea which fluid model is most suited to describe the fluid in a specific situation. However the weighting factor is applied after fitting the fluid models to the experimental data. Further improvement could be reached by incorporating a weighting factor in the fitting method. This would make the fit better around the required shear rate instead of just selecting the method which has the best fit at that shear rate value.

10.1.3. Processes

For this research two of the most uncertain parameters, loss and diameter, are assumed to be constant along the length of the drilling. This is however not realistic and further research should be conducted for a better understanding of the processes down hole into the influence of this location dependency and into what influences the amount of variation.

Research in this direction is performed but focused on pipes (Rice et al., 2015a,b). However getting an understanding of the interaction of the fluid flow with the Drillstring (DS) and optional bed would improve the understanding of the processes and increase the accuracy of the predictions as well as the calculated origin of samples taken from the return flow.

10.2. Source Location

Calculation of the source location of the samples is reasonably accurate. Through iteration and comparing the results to what is known, the scenarios could be adjusted to give a more accurate result. The Intersection Method used in determining the source location of the samples could be refined in the following ways:
• **Data:** some data are gathered at an insufficient frequency at an unsuitable location, examples are drill head location and return flow respectively. Additional data would improve the accuracy of the result without improving the method.

• **Location dependency:** the uncertainties with regard to diameter and drilling fluid loss are solved by the use of scenarios in this research. However these parameters are not the same along the borehole. Through iteration of the process up and including constructing a cross-section the diameter and loss could be made dependent on the location. Results of additional SIs are required for correlation.

• **Determination of the grain velocity:** this is done by implementation of the fluid and transport processes occurring in the borehole. Some of these processes might require additional investigation into the occurrence and magnitude of influence. At present the source location is based on the fluid head location. It is possible that the grains are transported at a slower rate than the fluid velocity. How much slower is unknown but depended on among others the fluid and the transport processes (Garcia-Hernandez et al., 2008).

### 10.3. Cross-section

By taking more data into account the cross-section can be significantly improved. These improvements might not all be visible ones but the reliability of the cross-section can increase as well. For example in this research the data gathered with respect to the torque and push/pull forces on the drilling rig are barely mentioned. When taking those data into account a more refined geo-technical cross-section might be constructed. This can be done by dividing the formation into geo-technical units with roughly the same properties based on the torque and push/pull forces that are measured.

### 10.4. Suitability of HDD for SI

The accuracy of the flow test could be improved with the use of a method that enables continuous measurements at the surface. A fairly simple example of such a method would be adding a pigment or a dye to the drilling fluid. A more complex example could be a chemical of some kind that could be measured continuously directly in the return flow. A continuous measurement would result in data about possible mixing in the borehole as well as well as determining the peak in the return flow more precisely. The flow processes occurring during continuous pumping in a borehole with a clean, thinner fluid, might be very different from what happens with a thicker fluid, loaded with grains in a borehole are the DS rotates and is pushed forward and the pump stops roughly every 5 - 8 minutes for about 10 minutes. The results might not be that useful for figuring out what happens downhole during drilling. It can however be used to determine the average diameter of the hole, which is required to get an accurate estimate of the source location of the samples.

What would enable a more precise measurement of the loss would be among others the installation of a flow meter on the pipe between the recycling unit and the tank where mud is made. This would serve to get an estimate of the magnitude of the loss of fluid due to the recycling unit as well as due to the amount of cuttings. This would improve the accuracy of a mass balance for the whole system. Adding a logged flow meter on the casing would help to both measure the return flow accurately and the amount of backflow, if present. Losses could be determined per length and this would result in an indication of the permeability of the soil as well as improve the accuracy of the source location.

Data loggers on the pipes instead of flow meters, which have to be read manually, would improve both the amount of data and would also allow for a location determination of where significant losses occur. A proposal for an improved site setup is presented in figure 10.1.

So far these are all improvements to be made on the surface but additional tools could be lowered into the borehole as well. In Texel a gamma ray log is put behind the drill head, which due to a change in companies is not used throughout all the drillings in Texel. But such a tool, that measures the clay content of the surrounding soil, can deliver very useful information about the situation in the borehole. Additional tools can be thought of that can result in useful information, for example a tool that fulfills the same function as the calliper log in Directional Drilling (DD) in the petroleum industry, would result in information about both the geometry in the borehole as well as the size of the borehole. Measuring the size of the borehole during drilling and pulling back would result in data about the sensitivity of the layers to erosion, depending on the difference in diameter on the same location.

Improvements as mentioned above will result in an increased suitability of HDD for SIs. This will lead to more knowledge and a better understanding of the soil and thus an optimization is possible of the both the design and required material.
For determination of the source of the sample fewer modifications to a regular project are required than described above or depicted in figure 10.1. However additional SI (old or new) will be required to correlate the results and find the most probable solution. To get the source of the samples through use of the Intersection Method, as described in this research, the following data are required:

- Drill location
- Pump rate and when the pump is on

Both the drill location and the pump rate/ pump times are monitored and recorded during regular projects. Pump time is often the same as drilling time, with the exclusion of additional pump time while no progress is made to clean the borehole. The other required parameters can be varied till a solution is found that consistent with what is known from additional SIs. Those parameters are diameter of the borehole and loss of drilling fluid.


Brownline B.V. (2010). Drillguide GST.


TNO (2013c). Subsurface Models - GeoTOP.


Appendices
Contents Appendices

A Geology Background 96
A.1 Grain Size 96
A.2 Formations 96
A.2.1 Naaldwijk Formation (NA) 97
A.2.2 Nieuwkoop Formation 98
A.2.3 Boxtel Formation (BX) 98
A.2.4 Kreftenhaye Formation (KR) 99
A.2.5 Eem Formation (EE) 99
A.2.6 Drente Formation (DR) 99
A.2.7 Urk Formation 100
A.2.8 Peelo Formation (PE) 101
A.2.9 Formation of Appelscha (AP) 101
A.2.10 Formation of Peize (PZ) 101
A.3 Pre-Existing Model 102

B Initial Site Investigations, Proposals and Calculations 103
B.1 Initial Site Investigation 103
B.2 Drilling Proposals 103
B.2.1 Texel - Proposed 104
B.2.2 Den Helder - Proposed 105
B.3 Boundary Conditions 106
B.4 Parameters, Dimensions and Variables 106
B.4.1 Geometry of the boring 106
B.4.2 Product pipe 106
B.4.3 Execution 106
B.4.4 Drilling fluid 107
B.4.5 Coefficient of friction 107
B.4.6 Other 107
B.5 Results 107
B.5.1 Minimal Drilling Fluid Pressure 107
B.5.2 Torque and Normal Force 108
B.5.3 Rate Of Penetration (ROP) 108
B.5.4 Infiltration and loss of drilling fluid 110
B.5.5 Transport Capacity 110
B.5.6 Height of Mound 112
B.5.7 Casing Length 112
B.5.8 Other 112

C Equipment and Tests 113
C.1 Laboratory Equipment 113
C.1.1 Marsh Funnel 113
C.1.2 (Roto) Viscometer, (Fann) Rheometer 113
C.1.3 Mud Balance 114
C.1.4 Electrical Conductivity Meter 114
C.1.5 Sand Content Kit 114
C.1.6 Sievetest 115
C.1.7 pH Meter 116
C.2 HDD Equipment 116
C.2.1 Mud Pulse 116
D  Additional Results
   D.1 Maximum Diameter for Transport ..................................................... 117
   D.2 Reynolds Number ............................................................................. 119
   D.3 Yield Point - Best Fit .............................................................. 121
      D.3.1 Texel ................................................................................. 121
      D.3.2 Den Helder ....................................................................... 122
   D.4 Lag Diagram ................................................................................. 123
      D.4.1 Texel ................................................................................. 123
      D.4.2 Den Helder ....................................................................... 128
E  Code of the SampleSource Program ....................................................... 133
F  Code of the MarsdiepVisual Program .................................................... 206
A more extensive overview of the geology is presented in this appendix.

A.1. Grain Size
In table A.1 the definitions used with regards to the grain size can be found.

Table A.1: The definition of grain sizes (ISO, 2013).

<table>
<thead>
<tr>
<th>Name</th>
<th>Size range (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very coarse soil</td>
<td></td>
</tr>
<tr>
<td>Large boulder</td>
<td>(&gt;630)</td>
</tr>
<tr>
<td>Boulder</td>
<td>200–630</td>
</tr>
<tr>
<td>Cobble</td>
<td>63–200</td>
</tr>
<tr>
<td>Gravel</td>
<td>20–63</td>
</tr>
<tr>
<td>Medium gravel</td>
<td>6.3–20</td>
</tr>
<tr>
<td>Fine gravel</td>
<td>2.0–6.3</td>
</tr>
<tr>
<td>Coarse sand</td>
<td>0.63–2.0</td>
</tr>
<tr>
<td>Sand</td>
<td>0.2–0.63</td>
</tr>
<tr>
<td>Medium sand</td>
<td>0.063–0.2</td>
</tr>
<tr>
<td>Fine sand</td>
<td>0.02–0.063</td>
</tr>
<tr>
<td>Silt</td>
<td>0.0063–0.02</td>
</tr>
<tr>
<td>Medium silt</td>
<td>0.002–0.0063</td>
</tr>
<tr>
<td>Fine silt</td>
<td>(\leq 0.002)</td>
</tr>
<tr>
<td>Clay</td>
<td></td>
</tr>
</tbody>
</table>

A.2. Formations
The different formations are described in chronological order, starting with the youngest.

The general paragraph of the descriptions is based on the Nomenclator from TNO.

The local occurrence paragraph is based on the descriptions of the initial drillings conducted for this project as well as the sieve curves from samples gathered during the initial Site Investigations (SIs). The sieve curves are created after the original descriptions and calculations are made and thus can result in a different description of the soil. Tables with characteristic percentages can be found at the formations for which a sieve curve is constructed. The characteristic percentages are based on those used in the preliminary calculations in order to be able to make a comparison afterwards; D10, D15, D50 and D90.

The risk section is based on the risk matrix behind the Geobrain or Soilrisk website (Deltares, 2010), which is based on the experience of a number of companies active in the underground in the Netherlands. The location of the Marsdiep project is on the boundary between two areas for which the risks are classified and as such an over prediction of the amount and severity of the risks may occur. The risks are subdivided into 9 ground related risks and 3 risks related to groundwater. Only the most severe risks for Horizontal Directional Drilling (HDD) are presented at the different formations in this thesis.

The parts will start with the name of the formation followed by, in brackets, the abbreviation used in figure 3.1 and the tables with the soil parameters in appendix ???. These tables are sorted on soil type so one table can contain the parameters used for multiple formations.
### Table A.2: Formation Descriptions, brief and in general, based on Dutch formation descriptions (Nugroho and van Meerten, 2015).

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Age &amp; Formation Name</th>
<th>Description of the soil type in the geological unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Na</td>
<td>Holocene: Naaldwijk Formation, layer set of Nieuwkoop</td>
<td>Clay and locally peat, on silty sand with varying grains size also inclusions of fine layers and shells; locally fining upwards sequences, an indication of gully filling.</td>
</tr>
<tr>
<td>Bx</td>
<td>Weichselien to Holocene: Boxtel Formation</td>
<td>Very fine to medium fine sand; silt and sandy units show results of cryoturbation.</td>
</tr>
<tr>
<td>EE-BB</td>
<td>Weichselien or Eemian: Eem Formation Bruine Bank</td>
<td>Fine to medium fine sand with clay layers; layering and the presence of thin fine coarsening upwards layers indicate a &quot;low-energy&quot; fluvial coast environment.</td>
</tr>
<tr>
<td>Dr</td>
<td>Saale Glaciation: Drente Formation</td>
<td>Coarse and gravelly sand or boulder clay</td>
</tr>
<tr>
<td>EE-URTY</td>
<td>Cromerien to Saale Glaciation: Urk Formation, layer set of Tynje</td>
<td>Coarse, mostly gravelly sand, with local clay and peat layers of limited spread and thickness in upper 10 meters of the unit (higher than NAP -50m); high but varying conus resistances with random pattern indicate &quot;high-energy&quot; fluvial activity</td>
</tr>
<tr>
<td>Val_a</td>
<td>Saale Glaciation or Eemian: Drente Formation or Eem</td>
<td>Coarse gravelly sand, in a complex or variable &quot;valley fill&quot;, top part consists of fine to moderately coarse sand</td>
</tr>
<tr>
<td>Val_b</td>
<td>Saale Glaciation or Eemian: Drente Formation or Eem</td>
<td>Coarse gravelly sand, in a complex or variable &quot;valley fill&quot;, with the possibility of a very coarse gully filling at the bottom containing pebbles and boulders present in a layer or as isolated blocks higher in the &quot;valley fill&quot;</td>
</tr>
<tr>
<td>PENI</td>
<td>Elster: Peelo Formation, layer set of Nieuwolda</td>
<td>Strong (pot)clay</td>
</tr>
<tr>
<td>PE</td>
<td>Elster: Peelo Formation</td>
<td>Medium fine sand with clay layers</td>
</tr>
<tr>
<td>URVE</td>
<td>Cromerien to Saale Glaciation: Urk Formation</td>
<td>Medium to very coarse, generally gravelly sand; possibly fineing upward toplayer (Elster)</td>
</tr>
<tr>
<td>AP, PZ</td>
<td>Pliocene and early Pleistocene: Formation of Peize, Appelscha</td>
<td>Medium to very coarse, generally gravelly sand</td>
</tr>
</tbody>
</table>

Note: Some of the names of the ages correspond to glacial periods, these names are depending on where they are identified as such, these names are different in different countries.

### A.2.1. Naaldwijk Formation (NA)

#### General

The Naaldwijk Formation is formed in the Holocene, from 0.01 MA ago to the present. The formation is of marine origin but is characterized by a strong variation in lithological content, varying from very coarse sand to weak silty clay. It mainly consists of gray clayey sand, weak silty, can be calcareous and contain shells as well.

Often this formation is present at the surface but when covered it is by: peat from the Nieuwkoop Formation, clay or sand from the Echteld Formation. These three formations are of the same age and can intersect into each other. The formation is often discordant deposited on underlaying formations and the transition is sharp in general. In the case that the Formation is deposited on the Eem Formation the border can be hard to determine, as is the case on the Texel side of the Marsdiep HDD project.

#### Local Occurrence

Within the borings the Naaldwijk Formation is identified in the Texel boring as a brown to gray, moderately fine, weak to moderately silty sand, which can contain traces of iron oxide, traces of micas, shells, small clay layers and traces of detritus at the bottom of the formation, which could indicate the transition to Nieuwkoop Formation. In the Den Helder boring it is identified as a brown to gray, moderately fine, well sorted, weak silty sand, which
can contain traces of organic material, traces of micas, shells and clay as well as some peat and gyttja layers are found towards the bottom of the formation, which could be Nieuwkoop Formation. The sieve curves can be described in Texel as a weak silty fine to coarse sand with one sample containing gravel and in Den Helder as a silty fine sand.

**Risks**
This formation can consist of a weak clay which may be active so chemically swelling can occur. The formation can consist of fine sand which is susceptible to erosion by the drilling fluid. The formation is also deemed to be unpredictably variable. At the bottom of the formation wood can be encountered and form obstacles. The groundwater is salt, which may pose a risk for the properties of the drilling fluid but no further risks are identified with relation to the groundwater.

**A.2.2. Nieuwkoop Formation**

**General**
The Nieuwkoop Formation is formed in the Holocene, from 0.01 MA ago to the present. The formation consists mainly of brown to black peat but can also contain clayey peat or sandy peat as well as Gyttja.

This formation is often found at the surface. If it is covered by another formation it is often the Naaldwijk Formation or the Echteld Formation. These three formations are of the same age and often intersect into each other. The bottom of this formation is often formed by the Boxtel Formation. If this is the case the transition is sharp. In coastal areas intersection with the Naaldwijk Formation and the Echteld Formation is more common.

**Local Occurrence**
The Nieuwkoop Formation is not distinguishable from the Naaldwijk Formation and is not identified separately. However it is suspected to be found between the Naaldwijk Formation and the Boxtel Formation, identifiable only by a slight increase in organic and clay content, in particular small detritus layers and small clay layers or inclusions. No sieve tests are conducted on samples from this formation in the initial SIs.

**Risks**
For the most part this formation is reasonably safe considering risks due to obstacles but it can contain parts of old oak trees and at the bottom the boundary is formed by the transition to sand layers. Going from weak to strong(er) layers may pose a range of risks, for example deviations from alignment or sideways bending of the Drillstring (DS) as a reaction on the strong layers at the front but weak layers on the sides. The groundwater is salt, which may pose a risk for the properties of the drilling fluid but no further risks are identified with relation to the groundwater.

**A.2.3. Boxtel Formation (BX)**

**General**
The Boxtel Formation is formed from the Middle Pleistocene to the Holocene, from 0.66 MA ago to present. In this formation a few ground types are predominantly present:

- Light yellow to dark brown, very fine to moderately coarse sand which is weak to strong silty and can be calcareous.
- Light yellow to light gray, very fine to moderately fine, very silty sand, which can be calcareous.
- Grayish brown to dark gray, weak to strong sandy loam, which can also contain clay and organic parts, can be calcareous.

This formation can be found on the surface in the south and eastern parts of the Netherlands, in the Northwest. However this formation is covered by the Echteld Formation, Naaldwijk Formation or Nieuwkoop Formation. The bottom boundary in the Northwest of the Netherlands is often the contact with the Eem Formation, which is a sharp transition to calcareous sands with fossils.

**Local Occurrence**
In the Texel boring this formation is identified as a gray, very to moderately fine, moderately to very silty sand, which can contain traces of organic material and small layers of peat and clay. In the Den Helder boring this formation is identified as a light brownish gray, very fine, moderately sorted, slightly silty sand. The sieve curves can be described as a silty fine sand.
Risks
In the parts of the formation consisting of eolian sands there is a major risk of softening while drilling and the soil is susceptible for erosion by drilling fluid. The formation is highly unpredictable due to the presence of filled river valleys and stones can be present towards the bottom of the formation. The groundwater is salt, which may pose a risk for the properties of the drilling fluid and locally over pressured aquifers can be found, which can cause seepage into the drilled borehole and local borehole instability.

A.2.4. Kreftenheye Formation (KR)

General
The Kreftenheye Formation is formed in the mid Pleistocene and early Holocene, from approximately 0.24 to a few thousand years ago. The Formation mostly consists of yellow gray to brown gray, moderately coarse to very coarse, colored sand, which can contain gravel and chalk. Locally some clay and peat layers can be found. The top of the formation in the Marsdiep area is typically formed by the Boxtel Formation or the Eem Formation. Often a sturdy, gray to black clay layer is present at the top of the Kreftenheye Formation. In this case the transition between the formations is sharp.

Local Occurrence
This formation is not identified during the initial SI. The formation is however identified during the investigation of the samples from the horizontal drilling as a brownish, colored sand. Slightly deviating from the standard description of the Kreftenheye Formation but it matches the description for unit A3, identified as the Kreftenheye formation (Peeters et al., 2016).

Risks
The formation can be locally weak and due to the coarseness highly permeable. The ground water is salt, which may pose a risk for the properties of the drilling fluid and there is a low risk for over-pressured aquifers.

A.2.5. Eem Formation (EE)

General
The Eem Formation is formed in the Eemian, an interglacial period and a stage in the Late Pleistocene from 0.126 to 0.116 MA ago. The Formation mostly consists of gray, moderately fine to very coarse sand with shells, often contains chalk and with locally present shell layers. It can also contain dark gray clay and chalk layers. The top of the formation in the Marsdiep area is formed by the Boxtel Formation and is noticeable as a change towards, often non-calcareous, sand without fossils. At the bottom boundary a diatomite or gyttja layer can be present and used as indication for the boundary. The underlaying formation is the Drente Formation in the area of the project.

Local Occurrence
In the Texel boring this formation is identified as a gray to dark gray, well sorted sand, which can contain a few detritus layers at the top of the formation, traces of colored material and clay layers. In the Den Helder boring this formation is identified as a dark yellow to brownish gray, very sandy clay with micas or a brownish gray, very to moderately fine, well sorted, slightly silty sand with traces of micas. The sieve tests reveal that the Bruine Bank section is finer than the Eem formation and the samples indicate a weak silty to silty fine sand in Texel and a very silty to silty fine sand in Den Helder for the Bruine Bank section. The curves from the Eem formation can be described in Texel as a coarse sand and in Den Helder as a weak silty fine to coarse sand.

Risks
The formation is considered weak and the risk of softening of the sand is present. At the bottom of the formation the permeability might be high. The groundwater is salt, which may pose a risk for the properties of the drilling fluid and locally over pressured aquifers can be found, which can cause seepage into the drilled borehole and local borehole instability.

A.2.6. Drente Formation (DR)

General
The Drente Formation is formed during the Middle and Late Saalian, a glacial period and a stage in the Middle Pleistocene from 0.238 to 0.126 MA ago. In this formation a few ground types are predominantly present;
• Moderately to very coarse sand, with gravel.
• Grayish blue to brownish gray, clay and loam, sandy with gravel.
• Grayish blue to brownish gray, moderately fine sand, badly sorted can also contain gravel, cobble and boulders.
• Dark gray to dark brown, weak to moderately silty clay, calcareous and strong.

In the western part of the Netherlands the top of this formation is formed by the Eem Formation. The transition is gradual and can be determined by the presence of shells or when present by a diatomite or a gyttja layer at the bottom of the Eem Formation. The Drente Formation is sharp and discordant deposited on older formations. In this case the Urk Formation, which is recognizable as slightly more colored. If gravel is present this can be used since the gravel in the Drente Formation has a glacial component, amongst others; flint, granite and more crystalline, while the gravel in the Urk Formation doesn’t have this component.

Local Occurrence
This formation is identified as a gray, strong sandy clay, which can contain boulders on the Texel side and as a sand which can contain boulders on the Den Helder side.

Risks
In this formation the permeability can be high and at the bottom of the formation obstacles might be encountered. The groundwater is salt, which may pose a risk for the properties of the drilling fluid and locally over pressured aquifers can be found, which can cause seepage into the drilled borehole and local borehole instability.

A.2.7. Urk Formation
General
The Urk Formation is a fluvial formation deposited by the Rhine from the Late Cromerian to the Middle Saalian. Cromerian is a stage on the border between Early and Middle Pleistocene and Saalian is the last stage in the Middle Pleistocene, thus the age of this formation ranges from circa 0.85 to 0.13 MA.

The Urk Formation mainly consists of gray, after oxidation yellow to brown, colored, moderately fine to very coarse, sand with gravel. Which can be calcareous and contain organic remains. The formation can also consist of fine to very coarse gravel, with a relatively high percentage of white quartz veins in the gravel.

The top of the formation is in general not clear when bordering on the Drente Formation. Only when gravel is present a decrease in glacial gravel might be noticed. Since the formation is intersected with the Peelo Formation this can be found on top as well. In this case the boundary is not clear but can be identified by a decrease in the median of sand grains and the Peelo Formation is less colorful. For the bottom boundary with the Peelo Formation the opposite is true. In the case that the Urk Formation borders on either the Peize or Appelscha Formation, the border can be noticed because the soil becomes more calcareous, the variety of colors increases and within the gravel an increase in quarts veins and a decrease in clear quartz.

Local Occurrence
The Urk Formation - Tynje Section is on the Texel side described as a gray, moderately fine to very coarse, slightly to very silty sand, with traces of colored material, shells, white quartz and organic material and a variation in sorting. The Tynje section is on the Den Helder side described as light gray to gray, slightly silty moderately coarse sand, with traces of organic material and shells and badly to moderately well sorted. In Den Helder the Veenhuizen section is identified as well. The Veenhuizen section is described as a gray, weak silty moderately fine to moderately coarse sand, with traces of colored material, pink grains, few black grains and well sorted. In general it appears that the Urk Formation is more diverse in appearance on the southern, Den Helder, side of the project. Sieve tests conducted on samples from this formation indicate that the Tynje section consists of a coarse sand on the Texel side and a fine sand on the Den Helder side. The Urk formation is only tested on the Den Helder side the curve indicates a weak silty fine sand.

Risks
The formation properties can vary unpredictably and as such there is a risk of encountering weak zones and the sudden transition to strong(er) zones as well. Both chemically and physical swelling of the clay present might occur and obstacles might be present at the bottom of the formation for the Tynje section. Urk and the Tynje section might be very permeable. The groundwater is salt, which may pose a risk for the properties of the drilling fluid and locally over pressured aquifers can be found, which can cause seepage into the drilled borehole and local borehole instability.
A.2.8. Peelo Formation (PE)

General
The Peelo Formation is formed during the Elsterian, a glacial period and a stage in the Middle Pleistocene from 0.465 to 0.418 MA ago. The formation mainly consists of yellowish gray to light or dark gray and brownish gray, very fine to moderately fine sand. It can also consist of light to dark gray and brownish black to black, weak silty and sandy often very calcareous clay, which can be very strong. This part of the formation is also called ‘potklei’ (English; pot clay), or the Nieuwolda section, which is a very strong over-consolidated clay and is present in the northern part of the Netherlands.

The top boundary is formed by the Urk Formation as well as the bottom. In general both boundaries are not sharp but can be noticed by a change in the median of the sand present as well as by a change in the variety of colors. The bottom can also make contact with the Formations of Peize and Appelscha.

Local Occurrence
Both distinct types of this formation can be found in the drillings from Den Helder and Texel. The Nieuwolda layers are described as dark to blackish gray, weak silty clay on the Den Helder boring and as a dark gray, very strong clay containing some gravel in Texel. In Den Helder the Peelo formation is described as a gray to dark gray, moderately to badly sorted, moderately coarse to very coarse sand, which can contain silt, gravel and traces of white quarts. Sieve tests indicate a weak silty, weak gravelly, fine to coarse sand.

Risks
This formation has a high risk of physical swelling where potclay occurs and can locally have a high permeability and the sand that is present is sensitive to softening. There is a risk of weak to strong transition zones due to the possibility of pot clay being present. The groundwater is salt, which may pose a risk for the properties of the drilling fluid but no further risks are identified with relation to the groundwater.

A.2.9. Formation of Appelscha (AP)

General
The Formation of Appelscha is a fluvial formation deposited by the eastern rivers during the Bavelian and the early Cromerian, roughly from 1.07 to 0.465 MA ago. In general it consists of a light gray to light yellow, moderately fine to very coarse sand, with a possible large gravel fraction or it consists of a fine to very coarse gravel with in the fine fraction relatively a lot of gravels with a specific origin for example Thuringian Forest porphyry or basanite.

The top boundary with the Urk Formation is determined by the mineral content of the gravel fraction, a change in the variety of colors and a change in chalk content. The bottom boundary with the Formation of Peize is also determined based on a change in the content of the gravel fraction.

Local Occurrence
The Formation of Appelscha is only described on the Den Helder side since the drilling on the Texel side didn’t reach the required depths. In this borehole it is described as a gray, moderately to very coarse, weak silty sand with some shells and white quartz grains. Sieve tests indicate a weak silty, weak gravelly, fine to coarse sand.

Risks
This formation can be coarse and very permeable as well as highly variable. Obstacles might be present and the groundwater is salt, which may pose a risk for the properties of the drilling fluid but no further risks are identified with relation to the groundwater.

A.2.10. Formation of Peize (PZ)

General
The Formation of Peize is a fluvial formation deposited by the eastern rivers from the Reuverian to the start of the Waalian, roughly from 3 to 1.20 MA ago. It mainly consists of a light gray to light yellow, moderately fine to very coarse sand, with a fine to moderately coarse gravel fraction. The formation doesn’t contain chalk which can be used to separate it from the surrounding formations.

The top boundary with the Formation of Appelscha can be identified by a change of content in the gravel fraction. The boundary with other formations is often identified based on the chalk content or color. The bottom boundary with the Formation of Breda, Formation of Oosterhout or the Formation of Maassluis, can be determined based on amongst others a change in mineral content, chalk percentage or a change in grain size.
Local Occurrence
Based the borings the Formation of Peize is not well recognized at the project site, it is not found in the Texel boring and with a remark that it also could be Formation of Appelscha in the Den Helder boring. It is described as a light gray to gray, moderately to very coarse, weak silty sand. In the sieve tests it is presented as a coarse sand.

Risks
This formation can be coarse and very permeable. Obstacles might be present and the groundwater is salt, which may pose a risk for the properties of the drilling fluid and locally over pressured aquifers can be found, which can cause seepage into the drilled borehole and local borehole instability.

A.3. Pre-Existing Model
TNO created a model, GeoTOP, of the dutch soils to -50m NAP with a grid size of 100x100m and a vertical resolution of 0.5m. This model currently covers parts of the Netherlands but the model is for the Marsdiep area. While comparing the cross-section made for the Marsdiep project, figure 3.1, with the cross-section from TNO, figure A.1 it becomes clear that neither of the cross-sections is complete. Since the cross-section for this project is based on additional information, seismic data, Cone Penetration Tests (CPTs) and boreholes, this is used as a basis in this research.

Figure A.1: Cross-section made with GeoTOP, available through DINOliset.nl, based on a collection of subsurface data, (TNO, 2013c), with Texel on the left and Den Helder on the right side.
Initial Site Investigations, Proposals and Calculations

This appendix contains information about the initial Site Investigations (SIs), proposals and the most relevant results of the preliminary calculations.

B.1. Initial Site Investigation

The positions of the initial SIs can be found in table B.1 and table B.2. The coordinates are in the dutch coordinate system.

Table B.1: Positions of the SIs in Texel.

<table>
<thead>
<tr>
<th>Texel</th>
<th>RD Coordinates [m]</th>
<th>Ground level [mNAP]</th>
<th>Depth [mNAP]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drilling B1</td>
<td>115034,0 557492,0</td>
<td>0,72</td>
<td>- 82,8</td>
</tr>
<tr>
<td>CPT DKMP1</td>
<td>115074,0 557455,0</td>
<td>0,93</td>
<td>- 73,1</td>
</tr>
<tr>
<td>CPT DKMP2</td>
<td>115068,0 557505,0</td>
<td>0,48</td>
<td>- 44,2</td>
</tr>
<tr>
<td>CPT DKMP3</td>
<td>115063,0 557555,0</td>
<td>0,42</td>
<td>- 44,4</td>
</tr>
<tr>
<td>CPT DKMP4</td>
<td>115057,0 557605,0</td>
<td>0,42</td>
<td>- 44,4</td>
</tr>
</tbody>
</table>

Table B.2: Positions of the SIs in Den Helder.

<table>
<thead>
<tr>
<th>Den Helder</th>
<th>RD Coordinates [m]</th>
<th>Ground level [mNAP]</th>
<th>Depth [mNAP]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drilling B2</td>
<td>115613,0 553189,0</td>
<td>3,95</td>
<td>- 99,30</td>
</tr>
<tr>
<td>CPT DKM5</td>
<td>115626,1 553185,0</td>
<td>5,57</td>
<td>- 36,5</td>
</tr>
<tr>
<td>CPT DKM5A</td>
<td>115627,1 553180,9</td>
<td>5,53</td>
<td>- 65,8</td>
</tr>
<tr>
<td>CPT DKM6</td>
<td>115636,8 553095,0</td>
<td>3,97</td>
<td>- 40,8</td>
</tr>
<tr>
<td>CPT DKM7</td>
<td>115648,0 553041,0</td>
<td>4,26</td>
<td>- 40,6</td>
</tr>
<tr>
<td>CPT DKM8</td>
<td>115659,0 552978,0</td>
<td>4,15</td>
<td>- 41,3</td>
</tr>
</tbody>
</table>

B.2. Drilling Proposals

This section contains the initial proposal for the pilot drilling. A quick overview of the alignments is presented in figure B.1.


B.2.1. Texel - Proposed

From the Texel side one or more pilot drillings will be attempted. At a depth of about NAP-35m to NAP-40m a layer consisting of coarse and gravelly sand or boulder clay (Drente Formation) can be encountered. The local composition, thickness and exact location are all relatively uncertain based on the results of the SIs performed in Texel.

At the end of each drilling a stop is proposed which should last for 4 days, during which there would be no flow or rotation in the borehole. Afterwards the torque and pressure necessary to get it going again are going to be measured to get an idea of the borehole stability. Due to the fact that this is the moment that the most equipment would be down-hole and thus the loss is the largest a compromise is made. Based on a graph of the maximum torque and tensile load, see figure B.2 the equipment could handle and the amount at the moment different scenarios are proposed for the stop.

- Scenario 1, both torque and tensile load are smaller than the 40% limit.
- Scenario 2, between the 40% and the 60% limit.
- Scenario 3, between the 60% and the 80% limit.
- Scenario 4, larger than the 80% limit.

Depending on the scenario the duration and start procedure are different. In general the drillbit would be pulled back from the face before reducing the flow to 0 L/min. At the start the flow would be brought to 1500 L/min in about 15 minutes before rotation would start or the other way around depending on the scenario. In all cases the torque would be measures while rotating at low speeds (5-10 Rotations Per Minute (RPM)) and no flow. The method for pulling back is the same, once the torque is below a threshold value it would start, nine pipes would be pulled while rotating but the tenth would be pulled statically.
PB1T-A
Due to the depth this boring will not be executed. The risk on fractures and blowouts would be too large. See subsection B.5 in appendix B for more information.

PB1T-B
This is the first boring that is going to be executed. The length in total will be about 2500m and the depth of the horizontal part will be about NAP-65m. This boring will encounter the "valley fill" which presumably consists of coarse gravelly sand with possible occurrence of boulders as layers at the bottom of old filled trenches or isolated higher on in the formation. If this boring doesn’t succeed after a few attempts, the next depth will be drilled.

PB1T-C1
This boring has a total length of about 1500m and the depth of the horizontal part will be about NAP-85m. It is currently unsure whether the Pot Clay will be encountered. If the Pot Clay is present as a large, thick layer beneath the valley fill, then after consultation and in agreement with PWN the next alignment, PB1T-C2, will be executed.

PB1T-C2
This boring has a total length of about 1500m and the depth of the horizontal part will be about NAP-100m. This boring will cut through the possibly present Pot Clay and end in the coarse gravelly sand formations (Formation of Peize, Appelscha) from the Pliocene or Early Pleistocene.

B.2.2. Den Helder - Proposed
Which of the proposed pilot drillings from the Den Helder side will be executed depends on the results from the borings on the Texel side. The borings starting in Den Helder will encounter beneath the Holocene top layers two clay layers. The first layer at a depth of approximately NAP-35m to NAP-40m can consist of coarse and gravelly sand or boulder clay (Drente Formation). The local composition, thickness and exact location are all uncertain. The second layer is found at about NAP-47m to NAP-55m and is probably made of Pot Clay (Peelo Formation, the Nieuwolda section). Before encountering the Eem formation at a depth of approximately NAP-40m a stop is proposed to test the borehole stability and the erosion susceptibility of the possible loosely packed top layers.
During the first few hours the flow will be maintained at a lower volume (500L/min) while no progress is made, in total the stop will last 2 days.

During the boring the effect of the Rate Of Penetration (ROP) shall be tested as well. In the horizontal part at a certain length the ROP will be reduced from 3m/min to 1.5m/min, in order to investigate what the effect is on the return flow, pressures and borehole stability.

**PB1DH-A1**
This boring has a total length of about 1000m and the depth of the horizontal part will be about NAP-65m. It ends in a layer consisting of mostly moderately fine sand, possibly loosely packed but coarse sand with clay layers can be encountered as well (melt water deposits of the Peelo Formation).

**PB1DH-A2**
This boring has a total length of about 2000m and the depth of the horizontal part will be at a depth based on the results of the borings on the Texel side of the Marsdiep.

### B.3. Boundary Conditions

Before taking a look at the calculations the boundary conditions have to be determined. Based on the dutch standards and the proposed alignment, see figure 3.4, the following boundary conditions can be determined (Nugroho and van Meerten, 2015):

- Dikes need to be crossed at a minimum depth of 10m (NEN 3651, 2012). By starting 100m away from the dikes and at an angle of 12°, the crossing is at a depth of -21.3m relative to the starting height.
- The pressure in the borehole is not allowed to exceed the strength of the surrounding soil, in order to prevent fracturing and blow-out.
- The overpressure during stand still is at least 15 kPa and preferable 20 kPa, to ensure hole stability.

### B.4. Parameters, Dimensions and Variables

Besides the boundary conditions the values of the following parameters either had to be estimated or are given by the client, PWN.

#### B.4.1. Geometry of the boring

- The entry and exit angle is 12°.
- The radius of the corners are 1000 m, in order to enable the contractor to construct pipes up to 0.194m or to be exact 7 5/8” in diameter.
- 3 options for construction depths, in other words depth of the horizontal part are considered; NAP-45m, NAP-65m and NAP-100m, after the calculations this is changed to NAP-65 m, NAP-85 m and NAP-100 m.
- Radius of corner at entry point is about 1000*diameter of the pipe.

#### B.4.2. Product pipe

- The pipe has an outside diameter of 278 mm, proposed by PWN.
- Wall thickness of 9 mm.
- Material is steel.
- The yield strength of the pipe is 445 MPa (FE445).
- The volumetric weight is 78.5 kN/m³.
- Design pressure and test pressure are to be determined later.

#### B.4.3. Execution

For the execution two scenarios are taken into account, for both scenarios the meeting in the middle method will be used due to the length of the alignment.

- Scenario 1; pilot boring with a drill bit of 311 mm or 12.25”, reamer of 406mm or 16” and pull in afterward.
- Scenario 2; pilot boring with a drill bit of 406 mm or 16” and pull in afterwards.

The following specifications are used in the different phases of the project, see section 2.3 for more information.
Pilot phase
- The diameter of the borehole is 331 mm (scenario 1) or 410 mm (scenario 2).
- The outer diameter of the drill string is 194 m or 7 5/8”.
- The wall thickness of the drill string is 10.2 mm.
- Flow rate of return fluid is 2000 liter/m.
- Circulation loss is 30%.
- The ROP is 0.05 m/s or 3 m/min.

Reaming phase and pull in phase
- The diameter of the borehole is 410 mm.
- The outer diameter of the drill string is 194 m or 7 5/8”.
- The wall thickness of the drill string is 10.2 mm.
- Flow rate of return fluid is 2000 liter/m.
- Circulation loss is 20%.
- The ROP is 0.25 m/s or 15 m/min.

B.4.4. Drilling fluid

Initial
- The volumetric weight is 10.5 kN/m³.
- The plastic viscosity is 0.014 Pa.s.
- The yield stress is 11 Pa.

Normative of the return flow
- The volumetric weight is 11.1 kN/m³.
- The plastic viscosity is 0.014 Pa.s.
- The yield stress is 14 Pa.

B.4.5. Coefficient of friction
Determined in accordance with the NEN 3650-1:2012 for determining the pulling force between:
- Roller and pipe (f1) is 0.10.
- Pipe and drilling fluid (f2) is 50 Pa. This is the pressure loss caused by friction.
- Pipe and soil (f3) is 0.20.

B.4.6. Other
- The volumetric weight of the sea water is 10.1 kN/m³.
- The average sea level is set on NAP+0m for the preliminary design.
- The dry density of the clay in the drilling fluid is 2650 kg/m³.

B.5. Results
In this section the results of the preliminary calculations can be found. Most of the calculations are made using DGeoPipeline.

B.5.1. Minimal Drilling Fluid Pressure
A summary can be found in table B.3. From these calculations it can be concluded that NAP-45m is not a feasible solution is since for both scenarios this results in fracturing and for scenario 1 blowouts occur as well. At a depth of NAP-65m fracturing still occurs in scenario 1 but not in scenario 2 and blowouts don’t occur at all. Neither fracturing nor blowouts occur at NAP-100m.
Table B.3: Summary of the calculations made for the drilling fluid pressures.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Design Depth</th>
<th>$P_{min}^{pi} &gt; P_{fr}$</th>
<th>$P_{max}$</th>
<th>$P_{min}^{re} &gt; P_{max}$</th>
<th>$P_{fr}$</th>
<th>$P_{max}$</th>
<th>$P_{min}^{pu} &gt; P_{max}$</th>
<th>$P_{fr}$</th>
<th>$P_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NAP-45m</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>1</td>
<td>NAP-65m</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>1</td>
<td>NAP-100m</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>NAP-45m</td>
<td>Yes</td>
<td>No</td>
<td>NA</td>
<td>NA</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>NAP-65m</td>
<td>No</td>
<td>No</td>
<td>NA</td>
<td>NA</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

$P_{min}$ is the minimum required drilling fluid pressure in pilot phase (pi), reaming phase (re) and pull in phase (pu).

$P_{fr}$ is the maximum drilling fluid pressure before fracturing occurs.

$P_{max}$ is the maximum drilling fluid pressure before blowout occurs.

NA means Not Applicable, since there is no reaming phase in scenario 2.

### B.5.2. Torque and Normal Force

For the torque and normal force required on the drilling rig two cases are evaluated, a favorable and unfavorable case. In the favorable case the friction coefficient is assumed to be 0.2 and for the unfavorable case a friction coefficient of 0.3 is used based on previous experiences. Since the meeting in the middle point is not decided the calculation gives an overestimation because the whole length is used. A summary of the results is presented in table B.4.

Table B.4: Summary of the calculations made with regard to the torque and normal force.

<table>
<thead>
<tr>
<th>Case</th>
<th>Scenario</th>
<th>Phase</th>
<th>$T_{min}$ [kNm]</th>
<th>$T_{max}$ [kNm]</th>
<th>$N_{min}$ [kN]</th>
<th>$N_{max}$ [kN]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Favorable</td>
<td>1</td>
<td>Pilot</td>
<td>60</td>
<td>70</td>
<td>900</td>
<td>1100</td>
</tr>
<tr>
<td>Favorable</td>
<td>1</td>
<td>Reaming</td>
<td>70</td>
<td>80</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Favorable</td>
<td>2</td>
<td>Pilot</td>
<td>60</td>
<td>70</td>
<td>1100</td>
<td>1300</td>
</tr>
<tr>
<td>Unfavorable</td>
<td>1</td>
<td>Pilot</td>
<td>80</td>
<td>90</td>
<td>1700</td>
<td>1900</td>
</tr>
<tr>
<td>Unfavorable</td>
<td>1</td>
<td>Reaming</td>
<td>90</td>
<td>100</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Unfavorable</td>
<td>2</td>
<td>Pilot</td>
<td>80</td>
<td>90</td>
<td>2250</td>
<td>2400</td>
</tr>
</tbody>
</table>

$T$ for Torque, $N$ for Normal or pressing force and NA for Not Applicable.

### B.5.3. Rate Of Penetration (ROP)

The ROP is important for a number of reasons but mainly because it influences the properties of the drilling fluid. The change in properties of the drilling fluid is determined by the ROP, volume of excavated soil and flow rate of the return fluid. In tables B.5, B.6 and B.7 the results can be found of calculation based on different amounts of drilling fluid, different penetration rates and the two scenarios. To maintain the return flow and make progress in practice often these boundary values are used.

- The density $\rho < 1130 \text{ kg/m}^3$
- The plastic viscosity $\mu < 0.04 \text{ Pa.s}$
- The yield stress $\tau < 14 \text{ Pa}$
- The sand content $\leq 15\%$

When taking this into account it is presented that in scenario 2 the density becomes higher than the boundary value for any of the calculated cases. The recommendations (Nugroho and van Meerten, 2015) are:

- Use a flow rate of 2000 l/min and in order to maintain the density the penetration rate shouldn’t be too high.
- The drilling fluid should be chosen in accordance to the different soils at different depths.
### Table B.5: Changing fluid parameters at different penetration rates and flow rates. Scenario 1, pilot phase.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1500</td>
<td>3.00</td>
<td>1224</td>
<td>0.015</td>
<td>14.19</td>
<td>8.42</td>
</tr>
<tr>
<td></td>
<td>1.50</td>
<td>1153</td>
<td>0.014</td>
<td>12.52</td>
<td>4.5</td>
</tr>
<tr>
<td>2000</td>
<td>3.00</td>
<td>1190</td>
<td>0.015</td>
<td>13.34</td>
<td>6.53</td>
</tr>
<tr>
<td></td>
<td>1.50</td>
<td>1133</td>
<td>0.013</td>
<td>12.13</td>
<td>3.44</td>
</tr>
<tr>
<td>2500</td>
<td>3.00</td>
<td>1168</td>
<td>0.014</td>
<td>12.84</td>
<td>5.33</td>
</tr>
<tr>
<td></td>
<td>1.50</td>
<td>1121</td>
<td>0.013</td>
<td>11.90</td>
<td>2.78</td>
</tr>
<tr>
<td>3000</td>
<td>3.00</td>
<td>1153</td>
<td>0.014</td>
<td>12.52</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>1.50</td>
<td>1113</td>
<td>0.013</td>
<td>11.74</td>
<td>2.33</td>
</tr>
</tbody>
</table>

### Table B.6: Changing fluid parameters at different penetration rates and flow rates. Scenario 1, reaming phase.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1500</td>
<td>3.00</td>
<td>1199</td>
<td>0.015</td>
<td>13.55</td>
<td>7.02</td>
</tr>
<tr>
<td></td>
<td>1.50</td>
<td>1138</td>
<td>0.013</td>
<td>12.23</td>
<td>3.71</td>
</tr>
<tr>
<td>2000</td>
<td>3.00</td>
<td>1170</td>
<td>0.014</td>
<td>12.88</td>
<td>5.41</td>
</tr>
<tr>
<td></td>
<td>1.50</td>
<td>1122</td>
<td>0.013</td>
<td>11.91</td>
<td>2.83</td>
</tr>
<tr>
<td>2500</td>
<td>3.00</td>
<td>1151</td>
<td>0.014</td>
<td>12.49</td>
<td>4.41</td>
</tr>
<tr>
<td></td>
<td>1.50</td>
<td>1112</td>
<td>0.013</td>
<td>11.73</td>
<td>2.28</td>
</tr>
<tr>
<td>3000</td>
<td>3.00</td>
<td>1138</td>
<td>0.013</td>
<td>12.22</td>
<td>3.71</td>
</tr>
<tr>
<td></td>
<td>1.50</td>
<td>1105</td>
<td>0.013</td>
<td>11.60</td>
<td>1.91</td>
</tr>
</tbody>
</table>

### Table B.7: Changing fluid parameters at different penetration rates and flow rates. Scenario 2.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1500</td>
<td>3.00</td>
<td>1323</td>
<td>0.019</td>
<td>17.23</td>
<td>13.79</td>
</tr>
<tr>
<td></td>
<td>1.50</td>
<td>1212</td>
<td>0.015</td>
<td>13.86</td>
<td>7.73</td>
</tr>
<tr>
<td>2000</td>
<td>3.00</td>
<td>1271</td>
<td>0.017</td>
<td>15.48</td>
<td>10.93</td>
</tr>
<tr>
<td></td>
<td>1.50</td>
<td>1180</td>
<td>0.014</td>
<td>13.11</td>
<td>5.96</td>
</tr>
<tr>
<td>2500</td>
<td>3.00</td>
<td>1236</td>
<td>0.016</td>
<td>14.49</td>
<td>9.05</td>
</tr>
<tr>
<td></td>
<td>1.50</td>
<td>1160</td>
<td>0.014</td>
<td>12.66</td>
<td>4.87</td>
</tr>
<tr>
<td>3000</td>
<td>3.00</td>
<td>1212</td>
<td>0.015</td>
<td>13.86</td>
<td>7.73</td>
</tr>
<tr>
<td></td>
<td>1.50</td>
<td>1146</td>
<td>0.014</td>
<td>12.38</td>
<td>4.11</td>
</tr>
</tbody>
</table>
B.5.4. **Infiltration and loss of drilling fluid**

Because the pressure maintained in the borehole will be higher than the surrounding pressure, during standstill and during drilling, the drilling fluid will infiltrate the surroundings. The results of the calculations can be found in table B.8 for the pilot phase and table B.9 for the reaming phase. The soil parameters of the valley fill are estimated so the calculated risks might be different. The deeper coarse sand layers might pose a bigger risk since the values for the permeability derived from the sieve curve are relatively small for coarse sand/ fine gravel layers.

**Table B.8: The drilling fluid loss, per layer in the pilot phase.**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Scenario 1 [L/m drilling]</th>
<th>Scenario 2 [L/m drilling]</th>
<th>Risk in scenario 1</th>
<th>Risk in scenario 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Na</td>
<td>15</td>
<td>5</td>
<td>Slight</td>
<td>Slight</td>
</tr>
<tr>
<td>Bx</td>
<td>20</td>
<td>10</td>
<td>Slight</td>
<td>Slight</td>
</tr>
<tr>
<td>EE-BB</td>
<td>25</td>
<td>15</td>
<td>Slight</td>
<td>Slight</td>
</tr>
<tr>
<td>Dr</td>
<td>40</td>
<td>20</td>
<td>Slight</td>
<td>Slight</td>
</tr>
<tr>
<td>EE-URTY</td>
<td>30</td>
<td>25</td>
<td>Slight</td>
<td>Slight</td>
</tr>
<tr>
<td>Val_a</td>
<td>380</td>
<td>210</td>
<td>High</td>
<td>Moderate</td>
</tr>
<tr>
<td>Val_b</td>
<td>400</td>
<td>215</td>
<td>High</td>
<td>Moderate</td>
</tr>
<tr>
<td>PENI</td>
<td>NA</td>
<td>NA</td>
<td>Slight</td>
<td>Slight</td>
</tr>
<tr>
<td>PE</td>
<td>40</td>
<td>30</td>
<td>Slight</td>
<td>Slight</td>
</tr>
<tr>
<td>URVE</td>
<td>25</td>
<td>20</td>
<td>Slight</td>
<td>Slight</td>
</tr>
<tr>
<td>AP</td>
<td>100</td>
<td>60</td>
<td>Moderate</td>
<td>Moderate</td>
</tr>
<tr>
<td>PZ</td>
<td>500</td>
<td>260</td>
<td>High</td>
<td>Moderate</td>
</tr>
</tbody>
</table>

NA stand for Not Applicable, since the PENI layer is a swelling clay.

**Table B.9: Drilling fluid loss, per layer in the reaming phase.**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Scenario 1 [L/m drilling]</th>
<th>Risk in scenario 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Na</td>
<td>5</td>
<td>Slight</td>
</tr>
<tr>
<td>Bx</td>
<td>10</td>
<td>Slight</td>
</tr>
<tr>
<td>EE-BB</td>
<td>20</td>
<td>Slight</td>
</tr>
<tr>
<td>Dr</td>
<td>20</td>
<td>Slight</td>
</tr>
<tr>
<td>EE-URTY</td>
<td>25</td>
<td>Slight</td>
</tr>
<tr>
<td>Val_a</td>
<td>280</td>
<td>High</td>
</tr>
<tr>
<td>Val_b</td>
<td>300</td>
<td>High</td>
</tr>
<tr>
<td>PENI</td>
<td>NA</td>
<td>Slight</td>
</tr>
<tr>
<td>PE</td>
<td>25</td>
<td>Slight</td>
</tr>
<tr>
<td>URVE</td>
<td>20</td>
<td>Slight</td>
</tr>
<tr>
<td>AP</td>
<td>100</td>
<td>Moderate</td>
</tr>
<tr>
<td>PZ</td>
<td>360</td>
<td>Moderate</td>
</tr>
</tbody>
</table>

NA stand for Not Applicable, since the PENI layer is a swelling clay.

B.5.5. **Transport Capacity**

The transport capacity is calculated based on the maximum diameter of the grains the fluid can transport based on the yield stress. The normative size of the grains is not allowed to surpass the calculated value in order for the grains to be transported by the drilling fluid. The cases of two different yield stresses are considered; 11 Pa and 14 Pa. The results is presented in table B.10 and table B.11 respectively. The case where the yield stress is 14 Pa is the most feasible. This is however very close to the boundary value as given in the part on ROP in this section. If it will suffice to transport the soil found in the valley fills, needs to be investigated in the pilot drilling. It is noticeable that it becomes less feasible with depth with especially the layers below NAP-70m that can cause problems.
Table B.10: The maximum grain size for transport, yield stress = 11 Pa.

<table>
<thead>
<tr>
<th>Layer</th>
<th>(d_{90}) [mm]</th>
<th>(d_{\text{max}}) [mm]</th>
<th>Risk for transport</th>
</tr>
</thead>
<tbody>
<tr>
<td>Na</td>
<td>0.49</td>
<td>0.70</td>
<td>Slight</td>
</tr>
<tr>
<td>Bx</td>
<td>0.32</td>
<td></td>
<td>Slight</td>
</tr>
<tr>
<td>EE-BB</td>
<td>0.42</td>
<td></td>
<td>Slight</td>
</tr>
<tr>
<td>Dr</td>
<td>0.51</td>
<td></td>
<td>Slight</td>
</tr>
<tr>
<td>EE-URY</td>
<td>0.58</td>
<td></td>
<td>Moderate</td>
</tr>
<tr>
<td>Val_a</td>
<td>0.74</td>
<td></td>
<td>Moderate</td>
</tr>
<tr>
<td>Val_b</td>
<td>1.29</td>
<td></td>
<td>High</td>
</tr>
<tr>
<td>PENI</td>
<td>0.07</td>
<td></td>
<td>Slight</td>
</tr>
<tr>
<td>PE</td>
<td>1.29</td>
<td></td>
<td>High</td>
</tr>
<tr>
<td>URVE</td>
<td>0.52</td>
<td></td>
<td>Slight</td>
</tr>
<tr>
<td>AP</td>
<td>1.50</td>
<td></td>
<td>High</td>
</tr>
<tr>
<td>PZ</td>
<td>1.50</td>
<td></td>
<td>High</td>
</tr>
</tbody>
</table>

Table B.11: The maximum grain size for transport, yield stress = 14 Pa.

<table>
<thead>
<tr>
<th>Layer</th>
<th>(d_{90}) [mm]</th>
<th>(d_{\text{max}}) [mm]</th>
<th>Risk for transport</th>
</tr>
</thead>
<tbody>
<tr>
<td>Na</td>
<td>0.49</td>
<td>0.90</td>
<td>Slight</td>
</tr>
<tr>
<td>Bx</td>
<td>0.32</td>
<td></td>
<td>Slight</td>
</tr>
<tr>
<td>EE-BB</td>
<td>0.42</td>
<td></td>
<td>Slight</td>
</tr>
<tr>
<td>Dr</td>
<td>0.51</td>
<td></td>
<td>Slight</td>
</tr>
<tr>
<td>EE-URY</td>
<td>0.58</td>
<td></td>
<td>Slight</td>
</tr>
<tr>
<td>Val_a</td>
<td>0.74</td>
<td></td>
<td>Slight</td>
</tr>
<tr>
<td>Val_b</td>
<td>1.29</td>
<td></td>
<td>High</td>
</tr>
<tr>
<td>PENI</td>
<td>0.07</td>
<td></td>
<td>Slight</td>
</tr>
<tr>
<td>PE</td>
<td>1.29</td>
<td></td>
<td>High</td>
</tr>
<tr>
<td>URVE</td>
<td>0.52</td>
<td></td>
<td>Slight</td>
</tr>
<tr>
<td>AP</td>
<td>1.50</td>
<td></td>
<td>High</td>
</tr>
<tr>
<td>PZ</td>
<td>1.50</td>
<td></td>
<td>High</td>
</tr>
</tbody>
</table>
B.5.6. Height of Mound

In order to see if the entry location is high enough to withstand the fluid pressures in the borehole, calculations are made taking a high sea level into account, NAP+1.5m. Volumetric weights of 10.5 and 10.1 kN/m\(^3\) are taken into account for the drilling fluid and salt sea water in the surroundings. Two cases are considered, an overpressure of 15 and 20 kPa. The results is presented in table B.12 and table B.13 respectively.

Table B.12: Required mound height, with high water and an overpressure of 15 kPa.

<table>
<thead>
<tr>
<th>Depth in the axis of the alignment [mNAP]</th>
<th>Drilling fluid pressure incl. overpressure [kPa]</th>
<th>Required drilling fluid level [mNAP]</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAP-6m (bottom of casing)</td>
<td>81</td>
<td>+ 1.73</td>
</tr>
<tr>
<td>NAP-23.5m</td>
<td>268</td>
<td>+ 1.98</td>
</tr>
<tr>
<td>Drilling to NAP-45m:</td>
<td>485</td>
<td>+ 1.16</td>
</tr>
<tr>
<td>Drilling to NAP-65m:</td>
<td>687</td>
<td>+ 0.40</td>
</tr>
<tr>
<td>Drilling to NAP-100m:</td>
<td>1040</td>
<td>- 0.94</td>
</tr>
</tbody>
</table>

Table B.13: Required mound height, with high water and an overpressure of 20 kPa.

<table>
<thead>
<tr>
<th>Depth in the axis of the alignment [mNAP]</th>
<th>Drilling fluid pressure incl. overpressure [kPa]</th>
<th>Required drilling fluid level [mNAP]</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAP-6m (bottom of casing)</td>
<td>86</td>
<td>+ 2.21</td>
</tr>
<tr>
<td>NAP-23.5m</td>
<td>272</td>
<td>+ 2.45</td>
</tr>
<tr>
<td>Drilling to NAP-45m:</td>
<td>490</td>
<td>+ 1.63</td>
</tr>
<tr>
<td>Drilling to NAP-65m:</td>
<td>692</td>
<td>+ 0.87</td>
</tr>
<tr>
<td>Drilling to NAP-100m:</td>
<td>1045</td>
<td>- 0.46</td>
</tr>
</tbody>
</table>

The existing heights of the entry points are NAP+0.42m for Texel and NAP+4.15m for Den Helder. Comparing the required and existing height it becomes clear that the entry site in Den Helder is sufficient while the entry point in Texel requires an artificial mound to the height of NAP+2m or NAP+2.5m depending on the required overpressure.

B.5.7. Casing Length

At both entry sites the top layers consist of fine sand. In order to reduce the risks of blow-outs and prevent borehole erosion a casing is recommended. The required depth in Den Helder is NAP-6m, the entry angle is 12° and the entry point height is NAP+4.15m. A casing of about 50m in length is required. In Texel a depth of NAP-5.5m is required, the entry angle is 12° and the entry point height is NAP+0.42m. A casing with a minimum length of 30m is required, the height of the mound is not included since it is unknown at this moment.

B.5.8. Other

In the report by Nugroho and van Meerten a number of other calculations are made as well, for example the pulling force and the maneuverability. But since they’re not related to the topic of this thesis the calculations are excluded.
C

Equipment and Tests

This appendix contains brief explanations and illustrations of the laboratory equipment that is used and the tests that are performed. It contains a brief section on different Horizontal Directional Drilling (HDD) equipment as well.

C.1. Laboratory Equipment

C.1.1. Marsh Funnel
The Marsh Funnel is a funnel with a certain shape and the exit has a specific diameter. There is a mesh (partially) covering the top of the funnel to avoid the exit becoming blocked by larger grains. It is used by filling the funnel to a given height and recording the time it takes for the fluid to fill a cup till a given level. This time can be used as an indication for the viscosity.

C.1.2. (Roto) Viscometer, (Fann) Rheometer
The Viscometer determines the shear rate and shear stress of fluids at various times at atmospheric pressure. It works by rotating a cylinder in the fluid around a free hanging weight, thus generating a shear rate. This rate is transferred onto the weight which will rotate to a certain extend, the resulting angle can be measured and used to gain other parameters as well. For example the plastic velocity is the angle at 600 rpm minus the angle at 300 rpm and the yield point is the angle at 300 rpm minus the plastic viscosity. In figure C.1a the viscometer that is used in Texel is presented. The following settings of the device are used; 600, 300, 100, 60, 30, 6 and 3 Rotations Per Minute (RPM). Since the device is sensitive the drilling fluid first had to be sieved in order to be used in this device. Otherwise the cuttings could block or damage the device.
The Mud Balance consists of a fixed volume cup attached to a lever, see figure C.1b. The cup is filled to a certain level, afterwards a lid containing a valve is attached and using a small pump more fluid is added, resulting in a slightly pressured fluid in which potential gas bubbles are very small and thus have very little influence upon the accuracy of the measurement. The lever is put with a fixed point on a balance and by moving a weight along the lever the density can be determined by reading it from the lever once the lever is in equilibrium.

The Mud Balance

The Mud Balance consists of a fixed volume cup attached to a lever, see figure C.1b. The cup is filled to a certain level, afterwards a lid containing a valve is attached and using a small pump more fluid is added, resulting in a slightly pressured fluid in which potential gas bubbles are very small and thus have very little influence upon the accuracy of the measurement. The lever is put with a fixed point on a balance and by moving a weight along the lever the density can be determined by reading it from the lever once the lever is in equilibrium.

C.1.4. Electrical Conductivity Meter

This device is used to get an indication of the salt present in the drilling fluid. While it can be correlated to the amount of salt it is not used for that purpose but rather to check for trends. The device actually measures the conductivity and that is not only caused by the amount of salt present but also by the amount of nutrients or impurities in the fluid. The meter depicted in figure C.1c is used by putting the electrode into the drilling fluid and waiting till the value on the screen has stabilized.

C.1.5. Sand Content Kit

A simple method is used to determine the amount of sand in the return fluid. All parts are depicted in figure C.2. The glass vial is filled to a certain marked level and then further filled with water. The vial is than emptied in the white cylinder which contains a sieve, the bottle is repeatedly filled with water and emptied through the sieve until it is clean. The white funnel is placed on the side where the vial is emptied and the sieve is than rotated and cleaned upside down, everything that is on the sieve goes through the funnel into the vial again. Once the sieve is empty, the vial is shaken and everything that settles within a few seconds is assumed to be sand.
C.1.6. **Sievetest**

There is not enough time in between taking samples in Texel to do a full sieve test, so an adapted version is conducted to get an indication of ratio between different size-fractions. A liter of the drilling fluid is put in the sieves, which are stacked from coarse to fine (5.6 mm, 2 mm, 1 mm, 0.6 mm, 0.355 mm, 0.212 mm and 0.063 mm). The vibrating table is turned on for 10-15 minutes. Afterwards the sieves are washed and put down on a white board to get an indication of the ratio between the amounts what is on the different sieves. The sieved drilling fluid is used as for the viscometer. The sieve equipment used is displayed in figure C.3. The sieve test on site is not only conducted to check for changes but the resulting sieved fluid is also required for the viscometer, since grains can damage the device and distort the results of the viscometer tests. A time effect is noted in how long after sieving the viscometer test is performed. Additional research should be conducted to determine the magnitude of this effect as well as the influence of a vibrating table, used in the sieve test, on the measured fluid properties. This effect is also noted when testing a sample from the clean inflow, before and after sieving on the vibrating table.

![Figure C.3: The sieve equipment used in Texel.](image-url)
C.1.7. **pH Meter**
In Den Helder a pH Meter is used since Teqgel requires a higher pH for optimum properties. This is discovered during the first drilling in Den Helder so the pH is monitored during the second drilling.

C.2. **HDD Equipment**
A distinction is often made in HDD rigs on size. Although there is no hard answer to what constitutes a mini-HDD, midi-HDD or maxi-HDD, this distinction is often used, amongst others by (Bueno, 2012; Denekamp et al., 2000; Baik et al., 2002). As an example the definitions given by the Plastics Pipe Institute (PPI) in their 2009 report "Guidelines for Use of Mini-Horizontal Directional Drilling for Placement of High Density Polyethylene Pipe – TR-46" as cited in (Bueno, 2012) are found in table C.1.

<table>
<thead>
<tr>
<th></th>
<th>Length [m]</th>
<th>Depth [m]</th>
<th>Diameter [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mini-HDD</td>
<td>&lt;200</td>
<td>&lt;5</td>
<td>&lt;0.3</td>
</tr>
<tr>
<td>Midi-HDD</td>
<td>&lt;300</td>
<td>&lt;25</td>
<td>&lt;0.6</td>
</tr>
<tr>
<td>Maxi-HDD</td>
<td>&gt;300</td>
<td>&gt;25</td>
<td>&gt;0.6</td>
</tr>
</tbody>
</table>

Table C.1: Definitions for different HDD rig classifications (Bueno, 2012).

In addition, for the Mini-HDD classification the following limits are given with respect to the maximum Pull Forces and Torque, approximate 90 KN and 1.3 KNm respectively. The definitions that are present in (Denekamp et al., 2000) and (Baik et al., 2002) are slightly different. In (Denekamp et al., 2000) an additional distinction in midi-rigs, between small and large midi-rigs, is made. It is however important to note that it is partially based on the same parameters length and pipe diameter. But as it is also important to mention that these definitions can vary greatly depending on the source. A general trend is that the equipment and techniques become more sophisticated with an increase in class. In other words; a maxi-HDD rig will have in general a more sophisticated tracking system than a mini-HDD rig.

C.2.1. **Mud Pulse**
To get the data to the surface a mud pulse system is used initially used in Texel. The mud pulse system sends pressure pulses through the drilling fluid in the Drillstring (DS). One initial problem with this system is that the update frequency of the drilling parameters is too low, due to the string of parameters it is sending to the surface. An adjustment is made that would allow the system to continue logging but it would only provide the data required for steering and store the additional data to be read out at the end. Due to failing amongst others to get a pump down gyro tool working, after several attempts and delays the company is replaced after a few days.
This appendix contains additional results.

**D.1. Maximum Diameter for Transport**

Table D.1: The $D_{90}$ is the sample, the $D_{\text{Calc}}$ is the calculated value using equation 5.42.

<table>
<thead>
<tr>
<th>Texel</th>
<th>$D_{90}$ [m]</th>
<th>$D_{\text{Calc}}$ BH [m]</th>
<th>$D_{\text{Calc}}$ HB [m]</th>
<th>$D_{\text{Calc}}$ BH [m]</th>
<th>$D_{\text{Calc}}$ HB [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-65-TXL-03</td>
<td>0.00031</td>
<td>0.002078</td>
<td>0.002017</td>
<td>0.001039</td>
<td>0.001008</td>
</tr>
<tr>
<td>B-65-TXL-04</td>
<td>0.000329</td>
<td>0.001188</td>
<td>0.000839</td>
<td>0.000594</td>
<td>0.00042</td>
</tr>
<tr>
<td>B-65-TXL-07</td>
<td>0.000264</td>
<td>0.001781</td>
<td>0.001644</td>
<td>0.000891</td>
<td>0.000822</td>
</tr>
<tr>
<td>B-65-TXL-09</td>
<td>0.000211</td>
<td>0.001781</td>
<td>0.001599</td>
<td>0.000891</td>
<td>0.000822</td>
</tr>
<tr>
<td>B-65-TXL-10</td>
<td>0.000194</td>
<td>0.001781</td>
<td>0.001644</td>
<td>0.000891</td>
<td>0.000822</td>
</tr>
<tr>
<td>B-65-TXL-12</td>
<td>0.000227</td>
<td>0.001484</td>
<td>0.001259</td>
<td>0.000742</td>
<td>0.000629</td>
</tr>
<tr>
<td>B-65-TXL-13</td>
<td>0.000206</td>
<td>0.001385</td>
<td>0.001185</td>
<td>0.000693</td>
<td>0.000592</td>
</tr>
<tr>
<td>B-65-TXL-15</td>
<td>0.000228</td>
<td>0.001385</td>
<td>0.001185</td>
<td>0.000693</td>
<td>0.000592</td>
</tr>
<tr>
<td>B-65-TXL-16</td>
<td>0.000188</td>
<td>0.001484</td>
<td>0.001259</td>
<td>0.000742</td>
<td>0.000629</td>
</tr>
<tr>
<td>B-65-TXL-28</td>
<td>0.000203</td>
<td>0.001286</td>
<td>0.000948</td>
<td>0.000643</td>
<td>0.000474</td>
</tr>
<tr>
<td>B-65-TXL-30</td>
<td>0.000243</td>
<td>0.001583</td>
<td>0.001343</td>
<td>0.000792</td>
<td>0.000671</td>
</tr>
<tr>
<td>B-65-TXL-32</td>
<td>0.000244</td>
<td>0.001484</td>
<td>0.001343</td>
<td>0.000742</td>
<td>0.000671</td>
</tr>
<tr>
<td>B-65-TXL-35</td>
<td>0.000375</td>
<td>0.001336</td>
<td>0.000908</td>
<td>0.000668</td>
<td>0.000454</td>
</tr>
<tr>
<td>B-65-TXL-40</td>
<td>0.000317</td>
<td>0.000891</td>
<td>0.000622</td>
<td>0.000445</td>
<td>0.000311</td>
</tr>
<tr>
<td>B-65-TXL-48</td>
<td>0.000307</td>
<td>0.001188</td>
<td>0.000948</td>
<td>0.000594</td>
<td>0.000474</td>
</tr>
<tr>
<td>B-65-TXL-51</td>
<td>0.000313</td>
<td>0.001583</td>
<td>0.001301</td>
<td>0.000792</td>
<td>0.000665</td>
</tr>
<tr>
<td>B-65-TXL-74</td>
<td>0.000467</td>
<td>0.001188</td>
<td>0.000829</td>
<td>0.000594</td>
<td>0.000415</td>
</tr>
<tr>
<td>B-65-TXL-80</td>
<td>0.000686</td>
<td>0.001089</td>
<td>0.000741</td>
<td>0.000544</td>
<td>0.00037</td>
</tr>
<tr>
<td>C-85-TXL-07</td>
<td>0.000434</td>
<td>0.001831</td>
<td>0.001644</td>
<td>0.000915</td>
<td>0.000822</td>
</tr>
<tr>
<td>C-85-TXL-09</td>
<td>0.000449</td>
<td>0.00188</td>
<td>0.001733</td>
<td>0.00094</td>
<td>0.000866</td>
</tr>
<tr>
<td>C-85-TXL-25</td>
<td>0.000319</td>
<td>0.00188</td>
<td>0.001735</td>
<td>0.00094</td>
<td>0.000868</td>
</tr>
<tr>
<td>C-85-TXL-31</td>
<td>0.000274</td>
<td>0.002573</td>
<td>0.002345</td>
<td>0.001286</td>
<td>0.001172</td>
</tr>
<tr>
<td>C-85-TXL-43</td>
<td>0.000472</td>
<td>0.001435</td>
<td>0.000691</td>
<td>0.000717</td>
<td>0.000346</td>
</tr>
<tr>
<td>C-85-TXL-45</td>
<td>0.000462</td>
<td>0.001237</td>
<td>0.001049</td>
<td>0.000619</td>
<td>0.000525</td>
</tr>
<tr>
<td>C-85-TXL-47</td>
<td>0.000479</td>
<td>0.001385</td>
<td>0.001175</td>
<td>0.000693</td>
<td>0.000587</td>
</tr>
<tr>
<td>C-85-TXL-51</td>
<td>0.000548</td>
<td>0.001583</td>
<td>0.001385</td>
<td>0.000792</td>
<td>0.000692</td>
</tr>
</tbody>
</table>
Table D.2: The $D_{90}$ is the sample, the $D_{Calc}$ is the calculated value using equation 5.42.

<table>
<thead>
<tr>
<th>Den Helder</th>
<th>$Y_g = 2$</th>
<th>$D_{90}$ [m]</th>
<th>$D_{Calc}$ BH [m]</th>
<th>$D_{Calc}$ HB [m]</th>
<th>$Y_g = 4$</th>
<th>$D_{Calc}$ BH [m]</th>
<th>$D_{Calc}$ HB [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>PB1DH-A3-85-02</td>
<td>0.000348</td>
<td>0.001845</td>
<td>0.000194</td>
<td>0.000923</td>
<td>9.69E-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A3-85-05</td>
<td>0.000345</td>
<td>0.001214</td>
<td>0.000763</td>
<td>0.000607</td>
<td>0.000382</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A3-85-11</td>
<td>0.000354</td>
<td>0.001311</td>
<td>0.000775</td>
<td>0.000656</td>
<td>0.000388</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A3-85-15</td>
<td>0.000319</td>
<td>0.00102</td>
<td>0.00024</td>
<td>0.00051</td>
<td>0.00012</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A3-85-16</td>
<td>0.000495</td>
<td>0.001263</td>
<td>0.000472</td>
<td>0.000631</td>
<td>0.000236</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A3-85-22</td>
<td>0.000398</td>
<td>0.001263</td>
<td>0.000644</td>
<td>0.000631</td>
<td>0.000322</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A3-85-43</td>
<td>0.00044</td>
<td>0.001748</td>
<td>0.001526</td>
<td>0.000874</td>
<td>0.000763</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A3-85-47</td>
<td>0.000439</td>
<td>0.001505</td>
<td>0.001163</td>
<td>0.000753</td>
<td>0.000581</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A3-85-51</td>
<td>0.000415</td>
<td>0.001457</td>
<td>0.001318</td>
<td>0.000728</td>
<td>0.000659</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A3-85-54</td>
<td>0.000332</td>
<td>0.001554</td>
<td>0.001357</td>
<td>0.000777</td>
<td>0.000678</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A3-85-58</td>
<td>0.000269</td>
<td>0.001845</td>
<td>0.001647</td>
<td>0.000923</td>
<td>0.000824</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A3-85-62</td>
<td>0.000428</td>
<td>0.0017</td>
<td>0.001524</td>
<td>0.00085</td>
<td>0.000762</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A3-85-66</td>
<td>0.000467</td>
<td>0.001894</td>
<td>0.001613</td>
<td>0.000947</td>
<td>0.000807</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A3-85-69</td>
<td>0.000433</td>
<td>0.0017</td>
<td>0.001359</td>
<td>0.00085</td>
<td>0.000679</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A3-85-71</td>
<td>0.000422</td>
<td>0.001651</td>
<td>0.001308</td>
<td>0.000826</td>
<td>0.000654</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A3-85-75</td>
<td>0.000493</td>
<td>0.001845</td>
<td>0.001647</td>
<td>0.000923</td>
<td>0.000824</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A1-65-02</td>
<td>0.000497</td>
<td>0.002039</td>
<td>0.001221</td>
<td>0.00102</td>
<td>0.00061</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A1-65-05</td>
<td>0.000494</td>
<td>0.0017</td>
<td>0.001134</td>
<td>0.00085</td>
<td>0.000567</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A1-65-14</td>
<td>0.000469</td>
<td>0.001651</td>
<td>0.001153</td>
<td>0.000826</td>
<td>0.000577</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A1-65-17</td>
<td>0.000536</td>
<td>0.001457</td>
<td>0.000891</td>
<td>0.000728</td>
<td>0.000446</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A1-65-20</td>
<td>0.000476</td>
<td>0.001263</td>
<td>0.000727</td>
<td>0.000631</td>
<td>0.000363</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A1-65-23</td>
<td>0.000556</td>
<td>0.0017</td>
<td>0.001526</td>
<td>0.00085</td>
<td>0.000763</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A1-65-26</td>
<td>0.000504</td>
<td>0.001651</td>
<td>0.001071</td>
<td>0.000826</td>
<td>0.000535</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB1DH-A1-65-29</td>
<td>0.000542</td>
<td>0.001748</td>
<td>0.001112</td>
<td>0.000874</td>
<td>0.000556</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Table D.3: The Reynolds number for every sample, both the Bingham (BH) and the Herschel-Bulkley (HB) fluid model and all scenarios.

<table>
<thead>
<tr>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
<th>Scenario 4</th>
<th>Scenario 5</th>
<th>Scenario 6</th>
<th>Scenario 7</th>
<th>Scenario 8</th>
<th>Scenario 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>HB</td>
<td>BH</td>
<td>HB</td>
<td>BH</td>
<td>HB</td>
<td>BH</td>
<td>HB</td>
<td>BH</td>
</tr>
<tr>
<td>10.6</td>
<td>12.3</td>
<td>9.1</td>
<td>10.7</td>
<td>5.2</td>
<td>6.1</td>
<td>5.7</td>
<td>6.7</td>
<td>4.9</td>
</tr>
<tr>
<td>9.0</td>
<td>13.0</td>
<td>7.8</td>
<td>11.3</td>
<td>4.4</td>
<td>6.5</td>
<td>4.8</td>
<td>7.1</td>
<td>4.2</td>
</tr>
<tr>
<td>9.9</td>
<td>12.0</td>
<td>8.6</td>
<td>10.4</td>
<td>4.9</td>
<td>5.9</td>
<td>5.3</td>
<td>6.5</td>
<td>4.6</td>
</tr>
<tr>
<td>9.8</td>
<td>12.1</td>
<td>8.5</td>
<td>10.5</td>
<td>4.8</td>
<td>6.0</td>
<td>5.3</td>
<td>6.6</td>
<td>4.5</td>
</tr>
<tr>
<td>9.9</td>
<td>12.0</td>
<td>8.6</td>
<td>10.4</td>
<td>4.9</td>
<td>6.0</td>
<td>5.3</td>
<td>6.5</td>
<td>4.6</td>
</tr>
<tr>
<td>12.2</td>
<td>15.5</td>
<td>10.6</td>
<td>13.5</td>
<td>6.0</td>
<td>7.7</td>
<td>6.5</td>
<td>8.5</td>
<td>5.7</td>
</tr>
<tr>
<td>13.0</td>
<td>15.8</td>
<td>11.2</td>
<td>13.7</td>
<td>6.4</td>
<td>7.9</td>
<td>6.9</td>
<td>8.6</td>
<td>6.0</td>
</tr>
<tr>
<td>13.1</td>
<td>16.1</td>
<td>11.3</td>
<td>14.0</td>
<td>6.4</td>
<td>8.0</td>
<td>7.0</td>
<td>8.8</td>
<td>6.1</td>
</tr>
<tr>
<td>12.5</td>
<td>15.7</td>
<td>10.8</td>
<td>13.6</td>
<td>6.1</td>
<td>7.8</td>
<td>6.7</td>
<td>8.5</td>
<td>5.8</td>
</tr>
<tr>
<td>14.2</td>
<td>19.6</td>
<td>12.3</td>
<td>17.0</td>
<td>7.0</td>
<td>9.8</td>
<td>7.6</td>
<td>10.7</td>
<td>6.6</td>
</tr>
<tr>
<td>11.3</td>
<td>14.3</td>
<td>9.8</td>
<td>12.4</td>
<td>5.6</td>
<td>7.1</td>
<td>6.1</td>
<td>7.8</td>
<td>5.3</td>
</tr>
<tr>
<td>14.6</td>
<td>17.0</td>
<td>12.1</td>
<td>14.2</td>
<td>6.8</td>
<td>8.0</td>
<td>7.4</td>
<td>8.8</td>
<td>6.4</td>
</tr>
<tr>
<td>20.9</td>
<td>29.7</td>
<td>18.1</td>
<td>25.8</td>
<td>10.2</td>
<td>14.9</td>
<td>11.1</td>
<td>16.3</td>
<td>9.6</td>
</tr>
<tr>
<td>31.2</td>
<td>40.3</td>
<td>27.0</td>
<td>35.1</td>
<td>15.3</td>
<td>20.3</td>
<td>16.7</td>
<td>22.4</td>
<td>14.5</td>
</tr>
<tr>
<td>31.1</td>
<td>38.4</td>
<td>26.9</td>
<td>33.3</td>
<td>14.3</td>
<td>18.1</td>
<td>15.7</td>
<td>19.9</td>
<td>13.5</td>
</tr>
<tr>
<td>29.8</td>
<td>37.1</td>
<td>25.7</td>
<td>32.2</td>
<td>14.4</td>
<td>18.3</td>
<td>15.8</td>
<td>20.1</td>
<td>13.5</td>
</tr>
<tr>
<td>45.1</td>
<td>61.7</td>
<td>39.0</td>
<td>53.7</td>
<td>21.5</td>
<td>30.2</td>
<td>23.8</td>
<td>33.6</td>
<td>20.3</td>
</tr>
<tr>
<td>70.9</td>
<td>93.4</td>
<td>61.4</td>
<td>81.4</td>
<td>28.9</td>
<td>39.4</td>
<td>31.5</td>
<td>43.5</td>
<td>27.3</td>
</tr>
<tr>
<td>20.2</td>
<td>24.4</td>
<td>17.4</td>
<td>21.2</td>
<td>9.9</td>
<td>12.2</td>
<td>10.8</td>
<td>13.3</td>
<td>9.4</td>
</tr>
<tr>
<td>21.4</td>
<td>25.4</td>
<td>18.5</td>
<td>22.1</td>
<td>10.5</td>
<td>12.6</td>
<td>11.4</td>
<td>13.8</td>
<td>9.9</td>
</tr>
<tr>
<td>22.4</td>
<td>27.3</td>
<td>19.4</td>
<td>23.7</td>
<td>11.0</td>
<td>13.5</td>
<td>12.0</td>
<td>14.7</td>
<td>10.6</td>
</tr>
<tr>
<td>17.8</td>
<td>21.6</td>
<td>15.4</td>
<td>18.7</td>
<td>8.8</td>
<td>10.8</td>
<td>9.6</td>
<td>11.8</td>
<td>8.3</td>
</tr>
<tr>
<td>31.8</td>
<td>59.8</td>
<td>27.6</td>
<td>52.1</td>
<td>15.6</td>
<td>30.1</td>
<td>17.0</td>
<td>33.1</td>
<td>14.7</td>
</tr>
<tr>
<td>37.9</td>
<td>45.6</td>
<td>32.8</td>
<td>39.6</td>
<td>18.5</td>
<td>22.7</td>
<td>20.3</td>
<td>24.9</td>
<td>17.5</td>
</tr>
<tr>
<td>34.9</td>
<td>42.5</td>
<td>30.2</td>
<td>36.9</td>
<td>17.0</td>
<td>21.1</td>
<td>18.6</td>
<td>23.2</td>
<td>16.1</td>
</tr>
<tr>
<td>36.0</td>
<td>42.7</td>
<td>31.1</td>
<td>37.1</td>
<td>17.6</td>
<td>21.3</td>
<td>19.2</td>
<td>23.3</td>
<td>16.6</td>
</tr>
</tbody>
</table>

**Texel - Reynolds Number**
<table>
<thead>
<tr>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
<th>Scenario 4</th>
<th>Scenario 5</th>
<th>Scenario 6</th>
<th>Scenario 7</th>
<th>Scenario 8</th>
<th>Scenario 9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Den Helder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
D.3. Yield Point - Best Fit

The yield points is presented in figure D.1 and figure D.2. From top to bottom in both figures the results is presented of the yield point from the model with the best fit, yield point from the model with the best fit with weighting factor 1 and the yield point from the model with the best fit with weighting factor 2.

D.3.1. Texel

(a) The yield point according to the best fit of the fluid models. No weighting factor.

(b) The yield point according to the best fit of the fluid models. Weighting factor according to shear rate 1.

(c) The yield point according to the best fit of the fluid models. Weighting factor according to shear rate 2.

Figure D.1: Yield point and the best fit to the experimental data for Texel.
D.3.2. Den Helder

(a) The yield point according to the best fit of the fluid models. No weighting factor.

(b) The yield point according to the best fit of the fluid models. Weighting factor according to shear rate 1.

(c) The yield point according to the best fit of the fluid models. Weighting factor according to shear rate 2.

Figure D.2: Yield point and the best fit to the experimental data for Den Helder.
D.4. Lag Diagram
The lag diagram for all scenarios and both sites.

D.4.1. Texel

Figure D.3: The lag diagram for scenario 1 in Texel.
Figure D.4: The lag diagram for scenario 2 in Texel.

Figure D.5: The lag diagram for scenario 3 in Texel.
Figure D.6: The lag diagram for scenario 4 in Texel.

Figure D.7: The lag diagram for scenario 5 in Texel.
Figure D.8: The lag diagram for scenario 6 in Texel.

Figure D.9: The lag diagram for scenario 7 in Texel.
Figure D.10: The lag diagram for scenario 8 in Texel.

Figure D.11: The lag diagram for scenario 9 in Texel.
D.4.2. Den Helder

Figure D.12: The lag diagram for scenario 1 in Den Helder.
Figure D.13: The lag diagram for scenario 2 in Den Helder.

Figure D.14: The lag diagram for scenario 3 in Den Helder.
Figure D.15: The lag diagram for scenario 4 in Den Helder.

Figure D.16: The lag diagram for scenario 5 in Den Helder.
Figure D.17: The lag diagram for scenario 6 in Den Helder.

Figure D.18: The lag diagram for scenario 7 in Den Helder.
Figure D.19: The lag diagram for scenario 8 in Den Helder.

Figure D.20: The lag diagram for scenario 9 in Den Helder.
This appendix contains the full code of the main program written for this thesis. The code is as used, which means that it is far from optimized and parts may be present but no longer in use.

```fortran
! ****************************************************************************
! PROGRAM: SampleSource
!
! PURPOSE: This program is used to gather the data from all different sources and
! analyzing all samples one by one from one location at a time and making calculations with
! regards to the source location and time as well as the processes that have taken place.
!
! REMARKS: Everything marked "TO BE DONE:" is to be included in the future to make the
! program and model more accurate, i.e.:
! * make all date&time calculation correct and completer for all cases, i.e.
! leap years and months of different days (31, 28, 30), those things. Years are totally
! ignored for now.
!
! ****************************************************************************

Program SourceSample

Use SourceModule

IMPLICIT NONE

! Variables
!
! Program Options
Integer :: ScenarioNo, ScenarioNo1, ScenarioNo2

Integer :: FullRunCounter

Character(10) :: Location

Integer :: RunOption
!
! RunOption = 1, results in a fast calculation with minimal output;
! RunOption = 2, results in maximum amount of output, separate files for all samples etc.

Character(3) :: FluidData ! Is No or Yes

Integer :: ProgramOption

Real :: Loss_P

! For reading the input and creation of the output files
Character(*), Parameter :: filefolder_in  =  "D:\hijnekam\Desktop\Program\ProgramInput\\!
    ! Inputfiles
Character(*), Parameter :: filefolder_out =  "D:\hijnekam\Desktop\Program\ProgramOutput\\!
    ! Outputfiles
Character(*), Parameter :: filefolder_out_1 =  "D:\hijnekam\Desktop\Program\ProgramOutput\Option1\\!
    ! Special Cases
Character(*), Parameter :: filefolder_out_extra =  "D:\hijnekam\Desktop\Program\ProgramOutput\ExtraFiles\\!
    ! Outputfiles
!
! Two other locations are used, these are however defined where required (in subroutines).
!
! Backup files are located in "D:\hijnekam\Desktop\Program\ProgramOutput\Scenarios" but this is not specified
!
! The backup files are extensive 2*9 files per sample, 1242 files in total.

Integer :: InputDim=1, InputCSV_PD=2, InputCSV_PD_Fix=3, InputCSV_Sur=4, SampleData=5,
    InputMP=6, SieveResults = ?, SieveResults_V = 8
```
Input files, all are used

Integer :: Output=21, OutputGraph_PD=22, OutputGraph_Sur=23, OutputSamples=24, SampleSourceLog=25, SampleLog=26, OutputLosses=27, OutputSamplesLoc=28 Integer :: OutputFluidProperties=30, OutputFluidPropTex=31 Integer :: Output transport properties=40, OutputTransportPropTex=41

Output files, not all of them are used in every run option.

Integer :: TableHans=99

Character(150) :: filename
Character(150) :: name
Character(150) :: title_graph
Character(150) :: title
Real :: Value, Margin
Integer :: Index_PD, Index_Sur
Integer :: Index_PDMF ! Adapted value for stability reasons, it is the value lower than Index_PD for which a mudflow is given in the ProData file.
Integer :: Index_Sol_PD, Index_Sol_Sur

Standards and constant variables

Integer :: i, j, MainI ! Do loop integers
Real(8), Parameter :: PI = 4.0d0*atan(1.d0) ! Quite precise
Real, Parameter :: g = 9.80665d0

Real, Parameter :: n_BH = 1.0d0
Real, Parameter :: MinYg = 2.0d0
Real, Parameter :: MaxYg = 4.0d0
Real, Parameter :: Rho_Grain = 2650.d0

Variables regarding reading of the input files

Integer :: nlines, nlines_data, Nlines_PD, Nlines_sur, io
Integer, Allocatable :: Bit_DPT(:), Hol_DPT(:), Car_POS(:), Car_SPD(:)
Integer, Allocatable :: Pull(:,), Push(:,), Torq_pos(:)
Integer, Allocatable :: Rot_SPD(:,), M_Press(:,), MudFlw(:,), Torq_neg(:)
Integer, Allocatable :: Press_1(:,), Press_2(:,), Press_3(:,), HS_D(:)
Character(8), Allocatable :: Date(:,), Time(:,)
Integer, Allocatable :: Joint(:,), Btot(:,)
Real, Allocatable :: CL(:,), InclinationRaw(:,), AzimuthRaw(:,)
Real, Allocatable :: HS(:,), Dip(:,), ElevationCalc(:,), ElevationMGS(:,)
Real, Allocatable :: R_Calc(:,), R_MGS(:,)
Real(8), Allocatable :: Away(:,), MD(:,)

Variables

Character(19), Allocatable :: DateTime_PD(:)
Character(19), Allocatable :: DateTime_Sur(:)
Character(19), Allocatable :: DateTime_Samples(:)
Character(19), Allocatable :: DateTimeSample
Integer :: TimeInt
Integer(2) :: Day, Month, Hours, Minutes, Seconds
Integer(4) :: Year
Integer, Allocatable :: TimeInt_PD(:,), TimeInt_Sur(:,)
Real, Allocatable :: PumpVol(:,)
Real, Allocatable :: TranspLen(:,)
Real(8), Allocatable :: TranspLenSum(:)
Real :: D_bh, D_ds
Real :: D_drillbit
Real :: Thickness_ds
Real :: Length_Pipe, Length_BHA
Real :: Planned_Azimuth
Real :: Conversion_Degr_Pa, Conversion_RPM_PerS
Real :: Conversion_Degr_Lb, Conversion_lb_Pa
Real :: A_bh, A_ds, A_annulus
Real, Allocatable :: Away_PDInt(:,)
Real :: SourceLoc_PD, SourceLoc_Sur, SourceLocAV
Real(8), Allocatable :: SourceLocAVAll(:)
Real :: TimeInS
Integer :: HH, MM, SS
Integer :: TimeIntAv
Integer, Allocatable :: TimeIntAvAll(:)
Integer :: TimeInt1, TimeInt2
Integer, Allocatable :: PumpTimesAll(:)
Integer :: PumpTime
Character(len=:), Allocatable :: Sample_ID(:, ! Length and number is variable depending on the location.
Character(len=:), Allocatable :: Sample_Code ! Length is variable depending on the location.
Integer :: No_Samples
Real(8), Allocatable :: PositionBitAll(:)
Real, Allocatable :: Temp(:, EC(:, pH(:, MF(:, Rho_Bf(:, SC(:,
Integer, Allocatable :: RPM_600(:, RPM_300(:, RPM_200(:, RPM_100(:,
Integer, Allocatable :: RPM_6(:, RPM_30(:, RPM_6(:, RPM_3(:, Gel10s(:, Gel10min(:,
Real(8), Dimension(8)) :: GammaY
Real(8), Dimension(8)) :: LogGamma ! Log(gamma)
Real(8), Dimension(20) :: Tau
Real(8), Dimension(20) :: Tau_HB
Real(8), Dimension(20) :: Tau_BH
Real(8), Dimension(8)) :: Y ! Log(Tau−Tau_yield)
Real :: Tau_Yield_HB
Real :: K, n
Real, Allocatable :: Tau_Yield_HB_Tot(:)
Real, Allocatable :: K_Tot(:, n_Tot(:, ! Best fit of all samples
Real :: Alpha, Delta
Real, Allocatable :: Alpha_Tot(:, Delta_HB(:, Delta_BH(:,
Real(8), Dimension(20)) :: Delta_ALL, Alpha_ALL
Real :: PV, YP
Real, Allocatable :: PV_Tot(:, YP_Tot(:)
Character(3), Allocatable :: Activity(:)
Integer, Allocatable :: Joint_MP(:, P_Mx1(:, P_Mx2(:, P_Min(:, P_Cal(:, P_Real(:,
Real, Allocatable :: Length_MP(:,
Integer :: Nlines_MP
Integer :: JointNo
Integer(2) :: Day_MP, Month_MP
Character(3) :: ActivityKey
Real, Allocatable :: Mud_P_Tot(:)
Real :: MaxPlugR
Real :: Tau_Y
Integer :: Mud_P
Real, Allocatable :: MaxPlugR_Tot(:)
Integer :: HB_No, BH_No
! Counts the number of samples that the fluid model is the best fit.
Integer :: NoFluid_No
! Counts the number of times the fluid data was missing or incomplete.
Integer :: DeltaWBH1_No, DeltaWBH2_No
Integer :: DeltaWBB1_No, DeltaWBB2_No
Real :: SourceDepth, SourceAway, SourceRight
Real, Allocatable :: SourceDepthTot(:), SourceAwayTot(:), SourceRightTot(:)
Real :: SamplingDepth, SamplingAway, SamplingRight
Real, Allocatable :: SamplingDepthTot(:), SamplingAwayTot(:), SamplingRightTot(:)
Character(1000) :: Message
Integer :: EndOption

! Sieve Related
Real, Allocatable :: D(:,,:), D_V(:,:), Sieve_Code(:)
Character(14), Allocatable :: Sieve_Code_V(:)
Real :: D90, D15
Real :: Rho_Inflow, Rho_Grain
Real :: Loss_BH, Loss_HB
Real :: LossPercentage_BH, LossPercentage_HB
Real, Allocatable :: InfiltrationLossBTL_BH_Tot(:), InfiltrationLossBTL_HB_Tot(:)
Real, Allocatable :: InfilLossPerc_BH_Tot(:), InfilLossPerc_HB_Tot(:)

! Max diameter
Real :: D_Max
Real, Allocatable :: D_MaxTot(:)
Real :: Max
Real, Allocatable :: VArr(:)
Integer :: VArrCountN0
Real :: AverageV
Real, Allocatable :: AverageV_Tot(:)
Real :: MaxV
Real, Allocatable :: MaxV_Tot(:)
Real :: Temp1
Real :: VolSol
Real :: Max_d_BH, c1, v_settle_BH
Real, Allocatable :: Max_d_BH_Tot(:), v_settle_BH_Tot(:)
Real :: Gamma_Ann1, Gamma_Ann2, Eff_Visc_HB, Re_HB2
Real, Allocatable :: Gamma_AnnTot1(:,), Gamma_AnnTot2(:,), Eff_Visc_HB_Tot(:,), Re_HB2_Tot(:)
Real :: DeltaWeightBH1, DeltaWeightBH2
Real :: DeltaWeightBH1, DeltaWeightBH2
Real :: Re_Gen_BH, Re_Gen_HB
Real, Allocatable :: Re_Gen_BH_Tot(:,), Re_Gen_HB_Tot(:)
Real :: DmaxTrMinBH, DmaxTrMinHB, DmaxTrMaxBH, DmaxTrMaxHB
Real, Allocatable :: DmaxTrMinBH_Tot(:,), DmaxTrMinHB_Tot(:,), DmaxTrMaxBH_Tot(:,),
DmaxTrMaxHB_Tot(:)

! START PROGRAM
!----------------------------------------
ScenarioNo = 1
FullRunCounter = 0

100 Continue
Write(*,'(A)') "Give a ProgramOption, or press '0' for more information:"
Read(*,'(I)') ProgramOption
Write(*,'(A)') "-----------------------------------------------------------------------------------"
Select Case(ProgramOption)
Case(0)
   Write(*,*)
Write(*,'(A)')  "ProgramOption 1 executes once for either Texel or Den Helder, loss and
diameter are screen input."
Write(*,'(A)')  "ProgramOption 2 runs 9 different scenarios for one location."
Write(*,'(A)')  "ProgramOption 3 executes 2 scenarios from screen input. 1 scenario is
possible by putting '0' as second scenario."
Write(*,')')
Write(*,'(A)')  "ProgramOption 4 runs all scenarios for both locations."
Write(*,')')
Write(*,')')
Go to 100

Case(1)

Write(*,'(A)')  "Give the location, either 'Texel' or 'Den Helder', most abbreviations
are recognized:
Read(*,'(A)')  Location
Write(*,')')

300 Continue  ! Redo, incorrect input
Write(*,'(A)')  "Give a loss percentage, in the following format '0.00':"
Read(*,'(F4.2)')  Loss_P
Write(*,'(A)')  "Give a diameter for the borehole, in [m] and the format '0.00':"
Read(*,'(F4.2)')  D_bh
Write(*,'(A,F4.2)')  "Loss_P : * ,Loss_P
Write(*,'(A,F4.2)')  "D_bh : * ,D_bh
If ( Loss_P >= 0.75d0 . or . Loss_P < 0.0d0 . or . D_bh < 0.0d0 . or . D_bh > 1.0d0 ) Then
Write(*,')')  "The given input is unrealistic or not tested, please try again."
Go To 300
End If

200 Continue  ! Redo, incorrect run option

Write(*,'(A)')  "Give program execute option, '1' = fast, '2' = full:"  
Read(*,'(I)')  RunOption

Select Case(RunOption)

Case(1)
Write(Message ,'(A,12.2)')  "The location is " / trim(Location) / " and the fast option
with minimal file output is chosen."

Case(2)
Write(Message ,'(A,12.2)')  "The location is " / trim(Location) / " and the full option
with maximum no. of output files is chosen."

Case Default
Write(*,')')  "Unknown RunOption, please try again."
Go To 200
End Select

Call WriteLog(2,trim(Message),SampleSourceLog)
Write(*,')')

Case(2)

Write(*,'(A)')  "Give the location, either 'Texel' or 'Den Helder', most abbreviations
are recognized:"  
Read(*,'(A)')  Location
Write(*,')')
RunOption = 1

Case (3)

Write(*, '(A)') "If only 1 scenario is preferred, put '0' for the second scenario."
Write(*, '(A)') "Give the first scenario:"
Read(*, '(I)') ScenarioNo1
Write(*, '(A)') "Give the second scenario:"
Read(*, '(I)') ScenarioNo2

ScenarioNo = ScenarioNo1

Write(*, '(A)') "Give the location, either 'Texel' or 'Den Helder', most abbreviations are recognized:"
Read(*, '(A)') Location
Write(*, '(A)') "Give the first location, either 'Texel' or 'Den Helder', most abbreviations are recognized:"

RunOption = 1

Case (4)

Write(*, '(A)') "Give the first location, either 'Texel' or 'Den Helder', most abbreviations are recognized:"
Read(*, '(A)') Location
Write(*, '(A)') "Give the first location, either 'Texel' or 'Den Helder', most abbreviations are recognized:"

ScenarioNo = 1
RunOption = 1

End Select

500 Continue ! Start Scenario Loop

Select Case(Location)
! Multiple abbreviations can be used
Case ('Den Helder', 'den helder', 'DH', 'dh', 'Dh')

Location = 'Den Helder'
! This is the name that is also put in the graphs and file names.
Open(SampleData, file = filefolder_in//"SampleData_DH.csv")
Open(InputDim, file = filefolder_in//"InputDim_DH.INP")
Open(InputCSV_PD, file = filefolder_in//"InputData_PD_DH.csv") ! The file as downloaded from ProData but titles (except for first line) are removed as well as the quotation marks around each term.
Open(InputCSV_Sur, file = filefolder_in//"InputData_Sur_DH_Total.csv") ! All survey data, from DH, combined into one file.
Open(InputMP, file = filefolder_in//"Input_MP_DH.csv") ! Contains the annular mudpressures per joint no.
Case ('Texel', 'texel', 'TXL', 'txl', 'T', 't')

Location = 'Texel'
Open(SampleData, file = filefolder_in//"SampleData_Texel.csv")
Open(InputDim, file = filefolder_in//"InputDim_Texel.INP")
Open(InputCSV_PD, file = filefolder_in//"InputData_PD_Texel_V2.csv") ! The values of this file are calibrated afterwards and rounded, the error this causes is supposed to be of limited influence considering the other uncertainties.
Open(InputCSV_Sur, file = filefolder_in//"InputData_Sur_Texel_Total.csv") ! All
survey data, from Texel, combined into one file.

Open(InputMP.file = filefolder_in/"Input_MP_T.csv") ! Contains the annular mudpressures per joint no.

Case Default
Write(*,*)
Write(*, '(A)') "The given location is not recognized, please start again."
Write(*,*)
Write(*, '(A)')

Go to 100

End Select

Open(SieveResults, file = filefolder_in/"SieveResults.csv") ! This file contains the data from both locations.

!-----------------------------
! Reading files
!-----------------------------

Call ReadingFileCSV_PD(InputCSV_PD, DateTime_PD, Bit_DPT, Hol_DPT, Car_POS, Car_SPD, Pull, Push, Torq_pos, Rot_SPD, M_Press, MudFlw, Torq_neg, Press_1, Press_2, Press_3, HS_D, Nlines_PD)

If (RunOption == 2) Write(*, '(A)') "Check A1"
If (RunOption = 2, full run, all check messages are shown as well.
Call ReadingFileCSV_Sur(InputCSV_Sur, Nlines_sur, DateTime_Sur, Joint, CL, MD, InclinationRaw, AzimuthRaw, Biot, Dip, ElevationCalc, ElevationMGS, Away, R_Calc, R_MGS)

If (RunOption == 2) Write(*, '(A)') "Check A2"

Call ReadingFileSampleData(Location, SampleData, No_Samples, DateTime_Samples, Sample_ID, Temp, EC, pH, MF, Rho_Bf, SC, RPM_600, RPM_300, RPM_200, RPM_100, RPM_60, RPM_30, RPM_6, RPM_3, Gel10s, Gel10min)

If (RunOption == 2) Write(*, '(A)') "Check A3"

If (trim(Location) == 'Den Helder')
Call ReadingFileMP(InputMP, Location, Nlines_MP, Activity, Joint_MP, Length_MP, P_Max1, P_Max2, P_Calc, P_Real)
Else If (trim(Location) == 'Texel')
Call ReadingFileMP(InputMP, Location, Nlines_MP, Activity, Joint_MP, Length_MP, P_Real, P_Min)
End if

If (RunOption == 2) Write(*, '(A)') "Check A4"

Call ReadingFileSieve(SieveResults, Sieve_Code, D)

If (RunOption == 2) Write(*, '(A)') "Check A5"

! Call ReadingFileSieve(SieveResults_V, Sieve_Code_V, D_V)

If (RunOption == 2) Write(*, '(A)') "Check A6"

!-----------------------------
! Dimensions & Conversions to SI
!-----------------------------

Read(InputDim,*) ! First line contains the location, location is already given by screen input in order to know which files are to be read, so this is redundant.

Select Case(Location)

Case('Den Helder')

Read(InputDim,*) D_drillbit ! [mm]
D_drillbit = D_drillbit/1000 ! [mm] to [m]

Read(InputDim,*) D_ds ! [mm]
D_ds = D_ds/1000 ! [mm] to [m]
Read ( InputDim , * ) Thickness ds  ! [nm]
Thickness ds = Thickness ds / 1000  ! [nm] to [m]

Read ( InputDim , * ) Length Pipe  ! [m]
Read ( InputDim , * ) Length_BHA  ! [m]

Read ( InputDim , * ) Planned_Azimuth  ! [ø]
Read ( InputDim , * ) Rho_Inflow  ! [–]
Rho_Inflow = Rho_Inflow * 1000  ! [kg/m³]

Case ( 'Texitel' )

Read ( InputDim , * ) D_drillbit  ! [mm]
D_drillbit = D_drillbit / 1000  ! [mm] to [m]

Read ( InputDim , * ) D_ds  ! [mm]
D_ds = D_ds / 1000  ! [mm] to [m]

Read ( InputDim , * ) Thickness ds  ! [mm]
Thickness ds = Thickness ds / 1000  ! [mm] to [m]

Read ( InputDim , * ) Length Pipe  ! [m]
Read ( InputDim , * ) Length_BHA  ! [m]

Read ( InputDim , * ) Planned_Azimuth  ! [ø]
Read ( InputDim , * ) Rho_Inflow  ! [–]
Rho_Inflow = Rho_Inflow * 1000  ! [kg/m³]

Read ( InputDim , * ) Conversion Degr Pa  ! [–]
Read ( InputDim , * ) Conversion Degr lb
Read ( InputDim , * ) Conversion lb Pa
Read ( InputDim , * ) Conversion RPM PerS  ! [–]

Case 'Default'

Write ( *, ' (A)' ) "Location is unfamiliar and thus the input file can't be read properly."
Read ( * , * )
Stop

End Select

If ( RunOption == 2 ) Write ( *, ' (A)' ) "Check A7"

If ( ( ProgramOption == 2 ) . or . ( ProgramOption == 3 ) . or . ( ProgramOption == 4 ) ) Then

Select Case ( ScenarioNo )

Case ( 1 ) ! Smallest diameter, no loss
Loss_P = 0.d0
D_bh = 1.1d0 * D_drillbit

Write ( * , * )
Write ( *, ' (A)' ) "-------------

Case ( 2 ) ! Smallest diameter, test loss
Loss_P = 0.07d0
D_bh = 1.1d0 * D_drillbit

Write ( * , * )
Write ( *, ' (A)' ) "-------------

Case ( 3 ) ! Smallest diameter, 30% loss
Loss_P = 0.30d0

Write ( * , * )
Write ( *, ' (A)' ) "-------------
\( D_{bh} = 1.1 d0 \ast D_{drillbit} \)

Write (*,*)
Write (*,'(A)') "

Case (4) ! Medium diameter, no loss
Loss_P = 0.d0
If (Location == 'Texel') \( D_{bh} = 0.40 d0 \)
If (Location == 'Den Helder') \( D_{bh} = 0.45 d0 \)
Write (*,*)
Write (*,'(A)') "

Case (5) ! Medium diameter, test loss
Loss_P = 0.07d0
If (Location == 'Texel') \( D_{bh} = 0.40 d0 \)
If (Location == 'Den Helder') \( D_{bh} = 0.45 d0 \)
Write (*,*)
Write (*,'(A)') "

Case (6) ! Medium diameter, 30% loss
Loss_P = 0.30d0
If (Location == 'Texel') \( D_{bh} = 0.40 d0 \)
If (Location == 'Den Helder') \( D_{bh} = 0.45 d0 \)
Write (*,*)
Write (*,'(A)') "

Case (7) ! Large diameter, no loss
Loss_P = 0.d0
If (Location == 'Texel') \( D_{bh} = 0.55 d0 \)
If (Location == 'Den Helder') \( D_{bh} = 0.80 d0 \)
Write (*,*)
Write (*,'(A)') "

Case (8) ! Large diameter, test loss
Loss_P = 0.07d0
If (Location == 'Texel') \( D_{bh} = 0.55 d0 \)
If (Location == 'Den Helder') \( D_{bh} = 0.80 d0 \)
Write (*,*)
Write (*,'(A)') "

Case (9) ! Large diameter, 30% loss
Loss_P = 0.30d0
If (Location == 'Texel') \( D_{bh} = 0.55 d0 \)
If (Location == 'Den Helder') \( D_{bh} = 0.80 d0 \)
Write (*,*)
Write (*,'(A)') "

Case Default
Write (*,*)
Write (*,'(A)') "Something went wrong."
Write (*,*)
Write (*,'(A)') "

Read(*,*)
Stop
End Select
Write (Name, '(A, 12.2, A)') Trim (Short (Location)) // "_OutputSamples_Scenario_", ScenarioNo, ".csv"
Open (OutputSamples, file = filefolder_out // trim (Name))
Write (OutputSamples, '(A)') ' Date Time ; Sample Code ; Main Sample ; Source Loc Av [m] ; Sampling Depth ; Sampling Away Tot ; Sampling Right Tot ; Source Depth Tot ; Source Right Tot ; Position Bit All [m] ; Pump Time [s] ; HB Tau Yield ; HB K ; HB n [−] ; Alpha ; Delta HB ; PV [Pa . s] ; YP [Pa] ; Delta Bingham ; Mud Pressure Annulus [Pa] ; Max Plug Flow Radius [m] ; Transport Parameter BH ; Transport Parameter HB ; Average velocity [m/s] ; Max Diameter Suspension ; ReBH ; ReHB ; MuHB ; HeBH ; Max_dBH ; v_settle_BH'
Write (Name, '(A, 12.2, A)') Trim (Short (Location)) // "_OutputSamples Loc_Scenario_", ScenarioNo, ".csv"
Open (OutputSamplesLoc, file = filefolder_out // trim (Name))
Write (OutputSamplesLoc, '(A)') ' Sample Code ; Date Time Sample ; Sampling Away ; Sampling Right ; Source Away ; Source Right '
Write (Name, '(A, 12.2, A)') Trim (Short (Location)) // "_FluidProperties_Scenario_", ScenarioNo, ".csv"
Open (OutputFluidProperties, file = filefolder_out // trim (Name))
Write (OutputFluidProperties, '(A)') ' Sample Code ; Date Time Sample ; HB Tau Yield ; HB K ; HB n [−] ; Alpha ; Delta HB ; PV [Pa . s] ; YP [Pa] ; Delta Bingham '
Write (Name, '(A, 12.2, A)') Trim (Short (Location)) // "_FluidPropTex_Scenario_", ScenarioNo, ".txt"
Open (OutputFluidPropTex, file = filefolder_out // trim (Name))
Write (OutputFluidPropTex, '(A)') ' Sample Code ; Date Time Sample ; HB Tau Yield ; HB K ; HB n [−] ; Alpha ; Delta HB ; PV [Pa . s] ; YP [Pa] ; Delta Bingham '
Write (Name, '(A, 12.2, A)') Trim (Short (Location)) // "_OutputLosses_Scenario_", ScenarioNo, ".csv"
Open (OutputLosses, file = filefolder_out // trim (Name))
Write (OutputLosses, '(A)') ' Sample Code ; Loss_P [%] ; Infil Loss Perc_BH [%] ; Infil Loss Perc_BH [%]'
Write (Name, '(A, 12.2, A)') Trim (Short (Location)) // "_TransportProperties_Scenario_", ScenarioNo, ".csv"
Open (OutputTransportProperties, file = filefolder_out // trim (Name))
Write (OutputTransportProperties, '(A)') ' Sample Code ; Date Time Sample ; MaxPlugR ; Average V ; D_Max ; Shear Rate Downhole1 [1/s] ; Shear Rate Downhole2 [1/s] ; Re_Gen_BH ; Re_Gen_HB ; D90 Sample ; Dmax BH Min ; Dmax BH Max ; Dmax HB Min ; Dmax HB Max ; Delta Weight BH1 ; Delta Weight BH2 ; Delta Weight HB1 ; Delta Weight HB2 '
Write (Name, '(A, 12.2, A)') Trim (Short (Location)) // "_TransportPropTex_Scenario_", ScenarioNo, ".txt"
Open (OutputTransportPropTex, file = filefolder_out // trim (Name))
Write (OutputTransportPropTex, '(A)') ' Sample Code ; Date Time Sample ; MaxPlugR ; Average V ; D_Max ; Shear Rate Downhole1 [1/s] ; Shear Rate Downhole2 [1/s] ; Eff Visc HB ; Re_Gen_BH ; Re_Gen_HB ; D90 Sample ; Dmax BH Min ; Dmax BH Max ; Max_dBH ; Delta Weight BH1 ; Delta Weight BH2 ; Delta Weight HB1 ; Delta Weight HB2 '
Write (Name, '(A, 12.2, A)') Trim (Short (Location)) // "_SampleSourceLog_Scenario_", ScenarioNo, ".csv"
Open (SampleSourceLog, file = filefolder_out // trim (Name))
Else
Write (Name, '(A, 12.2, A)') Trim (Short (Location)) // "_OutputSamples_.Dbh", "-_Loss_P_.csv" ! Includes the results of the main do loop, so results of all calculation and of all samples ...
Open (OutputSamples, file = filefolder_out // trim (Name))
Write (OutputSamples, '(A)') ' Date Time ; Sample Code ; Main ; Source Loc Av [m] ; Sampling Depth ; Sampling Away Tot ; Sampling Right Tot ; Source Depth Tot ; Source Right Tot ; Position Bit All [m] ; Pump Time [s] ; HB Tau Yield ; HB K ; HB n [−] ; Alpha ; Delta HB ; PV [Pa . s] ; YP [Pa] ; Delta Bingham ; Mud Pressure Annulus [Pa] ;
Maximum Plug Flow Radius [m] ; Transport Parameter BH ; Transport Parameter HB ; Average velocity [m/s] ; Max Diameter Suspension ; Re_BH ; Re_HB ; Mu_BH ; He_BH ; Max_d_BH ; v_settle_BH

Write (Name, ' (A, F4.2, A)') Trim(Short(Location)) // "_OutputSamplesLoc", D_Bh, "−", Loss_P, ".txt"
Open(OutputSamplesLoc, file = filefolder_out_1 // trim(Name))
Write(OutputSamplesLoc, '(A)') "Sample_Code, DateTimeSample, SamplingAway, SamplingDepth, SamplingRight, SourceDepth, SourceAway, SourceRight"

Write (Name, ' (A, F4.2, A)') Trim(Short(Location)) // "_OutputLosses", D_Bh, "−", Loss_P, ".csv"
Open(OutputLosses, file = filefolder_out_1 // trim(Name))
Write(OutputLosses, '(A)') "Sample Code; Loss_P [%]; InfilLossPerc_BH [%]; InfilLossPerc_HB [%]"

Write (Name, ' (A, F4.2, A)') Trim(Short(Location)) // "_SampleSourceLog", D_Bh, "−", Loss_P, ".txt"
Open(SampleSourceLog, file = filefolder_out_1 // trim(Name))

If (RunOption == 2) Open(TableHans, file = filefolder_out // trim(Name)) // "TableHans_" // trim(Short(Location)) // ".csv"

If (RunOption == 2) Write(TableHans, '(A)') "Sample_Code; Sieve_Code; Sample_Time_Away; Source_Away; Sample_Time_Length; Source_Length; Sample_Time_Depth; Source_Depth"

End If

! Have to be opened anyway, since it contains the data from both locations
Open(SieveResults_V, file = filefolder_in // "SieveResults_V.csv")

!-------------------------------
! Preparations Main Loop
!-------------------------------
! Declaring the LogGamma values to be used in determining the HB parameters
GammaY(1) = 3.0 * Conversion_RPM_PerS
GammaY(2) = 6.0 * Conversion_RPM_PerS
GammaY(3) = 30.0 * Conversion_RPM_PerS
GammaY(4) = 60.0 * Conversion_RPM_PerS
GammaY(5) = 100.0 * Conversion_RPM_PerS
GammaY(6) = 200.0 * Conversion_RPM_PerS
GammaY(7) = 300.0 * Conversion_RPM_PerS
GammaY(8) = 600.0 * Conversion_RPM_PerS

Do i = 1, size(GammaY)
LogGamma(i) = Log10(GammaY(i))
End Do

Call GeometryBoreH(P1, D_bh, D_ds, A_bh, A_ds, A_annulus)

HB_No = 0.0 ! Initial values for counters.
BH_No = 0.0
DeltaWBH1_No = 0.0
DeltaWBH2_No = 0.0
DeltaWHB1_No = 0.0
DeltaWHB2_No = 0.0
NoFluid_No = 0.0

Allocate(PumpTimesAll(No_Samples)) ! Enables a visual check
Allocate(TimeIntAvAll(No_Samples)) ! ""
Allocate(SourceLocAvAll(No_Samples)) ! Collects the value from every run of the loop.
Allocate(PositionBitAll(No_Samples)) ! Required for the lag diagram.
Allocate(Tau_Yield_HB_Tot(No_Samples))
Allocate(K_Tot(No_Samples))
Allocate(n_Tot(No_Samples))
Allocate(PV_Tot(No_Samples))
Allocate(YP_Tot(No_Samples))
Allocate(Alpha_Tot(No_Samples))
Allocate(Delta_Tot_HB(No_Samples))
Allocate (Delta_Tot_BH (No_Samples))
Allocate (Mud_P_Tot (No_Samples))
Allocate (MaxPlugR_Tot (No_Samples))
Allocate (SamplingDepthTot (No_Samples))
Allocate (SamplingAwayTot (No_Samples))
Allocate (SamplingRightTot (No_Samples))
Allocate (SourceDepthTot (No_Samples))
Allocate (SourceAwayTot (No_Samples))
Allocate (SourceRightTot (No_Samples))
Allocate (InfiltrationLossBTL_BH_Tot (No_Samples))
Allocate (InfiltrationLossBTL_HB_Tot (No_Samples))
Allocate (InfilLossPerc_BH_Tot (No_Samples))
Allocate (InfilLossPerc_HB_Tot (No_Samples))
Allocate (D_Max_Tot (No_Samples))
Allocate (AverageV_Tot (No_Samples))
Allocate (MaxV_Tot (No_Samples))
Allocate (Max_D_BH_Tot (No_Samples))
Allocate (v_settle_BH_Tot (No_Samples))
Allocate (Gamma_Ann_Tot1 (No_Samples))
Allocate (Gamma_Ann_Tot2 (No_Samples))
Allocate (Re_Gen_BH_Tot (No_Samples))
Allocate (Re_Gen_HB_Tot (No_Samples))
Allocate (DmaxTrMinBH_Tot (No_Samples))
Allocate (DmaxTrMinHB_Tot (No_Samples))
Allocate (DmaxTrMaxBH_Tot (No_Samples))
Allocate (DmaxTrMaxHB_Tot (No_Samples))

If (RunOption == 2) Write (*, ' (A) ' ) "Check 6"

! START MAIN LOOP
---
Mainloop: Do MainI = 1, No_Samples
Write (*, *)
Write (Message, ' (A,12.2,A,12.2,A,12.2,A,A) ' ) trim (Location) //" Scenario ",ScenarioNo ", " , MainI ", " , No_Samples, " , Sample No. ", Sample_ID (MainI) ! Shows progress of the program through the sample list.
Call WriteLog (1 , trim (Message))
Write (*, ' (A) ' ) " --- START MAIN LOOP ---"
Open (SampleLog , file = filefolder_out//" Log_ "//trim (Location) //" _"//trim (Sample_ID (MainI)) 
//".txt")
! Write (Message , ' (A,A,A) ' ) trim (Location) , " ", trim (Sample_ID (MainI))
! Call WriteLog (3 , Message , SampleLog)
DateTimeSample = DateTime_Samples (MainI)
If (RunOption == 2) Write (*, ' (A) ' ) "Check 6.1"
Sample_Code = Sample_ID (MainI)
Call FindDateTimeValueApprox (DateTimeSample, DateTime_PD, Index_PD)
Call FindApproxDateTime (DateTimeSample, DateTime_Sur, Index_Sur)
! Finds the sample date and time in the ProData file.
If (RunOption == 2) Write (*, ' (A) ' ) "Check 6.2"
Call SamplingLocation (Index_Sur, ElevationCalc, Away, R_Calc, SamplingDepth, SamplingAway, SamplingRight)
! Filling the vector containing the results of all the samples.
SamplingDepthTot(MainI) = SamplingDepth
SamplingAwayTot(MainI) = SamplingAway
SamplingRightTot(MainI) = SamplingRight

! Calculations - Main
!-----------------------------------------------
! Fluid Parameters
!-----------------------------------------------
If (RPM_600(MainI) <= 0 . or. RPM_300(MainI) <= 0 . or. RPM_200(MainI) <= 0 . or. RPM_100(MainI) <= 0 . or. RPM_60(MainI) <= 0 . or. RPM_30(MainI) <= 0 . or. RPM_6(MainI) <= 0 . or. RPM_3(MainI) <= 0) Then

FluidData = 'No' ! Definition of the FluidData case.
Write(Message, '(A)') "Fluid data is missing or incomplete, fluid properties couldn’t be determined."
Call WriteLog(1, trim(Message))

NoFluid_No = NoFluid_No + 1 ! Counter that keeps track of the amount of samples with, missing or incompleter data.

! Necessary or program start assigning its own values to the variables.
Tau_Yield_HB_Tot(MainI) = 0.d0
K_Tot(MainI) = 0.d0
n_Tot(MainI) = 0.d0
Alpha_Tot(MainI) = 0.d0
Delta_Tot_HB(MainI) = 0.d0
Delta_Tot_BH(MainI) = 0.d0
PV_Tot(MainI) = 0.d0
YP_Tot(MainI) = 0.d0

Else

FluidData = 'Yes' ! Definition of the FluidData case.
Write(Message, '(A)') "Fluid data is present and complete."
Call WriteLog(1, trim(Message))

! HB
!----------------
Call Herschel_Bulkley(LogGamma, Y, Tau_Yield_HB, K, n, Alpha, Delta, Sample_Code, MainI, Conversion_Degr_Pa, RPM_600, RPM_300, RPM_200, RPM_100, RPM_60, RPM_30, RPM_6, RPM_3, GammaY, Tau, Tau_Calc_HB, Alpha_All, Delta_All)

K_Tot(MainI) = K ! Filling the vector containing the results of all the samples.
n_Tot(MainI) = n
Tau_Yield_HB_Tot(MainI) = Tau_Yield_HB
Alpha_Tot(MainI) = Alpha
Delta_Tot_HB(MainI) = Delta

Call WriteLog(1, trim(Message))

! Bingham
!----------------
Call Bingham(MainI, PV, YP, RPM_600, RPM_300, Conversion_Degr_lb, Conversion_lb_Pa, Conversion_RPM_PerS, GammaY, Tau_Calc_BH, Tau, Delta)

PV_Tot(MainI) = PV ! Filling the vector containing the results of all the samples.
YP_Tot(MainI) = YP
Delta_Tot_BH(MainI) = Delta

Write(Message, '(A,F,A,F5.2,A,F8.3)') "PV, YP and Delta: " ,PV,"",",",YP,"and",Delta
Call WriteLog(1, trim(Message))
Write(*,*)
If (Delta_Tot_HB(MainI) < Delta_Tot_BH(MainI)) Then
  Write(Message, '(A)') "Fluid parameters determined and Herschel Bulkley results in the best fit."
  Call WriteLog(1, trim(Message))
  HB_No = HB_No + 1 ! Counter that keeps track of the amount of samples where Herschel–Bulkley is the best fit.
Else
  Write(Message, '(A)') "Fluid parameters determined and Bingham results in the best fit."
  Call WriteLog(1, trim(Message))
  BH_No = BH_No + 1 ! Counter that keeps track of the amount of samples where Bingham is the best fit.
End If
End If

If (RunOption == 2) Write(*,*) "Check"

Call TimeInterval_PD(TimeInt_PD, DateTime_PD, Index_PD)
! ProData time interval from the Index_PD to the start of the drilling.
! The subroutine takes the stops into account and the larger intervals due to missing data.

Call TransportLength_PD(PumpVol, Index_PD, A_bh, TranspLen, TranspLenSum, TimeInt_PD, MudFlw, Loss_PD)
If (RunOption == 2) Write(*,') (A)') "Check 6.3"

Call Intersection(DateTime_PD, Index_PD, DateTime_Sur, Index_Sur, TranspLenSum, MD, Index_Sol_PD, Index_Sol_Sur, SourceLoc_Sur, SourceLoc_PD, SourceLocAv, DateTimeSample)
! Works and delivers the indices of the PD and Sur databases that result in Away and TransLenSum having exact or approximately the same value.

SourceLocAvAll(MainI) = SourceLocAv ! Filling the vector containing the results of all the samples.
PositionBitAll(MainI) = MD(Index_Sur)

Call SourceLocation(Index_Sol_Sur, SourceLocAv, MD, ElevationCalc, Away, R_Calc, SourceDepth, SourceAway, SourceRight)
SourceDepthTot(MainI) = SourceDepth
SourceAwayTot(MainI) = SourceAway
SourceRightTot(MainI) = SourceRight
If (RunOption == 2) Write(*,') (A)') "Check 6.4"

If (Index_Sol_PD > 0 .and. Index_Sol_Sur > 0) Then
  Call TransportTimes(DateTime_PD, DateTime_Sur, Index_Sol_PD, Index_Sol_Sur, DateTimeSample, TimeInt1, TimeInt2, TimeIntAv)
  TimeIntAvAll(MainI) = TimeIntAv
If (RunOption == 2) Write(*,') (A)') "Check 6.5"

Call PumpTimes(MudFlw, TimeInt_PD, Index_Sol_PD, Index_Sol, PumpTime)
PumpTimesAll(MainI) = PumpTime
If (RunOption == 2) Write(*,') (A)') "Check 7"
Else
  Write(Message, '(A)') "No solution was found, so both transport and pump time can’t be calculated."
  Call WriteLog(1, trim(Message))
  TimeIntAvAll(MainI) = 0.d0
  PumpTimesAll(MainI) = 0.d0
End If

!-------------------------------------------------------------
Select Case (FluidData)

Case ('Yes')
  Write (Message, '(A)')  "Fluid data is found and complete."
  Call WriteLog (1, trim (Message))

  ! Write(*, '(A,F)')  "A_bh  _  _A_bh
  ! Write(*, '(A,F)')  "Loss_P  _  _Loss_P
  ! Write(*, '(A,F)')  "Index_PD  _  _Index_PD
  ! Write(*, '(A,F)')  "Index_Sol_PD  _  _Index_Sol_PD

  Call FluidVelocity (AverageV, MaxV, Index_PD, Index_Sol_PD, MudFlw, A_bh, Loss_P)

  AverageV_Tot(MainI) = AverageV
  MaxV_Tot(MainI)    = MaxV

  v = AverageV_Tot(MainI)

  If (Location == 'Den Helder')  D90 = D(MainI,90)/1000  ! [m]
  If (Location == 'Texel')  D90 = D(MainI+31,90)/1000  ! The array D contains all samples and those of DH are on top.

  Call DmaxTransport (MinYg, YP_Tot(MainI), Rho_Inflow, Rho_Grain, g, DmaxTrMinBH, 'BH')
  Call DmaxTransport (MinYg, Tau_Yield_HB_Tot(MainI), Rho_Inflow, Rho_Grain, g, DmaxTrMinHB, 'HB')
  Call DmaxTransport (MaxYg, YP_Tot(MainI), Rho_Inflow, Rho_Grain, g, DmaxTrMaxBH, 'BH')
  Call DmaxTransport (MaxYg, Tau_Yield_HB_Tot(MainI), Rho_Inflow, Rho_Grain, g, DmaxTrMaxHB, 'HB')

  DmaxTrMinBH_Tot(MainI) = DmaxTrMinBH
  DmaxTrMinHB_Tot(MainI) = DmaxTrMinHB
  DmaxTrMaxBH_Tot(MainI) = DmaxTrMaxBH
  DmaxTrMaxHB_Tot(MainI) = DmaxTrMaxHB

  Call MaxDiameterSusp (D_max, v, g, D_bh, Rho_Grain, Rho_Bf(MainI), SC(MainI))

  D_Max_Tot(MainI) = D_Max

  JointNo = Joint (Index_Sol_Sur)

  Call MudPressure (Location, DateTimeSample, P_Real, Activity, JointNo, Joint_MP, Mud_P)

  Mud_P_Tot(MainI) = Mud_P

  If (Delta_Tot_HB(MainI) < Delta_Tot_BH(MainI)) Then  ! Means that the Herschel Bulkley method is more precise than the Bingham method.
    Tau_Y = Alpha * Tau(1)
  Else
    Tau_Y = YP_Tot(MainI)
  End If

  Call MaxPlugRadius (Tau_Y, MD(Index_Sur), Mud_P_Tot(MainI), MaxPlugR)
  MaxPlugR_Tot(MainI) = MaxPlugR

  Write(*, '(A,F)')  "Mud_P_Tot(MainI)  :  _Mud_P_Tot(MainI)

  Call MaxPlugRadius2 (Tau_Y, MD(Index_Sur), Mud_P_Tot(MainI), MaxPlugR, D_bh, D_ds)

  D15 = D(MainI,15)/1000.d0  ! [m]

  If (MudFlw(Index_PD) == 0.d0) Then
    Do i=1,301
      If (MudFlw(Index_PD-i) > 0.d0) Then
        Index_PDMF = Index_PD-i
      End If
    End Do
  Else
    Index_PDMF = Index_PD
  End If

  Call InfiltrationLossBTL (Location, g, Loss_BH, SourceDepthTot(MainI), YP_Tot(MainI), DI5, D_bh, Mud_P_Tot(MainI), PI)
InfiltLossBTBLBH_Tot(MainI) = Loss_BH

If (Runoption == 2) Then
  Write(*,'(A,F9.14)',"Loss_BH: ",Loss_BH)
  Write(*,'(A,F9.14)') "SourceDepthTot(MainI) : ",SourceDepthTot(MainI)
  Write(*,'(A,F9.14)') "YP_Tot(MainI) : ",YP_Tot(MainI)
  Write(*,'(A,F9.14)') "D15 : ",D15
  Write(*,'(A,F9.14)') "D_bh : ",D_bh
  Write(*,'(A,F9.14)') "Mud_P_Tot(MainI) : ",Mud_P_Tot(MainI)
End If

LossPercentage_BH = ((Loss_BH * 1000.0*60.0) / (MudFlw(Index_PDMF))) * 100.0
InfiltLossPerc_BTBLBH_Tot(MainI) = LossPercentage_BH

If (Runoption == 2) Then
  Write(*,'(A,I4)') "Index_PD : ",Index_PD
  Write(*,'(A,I4)') "Index_PDMF : ",Index_PDMF
  Write(*,'(A,F9.14)') "MudFlw(Index_PDMF) : ",MudFlw(Index_PDMF)
End If

Write(*,'(A,F5.2)') "InfiltLossBTBLBH_Tot(MainI) : ",InfiltLossBTBLBH_Tot(MainI)
Write(*,'(A,F5.2)') "Loss Percentage BH : ",LossPercentage_BH

Call InfiltLossBTBL(Location, g, Loss_HB, SourceDepthTot(MainI), Tau_Yield_HB_Tot(MainI), D15, D_bh, Mud_P_Tot(MainI), PI)
InfiltLossBTBLBH_Tot(MainI) = Loss_HB

LossPercentage_HB = ((Loss_HB * 1000.0*60.0) / (MudFlw(Index_PDMF))) * 100.0
InfiltLossPerc_BTBLBH_Tot(MainI) = LossPercentage_HB

Write(*,'(A,F5.2)') "InfiltLossBTBLBH_Tot(MainI) : ",InfiltLossBTBLBH_Tot(MainI)
Write(*,'(A,F5.2)') "Loss Percentage HB : ",LossPercentage_HB

Write(OutputLosses,'(A,3(A,F7.2))') Sample_Code,";",Loss_P*100,";",InfiltLossPerc_BTBLBH_Tot(MainI)

C1 = 0.124d0 ! average between min and max of literature values for C1
Max_d_BH = MaxDiameterPlasticity(C1, YP_Tot(MainI), g, Rho_Bf(MainI), Rho_Grain)
Max_d_BH_Tot(MainI) = Max_d_BH

v_settle_BH = SettlingVel(g, Rho_Bf(MainI), Rho_Grain, Max_d_BH_Tot(MainI), PV_Tot(MainI), YP_Tot(MainI))
v_settle_BH_Tot(MainI) = v_settle_BH

Gamma_Ann1 = GammaAnn1(v, D_bh)
Gamma_AnnTot1(MainI) = Gamma_Ann1

Gamma_Ann2 = GammaAnn2(v, D_bh, D_ds)
Gamma_AnnTot2(MainI) = Gamma_Ann2

Write(*,'(A,F9.14)',"Gamma_Ann, method 1 and 2: ",Gamma_AnnTot1(MainI),Gamma_AnnTot2(MainI)

DeltaWeightBH1 = 0
Do i=1,size(GammaY)
  DeltaWeightBH1 = DeltaWeightBH1 + ((Tau_Calc_BH(i) - Tau(i))**2.0) * (1.0 / (Gamma_AnnTot1(MainI) - GammaY(i)**2.0))
End Do

DeltaWeightBH2 = 0
Do i=1,size(GammaY)
  DeltaWeightBH2 = DeltaWeightBH2 + ((Tau_Calc_BH(i) - Tau(i))**2.0) * (1.0 / (Gamma_AnnTot2(MainI) - GammaY(i)**2.0))
End Do

DeltaWeightHB1 = 0
Do i=1,size(GammaY)
  DeltaWeightHB1 = DeltaWeightHB1 + ((Tau_Calc_HB(i) - Tau(i))**2.0) * (1.0 / (Gamma_AnnTot1(MainI) - GammaY(i)**2.0))
End Do

DeltaWeightHB2 = 0
Do i=1,size(GammaY)
  DeltaWeightHB2 = DeltaWeightHB2 + ((Tau_Calc_HB(i) - Tau(i))**2.0) * (1.0 / (Gamma_AnnTot2(MainI) - GammaY(i)**2.0))
End Do
Gamma_Ann_Tot1 (MainI) - GammaY(i)**2.0)
End Do

DeltaWeightHB2 = 0
Do i = 1, size(GammaY)
   DeltaWeightHB2 = DeltaWeightHB2 + (( Tau_Calc_HB(i) - Tau(i))**2.0) * (1.0 / ((Gamma_Ann_Tot2 (MainI) - GammaY(i))**2.0))
End Do

If (DeltaWeightBH1 < DeltaWeightHB1) Then
   DeltaWBH1_No = DeltaWBH1_No + 1
   Write(*, '(A)') "Gamma_Ann 1: Bingham, the best fit to the data."
Else If (DeltaWeightBH1 > DeltaWeightHB1) Then
   DeltaWHB1_No = DeltaWHB1_No + 1
   Write(*, '(A)') "Gamma_Ann 1: Herschel-Bulkley, the best fit to the data."
End If

If (DeltaWeightBH2 < DeltaWeightHB2) Then
   DeltaWBH2_No = DeltaWBH2_No + 1
   Write(*, '(A)') "Gamma_Ann 2: Bingham, the best fit to the data."
Else If (DeltaWeightBH2 > DeltaWeightHB2) Then
   DeltaWHB2_No = DeltaWHB2_No + 1
   Write(*, '(A)') "Gamma_Ann 2: Herschel-Bulkley, the best fit to the data."
End If

Re_Gen_BH = ReynoldsGen (Rho_Bf (MainI) , D_bh , v , YP_Tot (MainI) , n_BH , PV_Tot (MainI) )
!Rho_bf , D, v , tau_yield , n, K, for bingham K=0 and n=1, deriving (Madlener2009) results in K
is replaced by the viscosity at very high shear thus PV.
Re_Gen_BH_Tot (MainI) = Re_Gen_BH

Re_Gen_HB = ReynoldsGen (Rho_Bf (MainI) , D_bh , v , Tau_Yield_HB_Tot (MainI) , n_Tot (MainI) , K_Tot (MainI) )
!Rho_bf , D, v, tau_yield , n.K
Re_Gen_HB_Tot (MainI) = Re_Gen_HB

Write(*, '(A)') "Check Fluid data 'Yes'"

Case( 'No' )

AverageV = 0.0
MaxV = 0.0
AverageV_Tot (MainI) = 0.0
MaxV_Tot (MainI) = 0.0
v = AverageV_Tot (MainI)
D90 = 0.0

Write (Message, '(A)') "No Fluid data means fluid flow calculations are not possible."
Call WriteLog (1, trim (Message))

DmaxTrMinBH = 0.0
DmaxTrMinBH_Tot (MainI) = DmaxTrMinBH
DmaxTrMinHB = 0.0
DmaxTrMinHB_Tot (MainI) = DmaxTrMinHB
DmaxTrMaxBH = 0.0
DmaxTrMaxBH_Tot (MainI) = DmaxTrMaxBH
DmaxTrMaxHB = 0.0
DmaxTrMaxHB_Tot (MainI) = DmaxTrMaxHB

D_Max_Tot (MainI) = 0.0
Mud_P = 0.0
Mud_P_Tot (MainI) = 0.0
MaxPlugR = 0.0
MaxPlugR_Tot (MainI) = 0.0
LossPercentage_BH = 0.0
InfilLossPerc_BH_Tot (MainI) = 0.0
LossPercentage_HB = 0.0
InfilLossPerc_HB_Tot (MainI) = 0.0

Write (OutputLosses , '(A,3(A,F7.2))') Sample_Code , ""," , Loss_P*100."," , InfilLossPerc_BH_Tot (MainI) , ""," , InfilLossPerc_HB_Tot (MainI)
Max_d_BH = 0.d0
Max_d_BH_Tot(MainI) = Max_d_BH
v_settle_BH = 0.d0
v_settle_BH_Tot(MainI) = v_settle_BH
Gamma_Ann1 = 0.d0
Gamma_Ann_Tot1(MainI) = Gamma_Ann1
Gamma_Ann2 = 0.d0
Gamma_Ann_Tot2(MainI) = Gamma_Ann2
DeltaWeightBH1 = 0.d0
DeltaWeightBH2 = 0.d0
DeltaWeightHB1 = 0.d0
DeltaWeightHB2 = 0.d0
Re_Gen_BH = 0.d0
Re_Gen_BH_Tot(MainI) = Re_Gen_BH
Re_Gen_HB = 0.d0
Re_Gen_HB_Tot(MainI) = Re_Gen_HB

Write(*,'(A)') 'Check Fluid data 'No'

Case Default
Write(*,'(A)') 'Something went wrong in the selection of the fluid data case loop.'
Read(*,*)
End Select

!-----------------------------------------------------
! Writes Files
!-----------------------------------------------------
! Adds a comma-delimited title in the file with the content and the unit of the vector

Write(OutputSamples, '(4(A),1,8(A,F),A,15..14(A,F))') Sample_Code, '', DateTimeSample, '', MainI, '', SourceLocAvAll(MainI), '', SamplingDepthTot(MainI), '', SamplingAwayTot(MainI), '', SamplingRightTot(MainI), '', SourceDepthTot(MainI), '', SourceRightTot(MainI), '', PositionBitAll(MainI), '', PumpTime, '', Tau_Yield_HB_Tot(MainI), '', K_Tot(MainI), '', n_Tot(MainI), '', Alpha_Tot(MainI), '', Delta_Tot_HB_BH_Tot(MainI), '', PV_Tot(MainI), '', YP_Tot(MainI), '', Delta_Tot_BH_BH_Tot(MainI), '', Mud_P_Tot(MainI), '', MaxPlugR_Tot(MainI), '', AverageV_Tot(MainI), '', D_Max_Tot(MainI), '', Max_d_BH_Tot(MainI), '', v_settle_BH_Tot(MainI)

Write(OutputSamplesLoc, '(2(A),6(F))') Sample_Code, DateTimeSample, SamplingAwayTot(MainI), SamplingDepthTot(MainI), SamplingRightTot(MainI), SourceAwayTot(MainI), SourceDepthTot(MainI), SourceRightTot(MainI)

Write(OutputFluidProperties, '(3(A),8(A,F))') Sample_Code, '', DateTimeSample, '', Tau_Yield_HB_Tot(MainI), '', K_Tot(MainI), '', n_Tot(MainI), '', Alpha_Tot(MainI), '', Delta_Tot_HB_BH_Tot(MainI), '', PV_Tot(MainI), '', YP_Tot(MainI), '', Delta_Tot_BH_BH_Tot(MainI)

Write(OutputFluidPropTex, '(2(A),8(F))') Sample_Code, DateTimeSample, Tau_Yield_HB_Tot(MainI), K_Tot(MainI), n_Tot(MainI), Alpha_Tot(MainI), Delta_Tot_HB_BH_Tot(MainI), PV_Tot(MainI), YP_Tot(MainI), Delta_Tot_HB_BH_Tot(MainI)

Write(OutputTransportProperties, '(3(A),16(A,F))') Sample_Code, '', DateTimeSample, '', MaxPlugR_Tot(MainI), '', AverageV_Tot(MainI), '', D_Max_Tot(MainI), '', Gamma_Ann_Tot(MainI), '', Gamma_Ann_Tot2(MainI), '', Re_Gen_BH_BH_Tot(MainI), '', Re_Gen_HB_BH_Tot(MainI), '', D90, '', DmaxTrMinBH_Tot(MainI), '', DmaxTrMaxBH_Tot(MainI), '', DmaxTrMinHB_Tot(MainI), '', DmaxTrMaxHB_Tot(MainI), '', DeltaWeightBH1, '', DeltaWeightBH2, '', DeltaWeightHB1, '', DeltaWeightHB2

Write(OutputTransportPropTex,'(2(A),16(F))') Sample_Code, DateTimeSample, MaxPlugR_Tot(MainI), AverageV_Tot(MainI), D_Max_Tot(MainI), Gamma_Ann_Tot(MainI), Gamma_Ann_Tot2(MainI), Re_Gen_BH_BH_Tot(MainI), Re_Gen_HB_BH_Tot(MainI), D90, DmaxTrMinBH_Tot(MainI), DmaxTrMaxBH_Tot(MainI), DmaxTrMinHB_Tot(MainI), DmaxTrMaxHB_Tot(MainI), DeltaWeightBH1, DeltaWeightBH2, DeltaWeightHB1, DeltaWeightHB2

If(RunOption == 2) Write(TableHans, '(3(A),6(A,F))') Sample_Code, '', Sieve_Code(MainI), '', SamplingAwayTot(MainI), '', SourceAwayTot(MainI), '', MD(Index_Sur), '', MX Index_Sol_Sur, '', SamplingDepthTot(MainI), '', SourceDepthTot(MainI)
If (RunOption == 2) Write(*,'(A)') "Check 9"

Write(*,'(A)') "Check Writing Files"

!--------------------------------------------------------
! Calling Gnuplot
!--------------------------------------------------------
! Enables the creation of graphs in order to visually check the answers without requiring
! the creation of two graphs in separate .csv files and merging them.
!--------------------------------------------------------

If (Index_Sol_PD > 0 .and. Index_Sol_Sur > 0) Then ! Efficiency measure if no solution
  is found. Index_Sol... <=0 than no plots are made either since they would contain
  nonsense.

Select Case (FluidData)
  Case ('Yes')
    Call f2gpSampleMulti(Sample_Code, Index_PD, Index_Sur, Index_Sol_PD, DateTime_PD,
      TranspLenSum, '2a', DateTime_Sur, MD, '2b', Alpha_All, Delta_All, '2c', GammaY, Tau, '3a',
      Tau_Calc_HB, '3b', Tau_Calc_BH, '3c', 'Date Time', 'Length [m]', 'Alpha [-]', 'Delta', 'Shear
      Rate [1/s]', 'Shear Stress [Pa]', 'Sample Parameters', 'Intersection', 'Alpha vs. Delta', 'Measurements
      vs. Calculations', 'Deviation', 'Measurements', 'Herschel Bulkley', 'Bingham', 'Pumped Length', 'Measured Distance', ScenarioNo)
    Write(Message,'(A)') "Fluid data is determined so 2 plots are made."
    Call WriteLog(1,Message)
  Case ('No')
    Call f2gp(Sample_Code, Index_PD, Index_Sur, Index_Sol_PD, DateTime_PD, TranspLenSum, '2a',
      DateTime_Sur, MD, '2b', 1, 'DateTime', 'Length [m]', 'Pumped Length', 'Measured Distance', ScenarioNo)
    Write(Message,'(A)') "Fluid data is not determined so only the intersection plot
      can be made."
    Call WriteLog(1,Message)
  Case Default
    Write(*,'(A)') "Something went wrong with the 'FluidData' case."
    Read(*,*)
    Stop
End Select
End If

If (RunOption == 2) Write(*,'(A)') "Check 10"

! Deallocation so the arrays can be reallocated for next cycles of the mainloop in which
! the content and size can be different.
!--------------------------------------------------------
Deallocation (TimeInt_PD, PumpVol, TranspLen, TranspLenSum)

If (RunOption == 2) Write(*,'(A)') "Check 11"

! If (RunOption == 2) Close(SampleLog)

!--------------------------------------------------------

End Do Mainloop

Write(*,*)
Write(*,'(A)') "The Main Loop has finished."
Write(*,*)
Write(*,'(A)') ""
E. Code of the SampleSource Program

Call WriteLog(2, trim(Message), SampleSourceLog)
Write(Message, '(A,13,A,13,A,13,A)')  " For Gamma_Ann was Bingham ", DeltaWHB1_No, " times the best fit, the Herschel–Bulkley model was " , DeltaWHB1_No, " times the best fit."
Call WriteLog(2, trim(Message), SampleSourceLog)
Write(Message, '(A,13,A,13,A,13,A)')  " For Gamma_Ann2 was Bingham ", DeltaWHB2_No, " times the best fit, the Herschel–Bulkley model was " , DeltaWHB2_No, " times the best fit."
Call WriteLog(2, trim(Message), SampleSourceLog)
Write(\(\ast,\) \(\ast,(A)\) ) "

--------------------

! Calling Gnuplot

If (RunOption == 2) Write(\(\ast,\) \(\ast,(A)\) ) "Check F1"

! Times overview
Write(filename, '(A,12,2)') trim(short(Location))// '_Times_vs_Length_ _Scenario_'.ScenarioNo
Write(title, '(A,12,2)') trim(Location)// 'Times vs. Length Scenario'.ScenarioNo
Call f2gp2(DateTime_Samples, TimeIntAvAll, PumpTimesAll, DateTime_Sur, Away, 1, filename, trim(title), 'Date Time', 'Time [s]', 'Length [m]', 'Transport Time [s]', 'Length [m]', 'Pump Time [s]')

! (x1,y1,y1a,y1b,x2,y2,type of graph, x-axis title, y-axis title1, y-axis title2, graph1 title, graph2 title, graph3 title)
If (RunOption == 2) Write(\(\ast,\) \(\ast,(A)\) ) "Check F2"

! Length vs. Source location
Write(filename, '(A,12,2)') trim(short(Location))// '_Length_vs_Source_ _Scenario_'.ScenarioNo
Write(title, '(A,12,2)') trim(Location)// 'Length vs. Source Scenario'.ScenarioNo
Call f2gp3(Location, DateTime_Samples, SourceLocAvAll, '3a', DateTime_Sur, MD, '2b', 1, trim(filename), trim(title), 'Date Time', 'Length [m]', 'Source Location [m]', 'Length [m]')
! (x1,y1,x2,y2,type of graph, title, x-axis title, y-axis title, graph1 title, graph2 title)
If (RunOption == 2) Write(\(\ast,\) \(\ast,(A)\) ) "Check F3"

! Lag Diagram
Write(filename, '(A,12,2)') trim(short(Location))// '_Lag_Diagram_Scenario_'.ScenarioNo
Write(title_graph, '(A,12,2)') trim(short(Location))// 'Lag Diagram Scenario'.ScenarioNo
Call f2gpLagD(Location, title_graph, PositionBitAvAll, SourceLocAvAll, 1, Position Head [m]', 'Source Location [m]', trim(filename), 'V T = V.L.', '1k', '1l', '1m', '1n')
! (xdata, ydata, type of graph, x-axis title, y-axis title, title, graph title)
Open(10, file = filefolder_out // 'ExtraFiles' // trim(filename) // '.txt')
Do i = 1, size(PositionBitAll)
Write(10, '(2(F))') PositionBitAll(i), SourceLocAvAll(i)
End Do
Close(10)
If (RunOption == 2) Write(\(\ast,\) \(\ast,(A)\) ) "Check F4"

Write(\(\ast,\) \(\ast,(A)\) ) "Check Graphs"

!--------------------

End of program

! Deallocation (for possible reruns within the program in the future)

Deallocate(DateTime_PD, Bit_DPT, Hol_DPT, Car_POS, Car_SPD, Pull, Push, Torq_pos, Rot_SPD)
Deallocate(M_Press, MudFlw, Torq_neg, Press_1, Press_2, Press_3, HS_D)
Deallocate(DateTime_Sur, Joint, CL, MD, InclinationRaw, AzimuthRaw, Btot, Dip)
Deallocate(ElevationCalc, ElevationMGS, Away, R_Calc, R_MGS)
Deallocate(DateTime_Samples, Sample_ID, Temp, EC, pH, MF, Rho_Bf, SC, RPM_600, RPM_300, RPM_200, RPM_100)
Dealocate (RPM_60, RPM_30, RPM_6, RPM_3, Gel10s, Gel10min)
Dealocate (Activity, Joint_MP, Length_MP)
If (Location == 'Den Helder') Dealocate (P_Max1, P_Max2, P_Calc, P_Real)
If (Location == 'Texel') Dealocate (P_Real, P_Min)
If (RunOption == 2) Write (*, '(A)') "Check Z1"
Dealocate (PumpTimesAll, TimeIntAvAll)
Dealocate (SourceLocAvAll, PositionBitAll, Tau_Yield_HB_Tot, K_Tot, n_Tot)
Dealocate (PV_Tot, YP_Tot, Alpha_Tot, Delta_Tot_HB, Delta_Tot_BH)
Dealocate (Mud_P_Tot, MaxPlugR_Tot)
If (RunOption == 2) Write (*, '(A)') "Check Z2"
Dealocate (SamplingDepthTot, SamplingAwayTot, SamplingRightTot)
Dealocate (SourceDepthTot, SourceAwayTot, SourceRightTot)
If (RunOption == 2) Write (*, '(A)') "Check Z3"
Dealocate (Sieve_Code, D)
! Dealocate (Sieve_Code_V, D_V)
Dealocate (D_Max_Tot)
Dealocate (AverageV_Tot, MaxV_Tot)
Dealocate (InfiltirationLossBTL_BH_Tot, InfiltirationLossBTL_HB_Tot)
Dealocate (Max_D_BH_Tot, v_settle_BH_Tot)
Dealocate (Gamma_Ann_Tot1)
Dealocate (Gamma_Ann_Tot2)
Dealocate (Re_Gen_BH_Tot, Re_Gen_BH_Tot)
Dealocate (DmaxTrMinBH_Tot, DmaxTrMinBH_Tot, DmaxTrMaxBH_Tot, DmaxTrMaxHB_Tot)
If (RunOption == 2) Write (*, '(A)') "Check Z4"
Write (*, '(A)') "Check Deallocation"
! Closing Files
Close (InputDim)
Close (InputCSV_PD)
Close (InputCSV_Sur)
Close (InputMP)
Close (SampleData)
Close (OutputLosses)
Close (OutputSamples)
Close (OutputSamplesLoc)
Close (OutputFluidProperties)
Close (OutputFluidPropTex)
Close (OutputTransportProperties)
Close (OutputTranspPropTex)
Close (OutputLosses)
Close (SieveResults)
! Close (SieveResults_V)
If (RunOption == 2) Close (TableHans)
If (RunOption == 2) Write (*, '(A)') "Check Z5"
Close (SampleSourceLog)
Write (*, '(A)') "Check Closing Files"
!---------------------------------------------------------------------
Select Case (ProgramOption)
Case 1
    Write (*, *)
Write (*, '(A)') "

Write (*, '(A)') "The chosen part of the program is executed, to go back to the start type '1' or type '0' to close the program:"

Read(*,*) EndOption

If (EndOption == 0) Stop
If (EndOption == 1) Go To 100

Case (2)
If (ScenarioNo == 9) Then
Write (*,'(A)') "All scenarios for location " /trim(Location)="/" are run."
Write (*,'(A)') "Press '0' to close the program or '1' to continue with the other location."
Read(*,*) EndOption
Select Case (EndOption)
Case (0)
Stop
Case (1)
Write(*,*)
Write(*,'(A)') "

Write(*,*) Go to 100
Case Default
Write(*,'(A)') "Unknown option program is terminated."
Read(*,*)
Stop
End Select
End If
ScenarioNo = ScenarioNo + 1
Go To 500

Case (3)
If (ScenarioNo == ScenarioNo2) Then
Write(*,'(A)') "Both scenarios have run, program will close."
Read(*,*)
Stop
Else
ScenarioNo = ScenarioNo2
If (ScenarioNo == 0) Then
Write(*,'(A)') "The scenario has run, the program will close."
Read(*,*)
Stop
End If
Write(*,*)
Write(*,'(A)') "The second chosen scenario will start now."
Write(*,*)
Write(*,'(A)') "

Go to 500
End If

Case (4)
If (ScenarioNo == 9) Then
FullRunCounter = FullRunCounter + 1
If (FullRunCounter == 1) Then
Write(*,'(A)') "All scenarios for location '"//trim(Location)//" are run."
Write(*,'(A)') "The second location will now be done"
If (Location == 'Texel') Then
  Location = 'Den Helder'
  ScenarioNo = 1
  Write(*,*), Write(*,'(A)') "
                            Go to 500
Else If (Location == 'Den Helder') Then
  Location = 'Texel'
  ScenarioNo = 1
  Write(*,*), Write(*,'(A)') "
                           Go to 500
End If
Else If (FullRunCounter == 2) Then
  Write(*,'(A)') "
                            Go to 500
  Write(*,*), Write(*,'(A)') "
                           All scenarios for both locations have run, the program will now close."
  Write(*,*), Write(*,'(A)') "
                       Read(*,*)
                      Stop
End If
End Select
End Program SourceSample

Module SourceModule
  Implicit None
  Contains
    !--------------------------------------------------------
    ! FUNCTIONS
    !--------------------------------------------------------
  Integer Function LineCount(File)
    ! Calculates the amount of lines in a file
    ! Based on http://web.utah.edu/thorne/computing/Handy_Fortran_Tricks.pdf, bottom last page
  Implicit None
  Integer :: File,nlines
  Integer :: io
  nlines = 0 ! Number of lines needs a start value
  Do Read (File,*),IOSTAT=io
    If (io > 0) Then
      Write(*,*), Write(*,'(A)') "Check input. Something is wrong..."
      Exit
  End Do
Else If (io < 0) Then
  ! WRITE(*,*) "End of file is reached." ! No longer necessary once it works.
  Exit
Else
  nlines = nlines + 1
End If
End Do
Rewind(File) ! Reading of the data starts at the top of the file again.
Linecount = nlines
End Function

!---------------------------------------------------------------

Integer Function ColumnCount/File, Separator/
! Counts the number of columns in a file by counting the separators.
! Assumed is that the file has the same number of columns for every line and that the lines
! are separated by a symbol which is not equal to the decimal or thousand sign and not
! equal to the symbol for empty spaces if those are present.
! Based on http://rosettacode.org/wiki/Count_occurrences_of_a_substring#Fortran
Implicit None

Integer :: File
Integer :: nSeparators
Character(*) :: Separator
Character(2048) :: Line
Integer :: Pos, posmin

Read(File,'(A)') Line
nSeparators = 0
Posmin = 1
Do ! Infinite loop, loop runs till the command is reached that closes it.
  Pos = Index(Line(posmin:),Separator)
  If (Pos == 0) Then ! EOF is reached, index returns 0 if substring is not found.
    ColumnCount = nSeparators + 1 ! nColumns is actually the number of separators.
    Rewind(File) ! Reading of the data starts at the top of the file again.
    Return ! Closes the do-loop and thus the function.
  End If
  nSeparators = nSeparators + 1
  posmin = posmin + pos + len(Separator)
End Do
End Function

!---------------------------------------------------------------

Character(100) Function Short(Long)
! Enables a shortcut to abbreviated words.
Implicit None

Character(*) :: Long

Select Case(Long)
Case('Den Helder')
  Short = 'DH'
Case('Texel')
  Short = 'T'
Case Default
  Write(*,'(A)') "There is no programmed short for this word."
End Select
Real Function ReynoldsGen(Rho_bf, D, v, tau_yield, n, K)
! Generalized Reynolds Number applicable to HB - extended, HB, PL, BH and Newtonian fluids, Madlener2009.
Implicit None
Real :: Rho_bf, D, v, tau_yield, n, K, m
m = (n * K * (8. d0 * v / D)**n) / (tau_yield + K * (8. d0 * v / D)**n)
ReynoldsGen = (Rho_bf * (v**(2. d0 - n)) * (D**n)) / ((tau_yield / 8. d0)**(D / v)**n) + K*
(((3. d0*m + 1. d0) / (4. d0*m)**n) * (8. d0**(n-1)))
End Function

Real Function Hedstrom(Rho_bf, D, Tau_Y, Viscosity)
! The Hedstrom Number
Implicit None
Real :: Rho_bf, D, Tau_Y, Viscosity
Hedstrom = (Rho_bf * Tau_Y * D * D) / (Viscosity * Viscosity)
End Function

Real Function MaxDiameterPlasticity(Cl, Tau_Y, g, Rho_bf, Rho_Grain)
! Based on plasticity theory, 0.048 <= cl <= 0.2
Implicit None
Real :: Cl, Rho_bf, Rho_Grain, Tau_Y, g
MaxDiameterPlasticity = (cl * Tau_Y) / (g * (Rho_Grain - Rho_bf))
End Function

Real Function SettlingVel(g, Rho_bf, Rho_grain, d, viscosity, Tau_y)
! Based on Wan 1985
Implicit None
Real :: g, Rho_bf, Rho_grain, d, viscosity, Tau_y
SettlingVel = (((Rho_grain - Rho_bf) * g * d * d) / (18 * viscosity)) * ((7 * d * Tau_y) / (24 * viscosity))
End Function

Real Function GammaAnn1(v, D_bh)
! Based on Denekamp 2000
Implicit None
Real :: v, D_bh
R_fac = 1.5 d0 ! Value found in Denekamp 2000
GammaAnn1 = R_fac * ((8 * v) / D_bh)
End Function
Real Function GammaAnn2(v, D_bh, D_ds)
! Based on Mendes 2005
Implicit None
Real :: v, D_bh, D_ds, Ro, Ri
Ro = D_bh / 2. d0
Ri = D_ds / 2. d0
GammaAnn2 = v / (Ro - Ri)
End Function

! SUBROUTINES

! Common, not program specific

Subroutine WriteLog (WriteOption, Message, Logfile1, Logfile2, Logfile3)
! Depending on the writeoption it writes messages to the screen (1), screen and up to three
! logfiles (2) or just up to three logfiles (3).
Implicit None
Integer :: WriteOption
Character(5) :: WFormat
Character(*) :: Message
Integer, optional :: Logfile1, Logfile2, Logfile3
WFormat = '(A)'
Select Case (WriteOption)
Case (1)
   Write(*, WFormat) trim (Message)
Case (2)
   Write(*, WFormat) trim (Message)
   If (present(Logfile1)) Then
      Write(Logfile1, WFormat) trim (Message)
      Write(Logfile1,*)
   End If
   If (present(Logfile2)) Then
      Write(Logfile2, WFormat) trim (Message)
      Write(Logfile2,*)
   End If
   If (present(Logfile3)) Then
      Write(Logfile3, WFormat) trim (Message)
      Write(Logfile3,*)
   End If
Case (3)
   If (present(Logfile1)) Then
      Write(Logfile1, WFormat) trim (Message)
      Write(Logfile1,*)
   End If
   If (present(Logfile2)) Then
      Write(Logfile2, WFormat) trim (Message)
      Write(Logfile2,*)
   End If
   If (present(Logfile3)) Then
      Write(Logfile3, WFormat) trim (Message)
      Write(Logfile3,*)
End Select
End If

Case Default
  Write(*,'(A)') "The Write option is unknown."
End Select
End Subroutine

! ----------------------------------------
! Reading files
! ----------------------------------------

Subroutine ReadingFileCSV_PD(File, DateTime_PD, Bit_DPT, Hol_DPT, Car_POS, Car_SPD, Pull, Push, Torq_pos, Rot_SPD, M_Press, MudFlw, Torq_neg, Press_1, Press_2, Press_3, HS_D, Nlines_PD)
  ! Containing header, 1st column is date and time (dd-mm-yyyy hh:mm:ss format), just real numbers, the other columns. Columns separated by ",

Implicit None

Integer :: File
Character(100) :: Header
Integer, allocatable :: DateTime_PD(:), Time(:)
Integer, allocatable :: Bit_DPT(:), Hol_DPT(:), Car_POS(:), Car_SPD(:), Pull(:,), Push(:,), Torq_pos(:)
Integer, allocatable :: Rot_SPD(:,), M_Press(:,), MudFlw(:,), Torq_neg(:,), Press_1(:,), Press_2(:,), Press_3(:,), HS_D(:)
Character(100) :: nlines, i, io, nlines_data, Nlines_PD
Integer :: Message
Integer :: SampleSourceLog=25

nlines = LINECOUNT(File)
nlines_data = nlines-1 ! File contains headers.

Allocate(DateTime_PD(nlines_data))
Allocate(Time(nlines_data))
Allocate(Bit_DPT(nlines_data))
Allocate(Hol_DPT(nlines_data))
Allocate(Car_POS(nlines_data))
Allocate(Car_SPD(nlines_data))
Allocate(Pull(nlines_data))
Allocate(Push(nlines_data))
Allocate(Torq_pos(nlines_data))
Allocate(Rot_SPD(nlines_data))
Allocate(M_Press(nlines_data))
Allocate(MudFlw(nlines_data))
Allocate(Torq_neg(nlines_data))
Allocate(Press_1(nlines_data))
Allocate(Press_2(nlines_data))
Allocate(Press_3(nlines_data))
Allocate(HS_D(nlines_data))

Read(file,'(A100)') Header

Do i=1,nlines_data
  Read(file,*), IOSTAT=io) DateTime_PD(i), Time(i), Bit_DPT(i), Hol_DPT(i), Car_POS(i), Car_SPD(i), Pull(i), Push(i), Torq_pos(i), Rot_SPD(i), M_Press(i), MudFlw(i), Torq_neg(i), Press_1(i), Press_2(i), Press_3(i), HS_D(i)
  DateTme_PD(i) = trim(adjustl(DateTime_PD(i)))/"" //trim(adjustl(Time(i)))))
  ! Space is counted as separator as well...
  If (io > 0) Then
    Write(*,*) "Check input, something is wrong, unit wise..."
  Else If (io < 0) Then
    Write(*,*) "End of file is reached, before end of loop."
  Exit
  End If
End Do
Write("The ProData file is read and ",nlines_data," lines containing data are found.", Message)
Call WriteLog(1, trim(Message))
Nlines_PD = nlines_data
End Subroutine

---

Subroutine ReadingFileCSV_Sur(File, nlines_sur, DateTime_Sur, Joint, CL, MD, Inclination_RAW, Azimuth_RAW, Btot, Dip, Elevation_Calc, Elevation_MGS, Away, R_Calc, R_MGS)
! Containing header, 1st column date (dd-mm-yy), 2nd column time (hh:mm), just real numbers the other columns, besides number 11 which contains integers. Columns separated by ",
Implicit None

Integer :: File
Character(130) :: Header
Real, Allocatable :: Date_I(:)
Real, Allocatable :: Time_R(:)
Integer, Allocatable :: Joint(:), Btot(:)
Real, Allocatable :: CL(:), Inclination_RAW(:), Azimuth_RAW(:)
Real, Allocatable :: Dip(:), Elevation_Calc(:), Elevation_MGS(:), R_Calc(:)
Real(8), Allocatable :: Away(:), MD(:)
Real, Allocatable :: nlines, i, io, nlines_data, Nlines_sur
Character(19), Allocatable :: DateTime_Sur(:)
Character(1000) :: Message
Integer :: SampleSourceLog=25

nlines = LINECOUNT(File)
nlines_data = nlines-1 ! File contains headers.
Allocate(Joint(nlines_data))
Allocate(Btot(nlines_data))
Allocate(CL(nlines_data))
Allocate(MD(nlines_data))
Allocate(Inclination_RAW(nlines_data))
Allocate(Azimuth_RAW(nlines_data))
Allocate(Dip(nlines_data))
Allocate(Elevation_Calc(nlines_data))
Allocate(Elevation_MGS(nlines_data))
Allocate(Away(nlines_data))
Allocate(R_Calc(nlines_data))
Allocate(R_MGS(nlines_data))
Allocate(Date_I(nlines_data)) ! Only used in this subroutine so deallocated at the end.
Allocate(Time_R(nlines_data))

Read(file, '(A100)') Header
Do i=1, nlines_data!-1 ! The "-1" accounts for the line read two lines above this sentence.
   Read(file, '(15.1x,F17.15,F1.9,F4.3,2(F))', IOSTAT=io) Date_I(i), Time_R(i), Joint(i), CL(i), MD(i), Inclination_RAW(i), Azimuth_RAW(i), Btot(i), Dip(i), Elevation_Calc(i), Elevation_MGS(i), Away(i), R_Calc(i), R_MGS(i)
   If (io > 0) Then
      Write(*,*), "Check input, something is wrong, unit wise..."
      Exit
   Else if (io < 0) Then
      Write(*,*), "End of file is reached, before end of loop."
      Exit
   End If
End Do
Nlines_sur = nlines_data
Call DateTimeFromSur(DateTime_Sur, Date_I, Time_R, Nlines_sur)
Write("The Survey file is read and ",nlines_data," lines containing data..."
are found."

Call WriteLog(1,trim(Message))
Deallocation(Date_I,Time_R)
End Subroutine

Subroutine ReadingFileSampleData(Location, File, No_Samples, DateTime_Samples, Sample_ID, Temp, EC, pH, MF, Rho, SC, RPM_600, RPM_300, RPM_200, RPM_100, RPM_60, RPM_30, RPM_6, RPM_3, Gel10s, Gel10min)

! Containing header, 1st column date (dd-mm-yy), 2nd column time (hh:mm), just real numbers the other columns, besides number 11 which contains integers. Columns separated by ",".
Implicit None

Integer :: File
Character (130) :: Header
Real, Allocatable :: Date_I(:)
Integer, Allocatable :: Time_R(:)
Character (19), Allocatable :: DateTime_Samples(:)
Character (len =:), Allocatable :: Sample_ID(:)
Integer, Allocatable :: Temp(:), EC(:), pH(:), MF(:), Rho(:), SC(:)
Integer, Allocatable :: RPM_600(:), RPM_300(:), RPM_200(:), RPM_100(:), RPM_60(:), RPM_30(:), RPM_6(:), RPM_3(:), Gel10s(:), Gel10min(:)
Character (10) :: Location
Character (1000) :: Message
nlines = LINECOUNT(File)
nlines_data = nlines - 1 ! File contains headers.
Allocate(Date_I(nlines_data))
Allocate(Time_R(nlines_data))
Select Case(Location)
Case('Den Helder')
Allocate(character (14) :: Sample_ID(nlines_data))
Case('Texel')
Allocate(character (11) :: Sample_ID(nlines_data))
Case Default
Write(*,'(A)') "Location is unknown, program is terminated."
Pause
Stop
End Select
Allocate(Temp(nlines_data))
Allocate(EC(nlines_data))
Allocate(pH(nlines_data))
Allocate(MF(nlines_data))
Allocate(Rho(nlines_data))
Allocate(SC(nlines_data))
Allocate(RPM_600(nlines_data))
Allocate(RPM_300(nlines_data))
Allocate(RPM_200(nlines_data))
Allocate(RPM_100(nlines_data))
Allocate(RPM_60(nlines_data))
Allocate(RPM_30(nlines_data))
Allocate(RPM_6(nlines_data))
Allocate(RPM_3(nlines_data))
Allocate(Gel10s(nlines_data))
Allocate(Gel10min(nlines_data))
Read(file,'(A100)') Header
Do i=1,nlines_data
Read(file,'(I5,1x,F17.15,1x,A,1x,3(F5.2,1x),F6.2,1x,2(F5.2,1x),10(I1))',IOSTAT=io) Date_I(i), Time_R(i), Sample_ID(i), Temp(i), EC(i), pH(i), MF(i), Rho(i), SC(i), RPM_600(i), RPM_300(i), RPM_200(i), RPM_100(i), RPM_60(i), RPM_30(i), RPM_6(i), RPM_3(i), Gel10s(i), Gel10min(i)
If (io > 0) Then
Write (*,*) "Check input, something is wrong, unit wise..."
Exit
Else if (io < 0) Then
  Write(*,*) "End of file is reached, before end of loop."
Exit
End If
End Do

No_Samples = nlines_data

Call DateTimeFromSur(DateTime_Samples, Date_I, Time_R, No_Samples)

Write(Message, '(A, I9, A)') "The Sample file is read and ", nlines_data, " lines containing data are found."
Call WriteLog(1, trim(Message))

Dealocate(Date_I, Time_R)

End Subroutine

---

Subroutine ReadingFileMP(File, Location, NLines_MP, Activity, Joint_MP, Length_MP, V1, V2, V3, V4)
!
! Reads the file containing the mud pressures.
! Implicit None

Integer :: File
Character(3), Allocatable :: Activity(:)
Integer, Allocatable :: Joint_MP(:,), P_Max1(:,), P_Max2(:,), P_Calc(:,), P_Real(:,), P_Max(:,), P_Min(:,)
Real, Allocatable :: Length_MP(:)
Integer :: nlines, i, io, nlines_data, NLines_MP
Character(1000) :: Message
Character(10) :: Location
Integer, Optional, Allocatable :: V1(:,), V2(:,)
Integer, Optional, Allocatable :: V3(:,), V4(:)

nlines = LINECOUNT(File)
nlines_data = nlines - 1 ! File contains headers.

Allocate(Activity(nlines_data))
Allocate(Joint_MP(nlines_data))
Allocate(Length_MP(nlines_data))
Allocate(V1(nlines_data))
Allocate(V2(nlines_data))

If (Present(V3)) Allocate(V3(nlines_data))
If (Present(V4)) Allocate(V4(nlines_data))

Read(file,*) ! Reads and skips the first line of the file

Do i = 1, nlines_data
  If (trim(Location) == 'Den Helder') Read(file, '(A3,1x,I3,3,1x,F6.1,4(1x,I4.4))', IOSTAT=io)
  Activity(i), Joint_MP(i), Length_MP(i), V2(i), V3(i), V4(i), V1(i)
  If (trim(Location) == 'Texel') Read(file, '(A3,1x,I3,3,1x,F6.1,2(1x,I4.4))', IOSTAT=io)
  Activity(i), Joint_MP(i), Length_MP(i), V1(i), V2(i)
  If (io > 0) Then
    Write(*,*) "Check input, something is wrong, unit wise..."
    Exit
  Else If (io < 0) Then
    Write(*,*) "End of file is reached, before end of loop."
    Exit
  End If
End Do

Write(Message, '(A, I4, A)') "The Mud Pressure file is read and ", nlines_data, " lines containing data are found."
Call WriteLog(1, trim(Message))
Nlines_MP = nlines_data

End Subroutine

!--------------------------------------------------

Subroutine ReadingFileSieve ( File, Sample_Code,D)
! Reads the sieve file, can be generalized to be applicable to files with variable parameter lengths.
Implicit None

Integer :: File
Integer :: nlines, nlines_data, i, io, j
Character(31), Allocatable :: GefFileName(:)
Character(7), Allocatable :: Boring_Code(:)
Character(3), Allocatable :: NEN5104_Code(:)
Character(7), Allocatable :: Sample_No(:)
Character(14), Allocatable :: Sample_Code(:)
Character(17), Allocatable :: Depth(:)
Character(1) :: lineTemp
Character(2048) :: Line
Character(40), Allocatable :: Var(:)
Character :: Separator
Integer :: nColumns
Character(2) :: Number
Character(7) :: Code
Real, Allocatable :: D(:,:)
Integer :: count

nlines = LineCount( File)
nlines_data = nlines-1 ! File contains headers.
Allocate( D( nlines_data , 99) ) ! (y,x)
Separator = " , "
nColumns = ColumnCount( File, Separator)
Write( *, '(A,110,A)' ) 'The sieve file is read and ', nlines_data, ' lines containing data are found.'
Allocate( Var(nColumns) )
Allocate( GefFileName(nlines_data) )
Allocate( Boring_Code(nlines_data) )
Allocate( NEN5104_Code(nlines_data) )
Allocate( Sample_No(nlines_data) )
Allocate( Sample_Code(nlines_data) )
Allocate( Depth(nlines_data) )

Read( File, '(A)' ) ! Skips headers
Do j = 1, nlines_data
   Read( File, '(A)' ) Line
   posmin = 1
   posmax = -1
   Count = 0
   Do i=1,nColumns
      Pos = index( Line( Posmin : ), " , " )
      If ( Pos == 0 ) Then
         Var(i) = Line( Posmin : )
         Exit
      End If
      Count = count + 1
      If ( Count == 5 ) exit
   Posmax = Posmax + Pos
Var ( i ) = Line ( Posmin : Posmax )
Posmin = Posmin + Pos
End Do

Read ( Var ( 2 ), '(A)' ) Boring_Code ( j )
Read ( Var ( 3 ), '(A)' ) NEN5104_Code ( j )
Read ( Var ( 4 ), '(A)' ) Sample_No ( j )

! Giving the gef results the same code as used in the other files.

Code = Sample_No ( j )
Read ( Code ( Len ( trim ( Code ) ) - 1 : ) , '(A2)' ) Number

If ( Index ( Boring_Code ( j ) , "A1" ) > 0 . and . Index ( Boring_Code ( j ) , "DH" ) > 0 ) Then
Sample_Code ( j ) = "PB1DH−A1−65−" / / Number
Else If ( Index ( Boring_Code ( j ) , "A3" ) > 0 . and . Index ( Boring_Code ( j ) , "DH" ) > 0 ) Then
Sample_Code ( j ) = "PB1DH−A3−85−" / / Number
Else If ( Index ( Boring_Code ( j ) , "B65" ) > 0 . and . Index ( Boring_Code ( j ) , "TXL" ) > 0 ) Then
Sample_Code ( j ) = "B−65−TXL−" / / Number
Else If ( Index ( Boring_Code ( j ) , "C85" ) > 0 . and . Index ( Boring_Code ( j ) , "TXL" ) > 0 ) Then
Sample_Code ( j ) = "C−85−TXL−" / / Number
Else ! The vertical file or other files.
Sample_Code ( j ) = trim ( Boring_Code ( j ) ) / / "" / / trim ( Sample_No ( j ) )
! Write ( * , * ) Sample_Code ( j )
End If

End Do
Rewind ( File )

Read ( File , * ) ! Skips headers
Do j = 1 , nLines_data
  count = 0
  Do While ( . TRUE . )
    Read ( File , '(A)' , Advance = 'No' ) lineTemp
    ! Write ( * , '(A)' ) lineTemp
    If ( lineTemp == ', ' ) count = count + 1
    If ( count == 5 ) exit
  End Do
  Read ( File , * ) D ( j , : )
End Do
Deallocate ( Var , GefFileName , Boring_Code , NEN5104_Code , Sample_No , Depth )

End Subroutine

!---------------------------------
! Finding a value
!---------------------------------

Subroutine FindDateTimeValueApprox ( Value , Array , Index )
! Finds a given value in a given array or an approximation.
! Based on http://rosettacode.org/wiki/Search_a_list#Fortran
Implicit None
Character ( 19 ) , intent ( in ) :: Value
Character ( 19 ) :: Array ( : )
Integer :: Index
Integer :: Day_v , Month_v , Year_v , Hours_v , Minutes_v , Seconds_v
Integer :: Day_a , Month_a , Year_a , Hours_a , Minutes_a , Seconds_a
Character ( 1000 ) :: Message
Integer :: SampleSourceLog = 25
Write ( * , * )
DO Index = 1 , size ( Array )
  If ( Value == Array ( Index ) ) Then
    Write ( Message , '(A,A,17)' ) Value , " is found at index:" , Index
    Call WriteLog ( 1 , trim ( Message ) , SampleSourceLog )
End If
End DO
Else
    Read( Value , '(12.2,1x,12.2,1x,14,1x,12.2,1x,12.2,1x,12.2,1x,12.2)') Day_v, Month_v, Year_v,
    Hours_v, Minutes_v, Seconds_v
    Read( Array( Index ), '(12.2,1x,12.2,1x,14,1x,12.2,1x,12.2,1x,12.2)') Day_a, Month_a,
    Year_a, Hours_a, Minutes_a, Seconds_a
    If (Day_v == Day_a . and . Month_v == Month_a . and . Year_v == Year_a . and . Hours_v ==
        Hours_a . and . Minutes_v == Minutes_a . and . Seconds_v <= Seconds_a) Then
        Write( Message , '(A,A,A) ' ) Value , " is not exactly found but at position ",
        Index , " , " , Array( Index ), " is found."
        Call WriteLog(1, trim( Message ) , SampleSourceLog)
        Return
    End If
End If
End Do

Do
    Write( Message , '(A,A,A) ' ) "Error: ", Value , " is not found in given array," ! Outside
    the do loop or every do would print this to the screen before continuing.
    Call WriteLog(1, trim( Message ) , SampleSourceLog)
    Write( Message , '(A) ' ) "This will be due to a missing data package, switching to method 2."
    Call WriteLog(1, trim( Message ) , SampleSourceLog)
    Call FindApproxDateTime( Value , Array , Index )
End Subroutine

!----------------------------------------------------------

Subroutine FindApproxDateTime( Value , Array , Index )
! Contrary to the other routine this one is based on finding the minimum time interval
between two datetime values
Implicit None

Character( 19 ) :: Array ( )
Character( 19 ) :: Value
Integer :: Index , i , j
Integer :: TimeInt
Character( 1000 ) :: Message
Integer :: SampleSourceLog =25
Write(* , *) ! Inserts an empty line for easier reading

Do i=1,60 ! Deviation loop, maximum 1 hour and just takes the closest
    since TimeIntervalTwoValues returns the absolute difference in seconds, so found value can
    be higher or lower than initial value but it is the closest.
    Do j=1, size( Array ) ! Array loop, cycles through the array for the given deviation
        Call TimeIntervalTwoValues( TimeInt , Value , Array( j ) )
        If ( TimeInt <= ( i*60) ) Then
            Write( Message , '(A,A,A,12,A) ' ) "The closest to value ", Value , ", minimum
            deviation of " , i , " minutes."
            Call WriteLog(1, trim( Message ) , SampleSourceLog)
            Write( Message , '(A,A,A,16.6) ' ) "is found to be ", Array( j ) , " at index ", j
            Call WriteLog(1, trim( Message ) , SampleSourceLog)
            Index = j
            Return
        End If
    End Do
End Do

Write( Message , '(A,13,A) ' ) "No solution is found within a maximum deviation of ", i , " minutes."
Index = 0
Call WriteLog(1, trim( Message ) , SampleSourceLog)
End Subroutine

!----------------------------------------------------------
! Conversion Subroutines
!

Subroutine DateTimeIntegerArray(DateTime , Day , Month , Year , Hours , Minutes , Seconds)
! Converts an array of character strings of format "dd-mm-yyyy hh:mm:ss" into integers to be calculated with.
Implicit None

Character(len=19) :: DateTime (:)
Integer(2) , allocatable :: Day (:), Month (:), Hours (:), Minutes (:), Seconds (:)
Integer(4) , allocatable :: Year (:)
Allocate (Day (size (DateTime)))
Allocate (Month (size (DateTime)))
Allocate (Year (size (DateTime)))
Allocate (Hours (size (DateTime)))
Allocate (Minutes (size (DateTime)))
Allocate (Seconds (size (DateTime)))

Do i=1 , size (DateTime)
   Read (DateTime (i) , ' (I2 , 1x , I2 , 1x , I4 , 1x , I2 , 1x , I2 , 1x , I2 )' ) Day (i) , Month (i) , Year (i) , Hours (i) , Minutes (i) , Seconds (i)
End Do
End Subroutine

Subroutine DateTimeIntegerSingle(DateTimeSingle , Day , Month , Year , Hours , Minutes , Seconds)
! Converts a character string of format "dd-mm-yyyy hh:mm:ss" into integers to be calculated with.
Implicit None

Character(len=19) :: DateTimeSingle
Integer(2) :: Day , Month , Hours , Minutes , Seconds
Integer(4) :: Year
Read (DateTimeSingle , ' (I2 , 1x , I2 )' ) Day , Month
End Subroutine

Subroutine DateTimeIntegerSingleBasic(DateTimeSingle , Day , Month)
! Converts a character string of format "dd-mm-yyyy hh:mm:ss" into integers to be calculated with.
Implicit None

Character(len=19) :: DateTimeSingle
Integer(2) :: Day , Month
Read (DateTimeSingle , (I2 , 1x ) ) Day , Month
End Subroutine

Subroutine DateTimeFromSur(DateTime_Sur , Date_I , Time_R , Nlines_sur)
! Converts the Date_I en Time_R to one vector, which can be searched and has the same format as the one read from Prodata.
! The conversion from excel no. to dd/mm/yyyy is partially based on http://www.codeproject.com/Articles/2750/Excel-serial-date-to-Day-Month-Year-and-vice-versa
Implicit None

Character(19) , Allocatable :: DateTime_Sur (:)
Integer(2) , Allocatable :: Day_L (:), Month_L (:), Hours_L (:), Minutes_L (:), Seconds_L (:)
Integer(4) , Allocatable :: Year_L (:)
Integer :: i,j,Nlines_sur
Integer :: Date_I (:)

Allocate (Day_L (size (DateTime_Sur)))
Allocate (Month_L (size (DateTime_Sur)))
Allocate (Year_L (size (DateTime_Sur)))
Allocate (Hours_L (size (DateTime_Sur)))
Allocate (Minutes_L (size (DateTime_Sur)))
Allocate (Seconds_L (size (DateTime_Sur)))
Allocate (Day (size (DateTime_Sur)))
Allocate (Month (size (DateTime_Sur)))
Allocate (Year (size (DateTime_Sur)))
Allocate (Hours (size (DateTime_Sur)))
Allocate (Minutes (size (DateTime_Sur)))
Allocate (Seconds (size (DateTime_Sur)))
Allocate (DateTime_Sur (size (DateTime_Sur)))

Do i=1 , size (DateTime_Sur)
   Read (DateTime_Sur (i) , ' (I2 , 1x , I2 , 1x , I4 , 1x , I2 , 1x , I2 , 1x , I2 )' ) Day (i) , Month (i) , Year (i) , Hours (i) , Minutes (i) , Seconds (i)
End Do
End Subroutine
Real :: Time_R(:)

Real :: l,n,z,y

Allocate(Day_L(Nlines_sur))
Allocate(Month_L(Nlines_sur))
Allocate(Year_L(Nlines_sur))
Allocate(Hours_L(Nlines_sur))
Allocate(Minutes_L(Nlines_sur))
Allocate(Seconds_L(Nlines_sur))
Allocate(DateTime_Sur(Nlines_sur))

Do i=1,Nlines_Sur

If (Date_I(i) == 60) Then
    Day_L = 29
    Month_L = 2
    Year_L = 1900
Else
    If (Date_I(i) < 60) Then
        Date_I(i) = Date_I(i)+1
    End If
End If

l = Date_I(i) + 68569 + 2415019

n = floor(( 4 * l ) / 146097)

l = l − floor(( 146097 * n + 3 ) / 4)
z = floor(( 4000 * ( l + 1 ) ) / 1461001)
l = l − floor(( 1461 * z ) / 4) + 31
y = floor(( 80 * l ) / 2447)

Day_L = l - floor( ( 2447 * y ) / 80)

Month_L = y + 2 - ( 12 * l )
Year_L = 100 * ( n - 49 ) + z + 1

End If

Hours_L(i) = int(mod(Time_R(i)*24.0,60.0))
Minutes_L(i) = int(mod(Time_R(i)*1440.0,60.0))
Seconds_L(i) = nint(mod(Time_R(i)*86400.0,60.0)) ! Due to possible errors when taking the only the integer part for seconds, the seconds are rounded towards the nearest integer.

If (Seconds_L(i) == 60) Then ! Additional security measure, to assure realistic values.
    Minutes_L(i) = Minutes_L(i) + 1
    Seconds_L(i) = 0
End If

Write(DateTime_Sur(i), '(',H12.2,A1,12.2,A1,14.1x,A1,12.2,A1,12.2,A1,12.2)') Day_L(i)," ",Month_L(i)," ",Year_L(i)," ",Hours_L(i)," ",Minutes_L(i)," ",Seconds_L(i)

End Do

End Subroutine

Subroutine SecToTimeR(TimeInS, HH, MM, SS)
! Calculates the time in seconds to HH:MM:SS
Implicit None

Real :: TimeInS
Integer :: HH,MM,SS

HH = int(TimeInS/3600)
MM = int((TimeInS−HH*3600)/60)
SS = int(TimeInS−HH*3600−MM*60)
Subroutine SecToTimeI (TimeInS, HH, MM, SS)
! Calculates the time in seconds to HH:MM:SS
Implicit None

Integer :: TimeInS
Integer :: HH, MM, SS

HH = int (TimeInS/3600)
MM = int ((TimeInS–HH*3600)/60)
SS = int (TimeInS–HH*3600–MM*60)

End Subroutine

Subroutine ZoomGPDays (DateTimeSource, DateTimeMin, DateTimeMax, X)
! Calculates the time min and max based on interval of X days
Implicit None

Character (19) :: DateTimeSource, DateTimeMin, DateTimeMax
Integer :: X
Integer (2) :: Day, Day_Min, Day_Max
Integer (2) :: Month, Month_Min, Month_Max
Integer (4) :: Year
Integer (2) :: Hours, Minutes, Seconds

Read (DateTimeSource, ' (I2, 1 x, I2, 1 x, I4, 1 x, I2, 1 x, I2, 1 x) ' ) Day, Month, Year, Hours, Minutes, Seconds

Day Min = Day – X
Month Min = Month

! Day_Min can become 0 or less so than the month has to be reduced and day min becomes the last or the x to last day of that month.
If (Day Min < 1) Then
  If ((Month – 1) == 1 . or. (Month – 1) == 3 . or. (Month – 1) == 5 . or. (Month – 1) == 7 . or. (Month – 1) == 8 . or. (Month – 1) == 10 . or. (Month – 1) == 12) Then
    Day Min = 31 – Day Min
    Month Min = Month – 1
  Else If ((Month – 1) == 2) Then
    Day Min = 28 – Day Min
    Month Min = Month – 1
  End If
End If

Write (DateTimeMin, ' (I2, 2, A1, I2, 2, A1, I4, 1 x, I2, 2, A1, I2, 2, A1, I2, 2) ' ) Day_Min, “-”, Month_Min, “-”, Year, Hours, “:”, Minutes, “:”, Seconds
Write (*, *) DateTimeMin

Day Max = Day + X
Month Max = Month

If (Day Max > 31) Then
  If ((Month) == 1 . or. (Month) == 3 . or. (Month) == 5 . or. (Month) == 7 . or. (Month) == 8 . or. (Month) == 10 . or. (Month) == 12) Then
    Day Max = Day Max – 31
    Month Max = Month + 1
  Else If ((Month) == 4 . or. (Month) == 6 . or. (Month) == 9 . or. (Month) == 11) Then
    Day Max = Day Max – 30
    Month Max = Month + 1
  Else If ((Month) == 2) Then
    Day Max = Day Max – 28
  End If
End If
Month_Max = Month + 1
End If
Else If (Day_Max > 30) Then
    If ((Month) == 4 . or. (Month) == 6 . or. (Month) == 9 . or. (Month) == 11) Then
        Day_Max = Day_Max - 30
        Month_Max = Month + 1
    ElseIf ((Month) == 2) Then
        Day_Max = Day_Max - 28
        Month_Max = Month + 1
    End If
Else If (Day_Max > 28) Then
    If (Month == 2) Then
        Day_Max = Day_Max - 28
        Month_Max = Month + 1
    End If
End If
Write (DateTimeMax, ' (I2,1x,I2,1x,I4,1x,I2,1x,I2,1x,I2) ' ) Day_Max,"-",Month_Max,"-",Year,Hours,"-",Minutes,"-",Seconds
Write(*,*) DateTimeMax
End Subroutine

Subroutine ZoomGPHours (DateTimeSource ,DateTimeMin ,DateTimeMax ,Y)
! Calculates the time min and max based on interval of Y hours
Implicit None
Character (19) :: DateTimeSource ,DateTimeMin ,DateTimeMax
Integer :: Y
Integer (2) :: Hours , Hours_Min , Hours_Max
Integer (2) :: Day , Day_Min , Day_Max
Integer (2) :: Month , Month_Min , Month_Max
Integer (4) :: Year
Integer (2) :: Minutes , Seconds
Read (DateTimeSource ,'(I2,i1x,i2,i4,i1x,i2,i1x,i2) ') Day ,Month ,Year ,Hours ,Minutes , Seconds
Hours_Min = Hours - Y
Day_Min = Day
Month_Min = Month
If (Hours_Min < 0) Then
    Hours_Min = 24 + Hours_min
End If
! Day_Min can become 0 or less so that the month has to be reduced and day_min becomes the last of the x to last day of that month.
If (Day_Min < 1) Then
    If ((Month-1) == 1 . or. (Month-1) == 3 . or. (Month-1) == 5 . or. (Month-1) == 7 . or. (Month-1) == 8 . or. (Month-1) == 10 . or. (Month-1) == 12) Then
        Day_Min = 31 + Day_Min
        Month_Min = Month - 1
    Else If ((Month-1) == 4 . or. (Month-1) == 6 . or. (Month-1) == 9 . or. (Month-1) == 11) Then
        Day_Min = 30 + Day_Min
        Month_Min = Month - 1
    Else If ((Month-1) == 2) Then
        Day_Min = 28 + Day_Min
        Month_Min = Month - 1
    End If
End If
Write (DateTimeMin, '(2(i2,i1x,i1x,i2,i1x,i2,i1x,i2) ') Day_Min,"-",Month_Min,"-",Year ,Hours_Min,"-",Minutes,"-",Seconds
Hours_Max = Hours + Y
Day_Max = Day
Month\_Max = Month

If (Hours\_Max > 23) Then  
! 23:59 becomes 00:00 the next day  
    Hours\_Max = Hours\_Max - 24  
    Day\_Max = Day\_Max + 1  
End If

If (Day\_Max > 31) Then
    If ((Month) == 1 . or. (Month) == 3 . or. (Month) == 5 . or. (Month) == 7 . or. (Month) == 8 . or. (Month) == 10 . or. (Month) == 12) Then
        Day\_Max = Day\_Max - 31
        Month\_Max = Month + 1
    Else If ((Month) == 4 . or. (Month) == 6 . or. (Month) == 9 . or. (Month) == 11) Then
        Day\_Max = Day\_Max - 30
        Month\_Max = Month + 1
    Else If ((Month) == 2) Then
        Day\_Max = Day\_Max - 28
        Month\_Max = Month + 1
    End If
Else If (Day\_Max > 30) Then
    If ((Month) == 4 . or. (Month) == 6 . or. (Month) == 9 . or. (Month) == 11) Then
        Day\_Max = Day\_Max - 30
        Month\_Max = Month + 1
    Else If ((Month) == 2) Then
        Day\_Max = Day\_Max - 28
        Month\_Max = Month + 1
    End If
Else If (Day\_Max > 28) Then
    If (Month == 2) Then
        Day\_Max = Day\_Max - 28
        Month\_Max = Month + 1
    End If
End If

Write (DateTimeMax, ’(2(12.2,A1),14.1x,2(12.2,A1),12.2) ‘)  
Day\_Max, ”-“ , Month\_Max, ”-“ , Year ,  
Hours\_Max, ”-“ , Minutes , ”-“ , Seconds

End Subroutine

!--------------------------------------------------------------

Subroutine TimeInterval\_PD(Time\_Int , DateTime , Index\_PD)
! Calculates the time intervals between the data points. Starting from top of DateTime vector.
! Change to only around point N if calculation time becomes unfeasible.
Implicit None

Integer , allocatable :: Time\_Int (: )  
Character(len=19) :: DateTime (: )
Integer(2) , allocatable :: Day (: ) , Month (: ) , Hours (: ) , Minutes (: ) , Seconds (: )
Integer(4) , allocatable :: Year (: )
Integer :: i , j
Integer :: Index\_PD
Integer :: DaysMonth

Call DateTimeIntegerArray (DateTime , Day , Month , Year , Hours , Minutes , Seconds)
Allocate ( Time\_Int (Index\_PD))

Time\_Int(1) = 0  
! Trick to enable forward calculation of the time intervals

Do i=1,(Index\_PD-1)
    j = Index\_PD+i-1
    If (Month(j) == 1 . or. Month(j) == 3 . or. Month(j) == 5 . or. Month(j) == 7 . or. Month(j) == 8 . or. Month(j) == 10 . or. Month(j) == 12) Then
        DaysMonth = 31
    Else If (Month(j) == 2) Then
        DaysMonth = 28  
        ! TO BE DONE: take leap years into account.
    Else If (Month(j) == 4 . or. Month(j) == 6 . or. Month(j) == 9 . or. Month(j) == 11) Then
        DaysMonth = 30
    End If
End Do
End If

TimeInt(j) = ((Month(j)−Month(j−1))∗(DaysMonth∗24∗60∗60)) + ((Day(j)−Day(j−1))∗(24∗60∗60)) + ((Hours(j)−Hours(j−1))∗(60∗60)) + ((Minutes(j)−Minutes(j−1))∗(60)) + ((Seconds(j)−Seconds(j−1))∗1)

If (TimeInt(j) >= (12∗60∗60)) Then ! Security measure for the fact that the drilling was in 12 hours shifts most of the time. An interval of 12+ hours will cause deviations further on in the program.

TimeInt(j) = TimeInt(j) − (12∗60∗60)
End If

If (TimeInt(j) >= (10∗60)) Then ! Security measure for the fact that sometimes data in packages of 10 min. is missing.

TimeInt(j) = (5∗60) ! Not entirely true, can cause a slight deviation in the travel time and thus in the source location.

End If ! It now takes 5 minutes instead of 10 minutes of whatever it was doing (full pumping – no pumping) at the moment the interval STOPS, since the Pumped volume is back calculated.

! For two missing data packages in a row (08-09-2015 08:14 - 08:34) the deviation is of course larger.

End Do

End Subroutine

!------------------------------------------------------

Subroutine TimeIntervalTwoValues(TimeInt, DateTime1, DateTime2, Option)
! Calculates the time intervals between two datetime values in [s], independent of which is larger.
Implicit None

Integer : : TimeInt ! [s]
Character (19) :: DateTime1, DateTime2
Integer (2) :: Day1, Month1, Hours1, Minutes1, Seconds1
Integer (4) :: Year1
Integer (2) :: Day2, Month2, Hours2, Minutes2, Seconds2
Integer (4) :: Year2
Integer, Optional :: Option

Call DateTimeIntegerSingle(DateTime1, Day1, Month1, Year1, Hours1, Minutes1, Seconds1)
Call DateTimeIntegerSingle(DateTime2, Day2, Month2, Year2, Hours2, Minutes2, Seconds2)

If (Month1 == 1 . or. Month1 == 3 . or. Month1 == 5 . or. Month1 == 7 . or. Month1 == 8 . or. Month1 == 10 . or. Month1 == 12) Then ! Jan, Mar, May, Jul, Aug, Oct, Dec
C2 = 31∗24∗60∗60
Else if (Month1 == 2) Then ! Feb
C2 = 28∗24∗60∗60
Else
C2 = 30∗24∗60∗60
End If

C3 = 24∗60∗60 ! Hours in a day are constant, 24
C4 = 60∗60 ! Minutes in a hour are constant, 60
C5 = 60 ! Seconds in a minute are constant, 60

If (Present(Option)) Then
TimeInt = (((Month1−Month2)∗C2)+(Day1−Day2)∗C3)+(Hours1−Hours2)∗C4)+((Minutes1−Minutes2)+C5)+(Seconds1−Seconds2)
Else
TimeInt = abs(((Month1−Month2)∗C2)+(Day1−Day2)∗C3)+(Hours1−Hours2)∗C4)+((Minutes1−Minutes2)+C5)+(Seconds1−Seconds2)
End If

End Subroutine

!------------------------------------------------------

Subroutine GeometryBoreH(PI, D_bh, D_ds, A_bh, A_ds, A_annulus)
! Calculates the geometry
Implicit None
Real ( 8 ) : : PI
Real :: D_bh , D_ds
Real :: A_bh , A_ds , A_annulus

A_bh = PI / 4 * D_bh ** 2
A_ds = PI / 4 * D_ds ** 2
A_annulus = A_bh - A_ds

End Subroutine

! Calculations – Preparations
!
! Fluid Parameters
!
! Herschel–Bulkey

Subroutine Herschel_Bulkey (X, Y, Tau_Yield , K, n , Min_Alpha , Min_Delta , Sample_Code , MainI ,
Conversion_Degr_Pa , RPM_600 , RPM_300 , RPM_200 , RPM_100 , RPM_60 , RPM_30 , RPM_6 , RPM_3 , GammaY, Tau ,
Tau_Calc_All , Alpha_All , Delta_All )
!
Based on http://www.eng.umd.edu/~nsw/chbe250/slope.htm, and is used to replace the
intercept and slope function as found in excel.
Implicit None

Integer :: RPM_600(:), RPM_300(:), RPM_200(:), RPM_100(:), RPM_60(:), RPM_30(:), RPM_6(:), RPM_3(:)

Real :: Conversion_Degr_Pa

Real :: MainI

Real (8) :: X(:,), Y(:,)

Real :: K, n

Real , Dimension (20) :: K_All , n_All

Real :: A

Real :: SumX, SumY

Real :: SumXX, SumXY

Real :: X_Average, Y_Average

Integer :: i, j

Integer :: Index

Real :: Tau_Calc

Real (8) , Dimension (20) :: Tau_Calc_All

Real :: Min_Alpha

Real :: Alpha ! Scaling factor on tau_yield to determine the
minimum delta.

Real (8) , Dimension (20) :: Alpha_All

Real :: Min_Delta

Real :: Delta ! Deviation between the measured and calculated
values, sum((tau-tau_calc)^2).

Real (8) , Dimension (20) :: Delta_All

Real (8) , Dimension (20) :: Tau

Character(*) :: Sample_Code

Real (8) , dimension (8) :: GammaY

Real :: Tau_Yield

Real , Dimension (20) :: Tau_Yield_All

Tau(1) = (RPM_3(MainI) * Conversion_Degr_Pa)

Tau(2) = (RPM_6(MainI) * Conversion_Degr_Pa)

Tau(3) = (RPM_30(MainI) * Conversion_Degr_Pa)

Tau(4) = (RPM_60(MainI) * Conversion_Degr_Pa)

Tau(5) = (RPM_100(MainI) * Conversion_Degr_Pa)

Tau(6) = (RPM_200(MainI) * Conversion_Degr_Pa)

Tau(7) = (RPM_300(MainI) * Conversion_Degr_Pa)

Tau(8) = (RPM_600(MainI) * Conversion_Degr_Pa)

Delta = 0.d0

Do j = 1,20 ! Taking tau_yield = 1*tau_yield distorts the results

Alpha = j*0.05d0

Tau_yield = Alpha * RPM_3(MainI) * Conversion_Degr_Pa

Do i = 1, size (Y)

If ((Tau(i) - Tau_yield) == 0) Then

Y(i) = 0d0 ! TO BE DONE, decide whether this or not taking it into

account is more accurate...

Else
    Y(i) = Log10(Tau(i) - Tau_yield)
End If

End Do

SumX = 0.00
SumY = 0.00

Do i=1,Size(X)
    SumX = SumX + X(i)
    SumY = SumY + Y(i)
    SumXX = SumXX + X(i)*X(i)
    SumXY = SumXY + X(i)*Y(i)
End Do

X_Average = SumX / float(size(X))
Y_Average = SumY / float(size(X))

n = (SumXY - SumX*Y_Average) / (SumXX - SumX*X_Average)
A = Y_Average - n*X_Average
K = 10**A

Delta = 0 ! Resets the value from the previous loop, w/o it the delta would only increase
Do i=1,Size(GammaY)
    Tau_Calc = Tau_Yield + K*GammaY(i)**n
    Delta = Delta + (Tau(i) - Tau_Calc)**2.00
End Do

K_All(j) = K
n_All(j) = n
Tau_Yield_All(j) = Tau_Yield
Delta_All(j) = Delta
Alpha_All(j) = Alpha
End Do

Index = Minloc(Delta_All,1)
Min_Delta = Delta_All(Index)
Min_Alpha = Index*0.05
K = K_All(Index)
n = n_All(Index)
Tau_Yield = Tau_Yield_All(Index)

Do i = 1,Size(GammaY)
    Tau_Calc_All(i) = Tau_Yield_All(Index) + K_All(Index)*GammaY(i)**n_All(Index)
End Do

! Call f2gp6(Sample_Code,GammaY,Tau,'3a',GammaY,Tau_Calc_BH,'3b',1,'Shear Rate [-]','Reading [Pa]','Measurements vs. Calculated','Measurements','Calculated')
! Call f2gp5(Sample_Code,Alpha_All,Delt_at_All,'2a',1,'Alpha [-]','Delta [Pa]','Alpha vs. Delta','Delta')

End Subroutine

!------------------------------------------------------
! Bingham
!------------------------------------------------------

Subroutine Bingham(MainI,PV,YP,RPM_600,RPM_300,Conversion_Degr_lb,Conversion_lb_Pa,
Conversion_RPM_PerS,GammaY,Tau_Calc_BH,Tau,Delta)
! Calculates Bingham parameters
! PV and YP are both in [Pa], YP is used as Tau_yield
Implicit None

Integer :: RPM_600(::),RPM_300(::)
Integer :: MainI
Real :: PV,YP
Real :: Conversion_Degri_lb, Conversion_lb_Pa, Conversion_RPM_PerS
Real(8),dimension(8) :: GammaY
Real(8), Dimension(20) :: Tau_Calc_BH
Real :: Delta  ! Deviation between the measured and calculated values, sum((tau-tau_calc)^2).
Real(8), Dimension(20) :: Tau
Integer :: i

PV = ((RPM_600(MainI)-RPM_300(MainI))\*Conversion_Degr_lb\_Conversion_lb_Pa)/(600*Conversion\_RPM\_PerS - 300*Conversion\_RPM\_PerS)
YP = (RPM_300(MainI) - (RPM_600(MainI) - RPM_300(MainI))\*Conversion_Degr_lb\_Conversion_lb_Pa

Delta = 0
Do i = 1, size(GammaY)
  Tau_Calc_BH(i) = YP + PV * GammaY(i)
  Delta = Delta + (Tau(i) - Tau_Calc_BH(i))**2.
d0
End Do

End Subroutine

! Calculations - Main

Subroutine TransportLength_PD(PumpVol, Index_PD, A_bh, TranspLen, TranspLenSum, TimeInt_PD, MudFlw, Loss)

! Calculates the amount of that is actually pumped, flowrate*time interval
! Volume drilled soil can be considered as equal to the amount of drilling fluid necessary
to fill that hole, this means that loss in volume occurs when considering the whole
system not when considering the hole alone.
! Since the drilled soil is recycled and thus removed from the fluid, ground water however
remains in the fluid, so excluding the filtration loss less of a loss than the volume of...
! ...the drilled soil although additional Teggel or OCMA is required to maintain the
properties of the drilling fluid. Some fluid and bentonite will also be lost due to the
recycling unit.
! Merged with the Transport Length subroutine so this one is REDUNDANT.
! Calculates the transported distance based on the pumped volume and cross-section

Implicit None

Integer :: i, j, Index_PD
Real :: A_bh ! [m^2]
Real, Allocatable :: TranspLen(:) ! [m]
Real(8), Allocatable :: TranspLenSum(:) ! [m]
Integer :: TimeInt_PD(:) ! [s]
Integer :: MudFlw(:) ! [L/min]
Real, Allocatable :: PumpVol(:) ! [L]
Real, Optional :: Loss
Real :: Loss_P ! Loss percentage

Allocate (PumpVol(Index_PD))
Allocate (TranspLen(Index_PD))
Allocate (TranspLenSum(Index_PD))

If (Present(Loss)) Then
  Loss_P = Loss
Else
  Loss_P = 0.3.d0
End If

Do i=1,Index_PD
  If (MudFlw(i) < 0 .or. MudFlw(i) > 2500) Then
    MudFlw(i) = 0
  End If
  PumpVol(i) = (MudFlw(i)/60)*TimeInt_PD(i) ! Mudflow is given per minute so it
  PumpVol(i) = PumpVol(i)*1.d0 - Loss_P ! Correction for the infiltration loss.
End Do

Do i = 1, Index_PD
  TranspLen(i) = (PumpVol(i)/1000.d0)/(A_bh)
End Do
TranspLenSum(Index_PD) = TranspLen(Index_PD)
Do i = 1, (Index_PD-1)
    j = Index_PD - i
    TranspLenSum(j) = TranspLenSum(j+1) + TranspLen(j)
End Do
End Subroutine

!------------------------------------------
Subroutine FluidVelocity(AverageV, MaxV, Index_PD, Index_Sol_PD, MudFlw, A_bh, Loss)
! Calculates the fluid velocity
Implicit None

Real, Allocatable :: V(:) ! [m/s]
Integer :: CountN0Arr
Real :: AverageV, MaxV
Integer :: MudFlw(:) ! [L/min]
Real :: A_bh ! [m^2]
Real :: Loss
Integer :: i, Index_Sol_PD, Index_PD

Allocate(V(Index_PD-Index_Sol_PD+1))
CountN0Arr = 0
Do i = Index_Sol_PD, Index_PD
    V(i+1-Index_Sol_PD) = (MudFlw(i)/1000.d0/60.d0)*(1.d0 - Loss))/A_bh
    If (V(i+1-Index_Sol_PD) > 0) CountN0Arr = CountN0Arr + 1
End Do

AverageV = Sum(V)/CountN0Arr
MaxV = Maxval(V)
Write(*,*) ’(2(A, I))’ "Index_Sol_PD, Index_PD: ", Index_Sol_PD, ", Index_PD
Write(*,*)
End Subroutine

!------------------------------------------
Subroutine Intersection(DateTime_PD, Index_PD, DateTime_Sur, Index_Sur, TranspLenSum,MD, Index_Sol_PD, Index_Sol_Sur, SourceLoc_Sur, SourceLoc_PD, SourceLocAv, DateTimeSample)
! Finds the values from the ProData data and derived data at the times of the Survey file.
Implicit None

Integer :: h, i, j, k, l, z, Index_PD, Index_Sur
Real(8) :: MD(:)
Real(8) :: TranspLenSum(:)
Character(19) :: DateTime_PD(:)
Character(19) :: DateTime_Sur(:)
Integer :: Index_Sol_PD, Index_Sol_Sur
Real :: Deviation
Real :: SourceLoc_PD, SourceLoc_Sur
Real :: SourceLoc
Character(19) :: DateTimeSample
Real :: SourceLocAV, SourceLocAV1, SourceLocAV2
Integer(2), Allocatable :: Day_Sur(:), Month_Sur(:), Hours_Sur(:), Minutes_Sur(:), Seconds_Sur(:)
Integer(4), Allocatable :: Year_Sur(:)
Integer(2), Allocatable :: Day_PD(:), Month_PD(:), Hours_PD(:), Minutes_PD(:), Seconds_PD(:)
Integer(4), Allocatable :: Year_PD(:)
Integer :: HH1, MM1, SS1, HH2, MM2, SS2
Call DateTimeIntegerArray(DateTime_Sur, Day_Sur, Month_Sur, Year_Sur, Hours_Sur, Minutes_Sur, Seconds_Sur)
Call DateTimeIntegerArray(DateTime_PD, Day_PD, Month_PD, Year_PD, Hours_PD, Minutes_PD, Seconds_PD)
Approach 1 – Making a list of Index numbers connecting the values from the Sur file to certain values from the PD Data and derived values.

Allocate(IndexSurPD(Index_Sur))

Do i = 1, Index_Sur
    If (Day_Sur(i) == Day_PD(j) .and. Month_Sur(i) == Month_PD(j) .and. Year_Sur(i) == Year_PD(j) .and. Hours_Sur(i) == Hours_PD(j) .and. Minutes_Sur(i) == Minutes_PD(j)) Then
        ! IndexSurPD(i) = j  ! Results in IndexSurPD = 0, all entries
    ! or
        ! IndexSurPD(z) = j  ! Results in IndexSurPD = -116300599, all entries.
        ! IndexSurPD(z) = z for some reason gives the same result...
        z = z + 1
        Write(*,*) IndexSurPD(z), i, j, Day_Sur(i), Day_PD(j), Minutes_Sur(i), Minutes_PD(j)
    End If
End Do

Approach 3 – (Approach 2 was fairly similar but based on interpolation to refine the source location. Approach 3 takes the average of the 2 solutions found. See comments.)

Write(*,*)
! Inserts an empty line for easier reading
Deviation = 0.1 d0
Do h = 1, 999
    ! Deviation loop, if exact match is found solution it won’t be used
    Do i = 1, Index_Sur
        ! Survey based loop, changes the survey time and date to the next value in the given survey time.
        k = Index_Sur + 1 - i
        Do j = 1, Index_PD
            If (Day_Sur(k) == Day_PD(j) .and. Month_Sur(k) == Month_PD(j) .and. Year_Sur(k) == Year_PD(j) .and. Hours_Sur(k) == Hours_PD(j) .and. Minutes_Sur(k) == Minutes_PD(j)) Then
                If (MD(k) == TranspLenSum(j)) Then
                    Write(*,'(A,i,A,i)') "Solution found at:" , k, " & ", j
                    Index_Sol_Sur = k
                    Index_Sol_PD = j
                    Write(*,'(A,i,F8.2)') "Index_Sol_Sur = ", Index_Sol_Sur , MD(k)
                    Write(*,'(A,i,F8.2)') "Index_Sol_PD = ", Index_Sol_PD , TranspLenSum(j)
                Return
            Else If (MD(k) <= (TranspLenSum(j) * (1 + Deviation / 100)) .and. MD(k) >= (TranspLenSum(j) * (1 - Deviation / 100))) Then
                Write(*,'(A,i,F17.2)') "Approximate solution found at:" , i , " & ", j
                Index_Sol_Sur = k
                Index_Sol_PD = j
                SourceLoc_Sur = MD(k)
                SourceLoc_PD = TranspLenSum(j)
                Write(*,'(A,i,F17.2,A,A,F7.2)') "Index_Sol_Sur, DateTime, MD = ", Index_Sol_Sur , " , DateTime_Sur(k), " , SourceLoc_Sur
                Write(*,'(A,i,F17.2,A,A,F7.2)') "Index_Sol_PD, DateTime, TranspLenSum = ", Index_Sol_PD , " , DateTime_PD(j), " , SourceLoc_PD
                Write(*,'(A,F5.1,A)') "Minimum Deviation = ", Deviation , "%"
            End If
        End Do
    End Do
End If

If (Deviation <= 1) Then
SourceLoc = MD(k)

Else If (Deviation > 1) Then    ! Pick one of the two methods.

    SourceLocAv1 = Abs((MD(k)−TranspLenSum(j))/2)+Min(MD(k),
                      SourceLocAv2 = (MD(k) + MD(k+1))/2

End If

Write(*, '(A,F8.3)') "SourceLocAv1 = ", SourceLocAv1
Write(*, '(A,F8.3)') "SourceLocAv2 = ", SourceLocAv2

End If

Write(*, ' (A,F7.2) ') "A more accurate source location [m]: ", SourceLocAv
Write(*, ' (A) ') "Which is the average of the two."

Return

End If

End Do

End Do

Deviation = Deviation + 0.1d0    ! Increases the allowed Deviation value by 0.1, thus
                                ! allowing a less accurate value to be found if no solution has been found. Maximum
                                ! deviation of an exact match of 0.1+999*0.1 = 100.0%.

End Do

Write(*, ' (A,F5.1,A) ') "A solution is not found with given maximum deviation , ", Deviation,"%"
                                ! If nothing is found for maximum Deviation , it displays that information

Index_Sol_PD = 1
Index_Sol_Sur = 1

End Subroutine

!------------------------------------------------------------------------

Subroutine SamplingLocation(Index, ElevationCalc, Away, R_Calc, SamplingDepth, SamplingAway,
                                SamplingRight)
                                ! Calculates the location at sampling time .
Implicit None

Real :: ElevationCalc (:), R_Calc (:)
Real(8) :: Away (:)
Integer :: Index
Real :: SamplingDepth, SamplingAway, SamplingRight

SamplingDepth = ElevationCalc(Index)
SamplingAway = Away(Index)
SamplingRight = R_Calc(Index)

End Subroutine

!------------------------------------------------------------------------

Subroutine SourceLocation(Index, SourceLocAv, MD, ElevationCalc, Away, R_Calc, SourceDepth,
                                SourceAway, SourceRight)
                                ! Calculates the source location based on the intersection found in the intersection
                                ! subroutine and by interpolation within the survey data .
Implicit None

Real :: ElevationCalc (:), R_Calc (:)
Real(8) :: Away (:), MD(:)
Integer :: Index
Real :: SourceDepth, SourceAway, SourceRight
Real :: SourceLocAv

Real :: c1, c2

If (SourceLocAv < MD(Index)) Then

    Write(*, '*')
    Write(*, '(A) ') "SourceLocAv < MD"

    c1 = abs(SourceLocAv − MD(Index)) / abs(MD(Index) − MD(Index −1))
    c2 = abs(SourceLocAv − MD(Index −1)) / abs(MD(Index) − MD(Index −1))

    Write(*, '(A,F8.3,F8.3) ') "c1 and c2: ", c1, c2
Write(*,'(A, F8.3, A, F8.3)') "ElevationCalc(Index) & ElevationCalc(Index-1) :
  "ElevationCalc(Index), " ElevationCalc(Index-1)
SourceDepth = c1 * ElevationCalc(Index-1) + c2 * ElevationCalc(Index)
Write(*,'(A, F8.3, A, F8.3)') "Away(Index) & Away(Index-1):
  "Away(Index), " Away(Index-1)
SourceAway = c1 * Away(Index-1) + c2 * Away(Index)
Write(*,'(A, F8.3, A, F8.3)') "R_Calc(Index) & R_Calc(Index-1):
  "R_Calc(Index), " R_Calc(Index-1)
SourceRight = c1 * R_Calc(Index-1) + c2 * R_Calc(Index)
Write(*,'(A, F8.3, A, F8.3, A, F8.3)') "SourceDepth, SourceAway, SourceRight:
  "SourceDepth, " SourceAway, " SourceRight
Write(*,*)
Else If (SourceLocAv > MD(Index)) Then
  Write(*,* ) "SourceLocAv > MD"
c1 = abs(SourceLocAv - MD(Index)) / abs(MD(Index) - MD(Index+1))
c2 = abs(SourceLocAv - MD(Index+1)) / abs(MD(Index) - MD(Index+1))
Write(*,'(A,F8.3,F8.3)') "c1 and c2: ", c1, c2
Write(*,'(A, F8.3, A, F8.3)') "ElevationCalc(Index) & ElevationCalc(Index-1) :
  "ElevationCalc(Index), " ElevationCalc(Index-1)
Write(*,'(A, F8.3, A, F8.3)') "Away(Index) & Away(Index-1):
  "Away(Index), " Away(Index-1)
SourceAway = c1 * Away(Index-1) + c2 * Away(Index)
Write(*,'(A, F8.3, A, F8.3)') "R_Calc(Index) & R_Calc(Index-1):
  "R_Calc(Index), " R_Calc(Index-1)
SourceRight = c1 * R_Calc(Index-1) + c2 * R_Calc(Index)
Write(*,'(A, F8.3, A, F8.3, A, F8.3)') "SourceDepth, SourceAway, SourceRight:
  "SourceDepth, " SourceAway, " SourceRight
Write(*,*)
Else If (SourceLocAv == MD(Index)) Then
  Write(*,*) "SourceLocAv == MD"
SourceDepth = ElevationCalc(Index)
SourceAway = Away(Index)
SourceRight = R_Calc(Index)
Write(*,'(A, F8.3, F8.3, F8.3)') "SourceDepth, SourceAway, SourceRight:
  "SourceDepth, " SourceAway, " SourceRight
Write(*,*)
End If
End Subroutine

-------------------
Subroutine TransportTimes(DateTime_PD, DateTime_Sur, Index_Sol_PD, Index_Sol_Sur, DateTimeSample,
  TimeInt1, TimeInt2, TimeIntAv)
  ! Calculates transport time but also pumptimes during the transport interval, which can be
  ! used as the drill time during the transport interval.
  ! Made separate to clean the intersection subroutine.
Implicit None
Character(19) :: DateTimeSample
Integer :: TimeInt1, TimeInt2
Integer :: TimeIntAv
Character(19) :: DateTime_PD(:)
Character(19) :: DateTime_Sur(:)
Integer :: Index_Sol_PD, Index_Sol_Sur
Integer :: HH1, MM1, SS1, HH2, MM2, SS2, HH3, MM3, SS3
Call TimeIntervalTwoValues(TimeInt1, DateTime_Sur(Index_Sol_Sur), DateTimeSample)
Call TimeIntervalTwoValues(TimeInt2, DateTime_PD(Index_Sol_PD), DateTimeSample)

If (TimeInt1 >= (12*60*60)) Then ! Stability measure, accounting for the stop during the night.
    TimeInt1 = TimeInt1 – (12*60*60)
End If
If (TimeInt2 >= (12*60*60)) Then
    TimeInt2 = TimeInt2 – (12*60*60)
End If
Call SecToTimeI(TimeInt1, HH1, MM1, SS1)
Call SecToTimeI(TimeInt2, HH2, MM2, SS2)
TimeIntAv = ((TimeInt1+TimeInt2)*0.5)
Call SecToTimeI(TimeIntAv, HH3, MM3, SS3)
End Subroutine TransportTimes

!-------------------------------------------------------

Subroutine PumpTimes(MudFlw, TimeInt_PD, Index_Sol_PD, Index_PD, PumpTime)
! Calculates pump time during the transport interval, which can be used as the drill time during the transport interval.
! Made separate to clean the intersection subroutine.
Implicit None

Integer :: i, Index_Sol_PD, Index_PD
Integer :: TimeInt_PD(:) ! [s]
Integer :: MudFlw(:) ! [L/min]
Integer :: PumpTime
Integer :: HH1, MM1, SS1, HH2, MM2, SS2
PumpTime = 0
Do i=Index_Sol_PD+1, Index_PD
    If (MudFlw(i) > 0) Then
        PumpTime = PumpTime + TimeInt_PD(i)
    End If
End Do
Call SecToTimeI(PumpTime, HH1, MM1, SS1)
End Subroutine PumpTimes

!-------------------------------------------------------

Subroutine DmaxTransport(Y_g, Tau_Yield, Rho_Inflow, Rho_Grain, g, D90, FluidType)
! Calculates the Transport parameter as found in Denekamp, 2000. 0.1< is a problem and ideal is >2 or 4.
Implicit None
Real :: Y_g
Real :: Tau_Yield
Real :: Rho_Inflow, Rho_Grain
Real :: D90
Real :: g
Character(2) :: FluidType
C h a r a c t e r ( 1 0 2 4 ) :: Message
I n t e g e r :: SampleLog=25

D90 = (3*Tau_Yield) / ( (Rho_Grain−Rho_Inflow) *g*Y_g )

W r i t e (Message, ' (A,F,A,F3.1,A) ' ) “The max diameter for a "// Fluid Type //" fluid is ",D90," mm while Yg is ",Yg,".”
C a l l W r i t e L o g (1.trim(Message),SampleLog)

E n d S u b r o u t i n e

S u b r o u t i n e MaxDiameterSusp (D_max, v, g, D_Hole, Rho_Grain, Rho_Bf, Conc_Grains)
! Calculates the maximum diameter for suspension transport
! Rewritten formula, derived in (Hanks, 1980) as found in (Bisschop, 1995).
! Assumed is that the volume concentration grains in drilling fluid is the same as the sand content.
I m p l i c i t N o n e

R e a l :: D_max ! [m], maximum diameter in suspension.
R e a l :: v ! [m/s], fluid velocity based on plug flow
R e a l :: c1,c2,c3,c4,c5 ! A number of constants
R e a l :: g ! [m/s^2]
R e a l :: D_Hole ! [m], diameter hole
R e a l :: Rho_Grain, Rho_Bf ! [kg/m^3], density of the grains and the borefluid
R e a l :: Conc_Grains ! [-]
C h a r a c t e r (1024) :: Message
I n t e g e r :: SampleLog=25

c1 = 1.32 d0
\[ c2 = 0.186 d0 \]
\[ c3 = 2.0 d0 \]
\[ c4 = 0.5 d0 \]
\[ c5 = 0.23 d0 \]

Rho_Bf = 1000.d0*Rho_Bf ! Convert to [kg/m^3]

W r i t e (*, ' (A,F) ' ) "V : " , v
W r i t e (*, ' (A,F) ' ) "D_Hole : " , D_Hole
W r i t e (*, ' (A,F) ' ) "Rho_Grain : " , Rho_Grain
W r i t e (*, ' (A,F) ' ) "Rho_Bf : " , Rho_Bf
W r i t e (*, ' (A,A) ' ) "Conc_Grains : " , Conc_Grains

D_max = D_Hole * ( (v*(c3*g*D_Hole*( (Rho_Grain−Rho_Bf) / Rho_Bf ) )**(−c4) ) / (c1*Conc_Grains**c2) )**(1/c5) )

W r i t e (Message, ' (A,F,A) ' ) “The maximum diameter for suspension transport is ",D_max,".”
C a l l W r i t e L o g (1.trim(Message),SampleLog)

E n d S u b r o u t i n e

S u b r o u t i n e MudPressure (Location, DateTimeSample, AP, Activity, JointNo, Joint_MP, Mud_P)
! Reads the annular mud pressure
I m p l i c i t N o n e

C h a r a c t e r (19) :: DateTimeSample
I n t e g e r :: AP(:)
I n t e g e r :: Mud_P
I n t e g e r :: i
C h a r a c t e r (3) :: Activity (:)
C h a r a c t e r (3) :: ActivityKey
I n t e g e r (2) :: Day_MP, Month_MP
I n t e g e r :: Joint_MP (:)
I n t e g e r :: JointNo
C h a r a c t e r (*) :: Location

C a l l D a t e T i m e I n t e g e r S i n g l e B a s i c (DateTimeSample,Day_MP,Month_MP)
If (Location == 'Den Helder') Then
    If (Day_MP <= 11 and Month_MP == 9) Then
        ActivityKey = '85J'
    Else If (Day_MP >= 12 and Day_MP <= 21 and Month_MP == 9) Then
        ActivityKey = '85M'
    Else If (Day_MP >= 22 and Month_MP == 9) Then
        ActivityKey = '65J'
    End If
Else If (Location == 'Texel') Then
    If (Day_MP <= 27 and Month_MP == 7) Then
        ActivityKey = '65m'
    Else If (Day_MP >= 28 and Month_MP == 7) Then
        ActivityKey = '85m'
    Else If (Month_MP == 8) Then
        ActivityKey = '85m'
    End If
End If

Do i = 1, size(Activity)
    If (Activity(i) == ActivityKey and JointNo == Joint_MP(i)) Then
        Mud_P = AP(i)
        Write(*,*) ActivityKey, i, Activity(i), JointNo, Joint_MP(i), Mud_P
    End If
End Do
End Subroutine

Subroutine MaxPlugRadius(Tau_Y, L, DP, MaxPlugR)
    ! Calculates the maximum plug flow radius in pipes Kelessidis, 2006
    Implicit None
    Real :: MaxPlugR
    Real :: Tau_Y, DP ! DP is in [kPa]
    Real(8) :: L

    MaxPlugR = 2.0d0 * Tau_Y * (L / (DP * 1000.0d0)) ! Factor 1000 changes the unit from [kPa] to [Pa] which is the same as the Tau_Y.
End Subroutine

Subroutine MaxPlugRadius2(Tau_Y, L, DP2, MaxPlugR, D_bh, D_ds)
    ! Calculates the maximum plug flow radius in pipes Kelessidis, 2006
    Implicit None
    Real :: MaxPlugR
    Real :: Tau_Y, DP2 ! DP is in [kPa]
    Real(8) :: L
    Real :: ya, yb
    Real :: h
    Real :: D_bh, D_ds
    h = (D_bh - D_ds) / 2.0d0
    ya = (h / 2.0d0) - (Tau_Y / ((DP2 * 1000.0d0) / L))
    yb = (h / 2.0d0) + (Tau_Y / ((DP2 * 1000.0d0) / L))
    MaxPlugR = yb - ya
End Subroutine
Subroutine InfiltrationLossBTL(Location, g, Loss, Depth, Tau_Yield, D15, D_bh, Mud_P, Pi)

! Calculates the infiltration loss based on Denekamp, 2000.

Implicit None

Real :: Alpha, Kappa ! Constants
Real :: D15, D_bh
Real :: y_a, y_c
Real(8) :: pi
Real :: Mud_P ! [kPa]
Real :: n0 ! [-]
Real :: Loss ! [m/s]
Real :: Depth ! [m NAP]
Real :: dp ! [Pa]
Real :: Tau_Yield ! [Pa]
Real :: D_pores ! [m]
Real :: V_drill ! [m/s]
Real :: g

Character(*) :: Location

Alpha = 0.11d0 ! [-] 8/75
Kappa = 2.0d0 ! [-]
n0 = 0.4d0 ! [-]
D_pores = 0.44d0*D15 ! [m] 4/9
y_c = 0.002d0 ! [m], a standard value by cebo is 2 mm. no formula for this is found in Denekamp, 2000.

Write(*,*) "Mud_P, Depth : ", Mud_P, Depth
If (Location == 'Texel') dp = Mud_P * 1000.0d0 - ((-1.0d0 * (Depth - 0.18d0)) * g * 1000.0d0)
If (Location == 'Den Helder') dp = Mud_P * 1000.0d0 - ((-1.0d0 * (Depth - 0.38d0)) * g * 1000.0d0)
Write(*,*) "dp, Alpha, D_Pores, D15 : ", dp, " , " , Alpha, " , " , D_Pores, " , " , D15
y_a = (dp * Alpha * D_Pores) / (Kappa * Tau_Yield)
Write(*,*) "y_a : ", y_a
If (Location == 'Texel') v_drill = 0.75d0 / 60.0d0 ! Based on dot graph found in interpretation report, which is in [m/min]
If (location == 'Den Helder') v_drill = 1.1d0 / 60.0d0 ! Since [m/s] is required it is divided by 60.0d0.
Loss = PI * v_drill * ( n0 * ( D_bh * y_a + y_a**2) + y_c * ( D_bh - y_c))
End Subroutine

Plotting the data

Subroutine f2gp(ID, n1, n2, n3, xdata1, ydata1, style1, xdata2, ydata2, style2, plot_type, xlabel, ylabel, title1, title2, scenario)
! Writes file(s) with data for graph(s), writes a file to be read by gnuplot and calls gnuplot to read it and construct a graph according to the instructions in the file and the data in the first file(s).
! Source: http://implicitNone.com/fortran--gnuplot/, simple and more to the point than the PlotModule.f90
! Adapted to my purposes, a.o. datetime axis.
! Plots the away versus the transported length. In the mainloop since it contains the results of 1 sample.

Implicit None
`Character(*), Parameter :: filefolderScenarios_out = "D:\hjneka\Desktop\Program\ProgramOutput\Scenarios\"`

`Character(*), ID : : ID ! Sample ID, used for title of graph and graphfilename.`

`Integer :: n1, n2 ! number of data points`

`Integer :: n3 ! index of intersection of the PD data`

`Character(19), xdata1(():) ! first x data array`

`Character(19), DateTimeSource`

`Character(19), DateTimeSample`

`Character(19), DateTimeMin`

`Real(8), ydata1(():) ! first y data array`

`Character(19), xdata2(():) ! second x data array`

`Real(8), ydata2(():) ! second y data array`

`Integer :: plot_type ! 1 for linear plot, 2 for log plot, 3 for log-log plot`

`Character(len=*) :: xlabel, ylabel, title1, title2 ! plot axis labels and title`

`Integer :: TimeInt`

`Integer :: i, X, Y`

`Integer(2) :: Day, Day_Min, Day_Max`

`Integer(2) :: Month, Month_Min, Month_Max`

`Integer(4) :: Year`

`Integer(2) :: Hours, Minutes, Seconds`

`Character(2) :: style1, style2`

`Character(25) :: Layout1, Layout2`

`Integer, optional :: Scenario`

`Character(100) :: Filename`

`DateTimeSource = xdata1(n3)`

`DateTimeSample = xdata1(n1)`

`X = 1 ! Amount of days that will be subtracted from the source datetime and added to it, in order to respectively form the minimum and maximum of the zoomed plot`

`Y = 6 ! Amount of hours that will be subtracted from the source datetime and added to it, in order to respectively form the minimum and maximum of the zoomed plot`

`Call ZoomGPDays(DateTimeSource, DateTimeMin, DateTimeMax, X)`

`Call ZoomGPHours(DateTimeSource, DateTimeMin, DateTimeMax, Y)`

`Open(10, Access = 'SEQUENTIAL', File = 'xydata1.dat') ! ProData data`

`Do i = 1, n1`

`Write(10, '(A19,1x,F9.3)') xdata1(i), ydata1(i)`

`End Do`  

`Close(10, Status = 'KEEP')`

`If (Present(Scenario)) Then`

`Write(Filename, '(A,A,12.2,A)') trim(ID), 'PumpedVol Scenario', Scenario, '.txt'`

`Open(11, Access = 'Sequential', File = filefolderScenarios_out // trim(Filename))`

`Do i = 1, n1`

`Write(11, '(A19,1x,F9.3)') xdata1(i), ydata1(i)`

`End Do`  

`Close(11, Status = 'KEEP')`

`End If`  

`Open(10, Access = 'SEQUENTIAL', File = 'xydata2.dat') ! Survey Data`

`Write(10, '(A19,1x,F9.3)') xdata2(i), ydata2(i) ! Required for if loop`

`Do i = 2, n2`

`Call TimeIntervalTwoValues(TimeInt, xdata2(i), xdata2(i-1))`

`If (TimeInt > (24*60+60)) Then ! Creates gaps in the data which will cause the line to be broken in the graph.`

`Write(10,*) ! Gaps are put in the data file if the time between two points is more than 24 hrs, this means that the lines will continue for the nights over and between different drillings.`

`End If`  

`Write(10, '(A19,1x,F9.3)') xdata2(i), ydata2(i)`
End Do

Close (10, STATUS= 'KEEP')

If (Present(Scenario)) Then
  Write (Filename, '(A.A.12.2,A)') trim(ID), ' Away Scenario', Scenario, '_.txt'
  Open (11, Access= 'Sequential', File= filefolderScenarios_out//trim(Filename))
  Write (11, '(A19,F9.3)') xdata2(1), ydata2(1)
  Do i=2,n2
    If (ydata2(i) < ydata2(i-1)) Then
      Write (11,*)
    End If
    Write (11, '(A19, F9.3)') xdata2(i), ydata2(i)
  End Do
  Close (11, STATUS= 'Keep')
End If

! Create gnuplot command file

Open (10, ACCESS= 'SEQUENTIAL', FILE= 'gp.txt')
Write (10,*) ' set terminal png size 2400,800' ! Sets it to create .png
Write (Filename, '(A, I2.2, A)') trim(ID) // 'Intersection Scenario', Scenario, '_.png'
Write (10, ' (A)') ' set output "" // D:/hijnekim/Desktop/Program/Graphs/Samples//trim(filenam)e) // "" ! Creates .png file and stores it in a different folder. The "" is different from the default or fortran ("\")!!!
Write (10,*) ' set xdata time'
Write (10,*) ' set timefmt "%d-%m-%Y %H:%M:%S"
Write (10,*) ' set key opaque right bottom box'
Write (10, ' (A, A.A.A, A)') ' set xrange ["", DateTimeMin, ";", DateTimeMax, "]"'
Write (10, ' (A)') ' set yrange [0:]*'
Write (10, ' (A)') ' set multiplot'
Write (10, ' (A)') ' set title "" // "": Bold // trim(filenam)e) // "")! Graph Title, title in bold works
Write (10, ' (A)') ' set xlabel "" // Trim(xlabel) // "" offset 0,-1' ! Offset changes the position of the title, in this case lowers it, w/o it the title and tic labels overlap. First number offsets the horizontal position.
Write (10, ' (A)') ' set ylabel "" // Trim(ylabel) // ""'
Write (10, ' (A)') ' set ytics 500'
Write (10, ' (A)') ' set mytics 5'
Write (10, ' (A)') ' set grid mytics ytics xtics'
Call GraphStyle(Style1, Layout1)
Call GraphStyle(Style2, Layout2)
If (plot_type==2) Write (10,*) ' set log y' ! Logarithmic scale on the y-axis only
If (plot_type==3) Then ! Logarithmic scale on both axis.
  Write (10,*) ' set log x'
Write (10,*) ' set log y'
End If
If (n1>0. AND. n2>0) Then
  Write (10, ' (A)') ' plot "xydata1.dat" using 1:3 with '/Layout1/' // title "" // Trim(title1) // "", "xydata2.dat" using 1:3 with '/Layout2/' // title "" // Trim(title2) // "" ! Because Gnuplot reads time and data as two separate columns the y-data becomes column 3.
End If
If (n1>0. AND. n2==0) Write (10, ' (A)') ' plot "xydata1.dat" using 1:3 with '/Layout1/' // title "" // Trim(title1) // "" ! If one of the arrays is empty only one graph will be plotted.
If (n2>0. AND. n1==0) Write (10, ' (A)') ' plot "xydata2.dat" using 1:3 with '/Layout2/' // title "" // Trim(title2) // "" !

! Smaller plot
Write (10,*) ' set size 0.3,0.3'
Write(10,*) 'set origin 0.65,0.55'
Write(10,*) 'set bmargin 3; set tmargin 3; set lmargin 6; set rmargin 2'
! Determines the size of the canvas that is cleaned around the plot (excludes the title and axis)
Write(10,*) 'clear'
Write(10,*) 'set title "//"{/Bold "//"Zoom}"
Write(10,*) 'set xdata time'
Write(10,*) 'set timefmt "%d-%m-%Y %H:%M:%S"
write(10,*) 'set xrange [*:]* '
Write(10,*) 'set yrange [0:*] '
Write(10,*) 'set xlabel ""'
Write(10,*) 'set ylabel ""'
Write(10,*) 'set ytics 5000'
Write(10,*) 'unset mytics'
Write(10,*) 'set key off'
Write(10,*) 'unset grid'
Write(10,*) 'set grid'
! And finally let's plot the same set of data, but in the smaller plot
Write(10,'(A)') 'plot "xydata1.dat" using 1:3 with '//Layout1//' title "'//TRIM(title1)//"",
"xydata2.dat" using 1:3 with '//Layout2//' title "'//TRIM(title2)//"
! It's important to close the multiplot environment!!!
Write(10,*) 'unset multiplot'
Close(10,STATUS='KEEP')
! Plot curve with gnuplot and cleanup files
!-------------------------------------------------------------------------------
Call SYSTEM('gnuplot gp.txt')
! Pause ! To check the gnu file before deletion if the graph is not as expected
.Call SYSTEM('del gp.txt')
.Call SYSTEM('del xydata1.dat')
.Call SYSTEM('del xydata2.dat')
! Check message
!-------------------------------------------------------------------------------
Write(*,'(A)') 'File '//ID//' Intersection.png' '//"'/' is created.'
End Subroutine
!-------------------------------------------------------------------------------
Subroutine f2gp2(xdata1, ydata1a, ydata1b, xdata2, ydata2, plot_type, filename, title, xlabel, ylabel1, xlabel1, ylabel2, ylabel3)
! Writes file(s) with data for graph(s), write(s) a file to be read by gnuplot and calls
! gnuplot to read it and construct a graph according to the instructions in the file and
! the data in the first file(s).
! Source: http://implicitnone.com/fortran-gnuplot/, simple and more to the point than the
! PlotModule.f90
! Adapted to my purposes, a.o. datetime axis.
! Plots the length versus the transport and pump times. This is done outside of the
! mainloop since it contains the results of multiple samples.
Implicit None
 Integer :: n1,n2 ! number of data points
 Character(19) :: xdata1 (:)
 Integer :: ydata1a (:)
 ! first x data array, a
 TimeIntAvAll
 Integer :: ydata1b (:)
 ! first y data array, b
 PumpTime
 Character(19) :: xdata2 (:)
 Real(8) :: ydata2 (:)
 ! second x data array, a
 ! second y data array
 Integer :: plot_type
 ! 1 for linear plot, 2 for log
plot, 3 for log-log plot

Character(*) :: xlabel, ylabel1, ylabel2 ! plot axis labels and title
Character(*) :: title, title1, title2, title3
Integer :: TimeInt
Character(*) :: filename

n1 = size(xdata1)
n2 = size(xdata2)

! Write data on two separate files

Open(10, Access=’SEQUENTIAL’, File= ’xydata1.dat’) ! ProData and related data
Do i=1,n1
Write(10, ’(A19,1x,I9,1x,I9)’) xdata1(i), ydata1a(i), ydata1b(i)
End Do
Close(10, Status=’KEEP’)

Open(10, Access=’SEQUENTIAL’, File= ’xydata2.dat’) ! Survey and related data
Write(10, ’(A19,1x,F9.3)’) xdata2(1), ydata2(1) ! Required for if loop
Do i=2,n2
If (ydata2(i) < ydata2(i-1)) Then ! Creates gaps in the data
    Write(10,’ ’) ! Gaps are put in the data file
    if the time between two points is more than 24 hrs, this means that the lines will continue for the nightshift stops.
End If
Write(10,’(A19,1x,F9.3)’) xdata2(i), ydata2(i)
End Do
Close(10, Status=’KEEP’)

! Create gnuplot command file

Open(10, Access=’SEQUENTIAL’, File= ’gp.txt’) ! ProData and related data
Write(10,’ ’) ! Sets it to create .png
Write(10,’ ’) ! Set output ‘//’ ‘/’ ‘D:/hijnekam/Desktop/Program/Gaphs/’ ‘/‘ trim(filename)’ ‘/‘ .png
   ! Creates .png file and stores it in a different folder. The ‘’‘’ used is different from the default or format (‘”,”!!!)
Write(10,’ ’) ! Graph Title, title in bold, works
Write(10,’ ’) ! set xlabel ‘//’ ‘/‘ trim(xlabel)’ ‘/‘ offset 0,-1’ ! Offset changes the position of the title, in this case lowers it, w/o it the title and tic labels overlap.
   ! First number offsets the horizontal position.
Write(10,’ ’) ! set ylabel ‘//’ ‘/‘ trim(ylabel1)’ ‘/‘
Write(10,’ ’) ! set y2label ‘//’ ‘/‘ trim(ylabel2)’ ‘/‘
Write(10,’ ’) ! set ytics
Write(10,’ ’) ! set xdata time
Write(10,’ ’) ! set timefmt ‘ ‘%d-%m-%Y %H:%M:%S’’’
Write(10,’ ’) ! set key opaque right top box
Write(10,’ ’) ! set yrange [0:*]’ ! Forces the minimum of the y-axis to be 0 while the upperlimit is still flexible.
Write(10,’ ’) ! set y2range [0:*]
Write(10,’ ’) ! set style line 1 lc rgb ’”#d181f”’ lt 1 lw 1 pt 7 ps 1’ ! Red, TransLenSum
Write(10,’ ’) ! set style line 2 lc rgb ’”#00008b”’ lt 1 lw 1 pt 7 ps 1’ ! (dark) Blue, Away
Write(10,’ ’) ! set style line 3 lc rgb ’”#ffa500”’ lt 1 lw 1 pt 7 ps 2’ ! Orange, TimeIntAvAll
Write(10,’ ’) ! set style line 4 lc rgb ’”#006400”’ lt 1 lw 1 pt 7 ps 2’ ! (dark) Green, PumpTime
If (plot_type==2) Write(10,’ ’) ! set log y’ ! Logarithmic scale on the y-axis only
If (plot_type==3) Then
    Write(10,’ ’) ! set log x’ ! Logarithmic scale on both axis.
If(n1>0.AND.n2>0) Then

Write(10, '(A)') 'plot "xydata1.dat" using 1:3 with points ls 3 axes xyl1 title ''/TRIM(title1)'' ! If one of the arrays is empty only one graph will be plotted.

If(n2>0.AND.n1==0) Write(10, 'plot "xydata2.dat" using 1:3 with points pt 5 ps 1 title ''/TRIM(title2)'' !

Close(10,STATUS='KEEP')

! Plot curve with gnuplot and cleanup files

Call SYSTEM('gnuplot gp.txt')

! Pause ! To check the gnu file before deletion if the graph is not as expected.

Call SYSTEM('del gp.txt')

! Check message

Write(*,'(A)') 'File ''/''/trim(filename)''/''./''/'' is created.'

End Subroutine

Subroutine f2gp3(Location, xdata1, ydata1, style1, xdata2, ydata2, style2, plot_type, filename, title, xlabel, ylabel, title1, title2)

! Writes file(s) with data for graph(s), writes a file to be read by gnuplot and calls gnuplot to read it and construct a graph according to the instructions in the file and the data in the first file(s).

! Source: http://implicitnone.com/fortran-gnuplot/, simple and more to the point than the PlotModule.f90

! Adapted to my purposes, a.o. datetime axis.

! Plots the away versus the calculated average source location.

Implicit None

Integer :: n1, n2 ! number of data points
Character(19) :: xdata1(:) ! first x data array
Real(8) :: ydata1(:) ! first y data array
Character(19) :: xdata2(:) ! second x data array
Real(8) :: ydata2(:) ! second y data array
Integer :: plot_type ! 1 for linear plot, 2 for log plot, 3 for log-log plot
Character(len=*) :: xlabel, ylabel, title, title1, title2 ! plot axis labels and title
Character(*) :: filename
Integer :: TimeInt

Integer :: i

Character(2) :: style1, style2
Character(25) :: Layout1, Layout2
Character(*) :: Location

n1 = size(xdata1)
n2 = size(xdata2)

!--------------------------------------------
! Write data on two separate files

Open(10, Access= 'SEQUENTIAL', File = 'xydata1.dat')  ! ProData data
Do i=1,n1
  Write(10, '(A19,1x,F9.3)') xdata1(i),ydata1(i)
End Do
Close(10, Status='KEEP')

Open(10, Access= 'SEQUENTIAL', File = 'xydata2.dat')  ! Survey Data
Write(10, '(A19,1x,F9.3)') xdata2(1),ydata2(1)  ! Required for if loop
Do i=2,n2
  If (ydata2(i) < ydata2(i-1)) Then
    Write(10,* )  ! Gaps are put in the data file if the time between two points is more than 24 hrs, this means that the lines will continue for the nightshift stops.
  End If
  Write(10, '(A19,1x,F9.3)') xdata2(i),ydata2(i)
End Do
Close(10, Status='KEEP')

! Create gnuplot command file

Open(10, Access= 'SEQUENTIAL', File = 'gp.txt')
Write(10, ' (A) ' ) ' set terminal png size 2400,800 ' ! Sets it to create .png
Write(10, ' (A) ' ) ' set output ' '//D:/hijnekam/Desktop/Program/Graphs//' // trim (filename) // '.png // '!' Creates .png file and stores it in a different folder. The '//' used is different from the default or fortran ('"')!!!
Write(10, ' (A) ' ) ' set title ' '//"/Bold ' // trim (title) // '"' ! Graph Title, title in bold, works
Write(10, ' (A) ' ) ' set xlabel ' '//"/TRIM(xlabel)/" ' ! Offsets changes the position of the title, in this case lowers it, w/o it the title and tic labels overlap. First number offsets the horizontal position.
Write(10, ' (A) ' ) ' set ylabel ' '//"/TRIM(ylabel)/" ' ! Forces the minimum of the y-axis to be 0 while the upper limit is still flexible.
Write(10, ' (A) ' ) ' set key opaque right top box'  ! Create gnuplot command file

Call GraphStyle(Style1,Layout1)
Call GraphStyle(Style2,Layout2)
If (plot_type==2) Write(10,* ) ' set log y '  ! Logarithmic scale on the y-axis only
If (plot_type==3) Then Write(10, '(A)') ' set log x'
Write(10, '(A)') ' set log y'
End If
If (n1>0.AND.n2>0) Then
  Write(10, '(A)') ' plot "xydata1.dat" using 1:3 with ' '//Layout1//' title ""//TRIM(tit1e1)// "", "xydata2.dat" using 1:3 with ' '//Layout2//' title ""//TRIM(tit1e2)// "" ! Because GnuPlot reads time and data as two separate columns the y-data becomes column 3.
End If
If (n1>0.AND.n2==0) Write(10,* ) ' plot "xydata1.dat" using 1:3 with ' '//Layout1//' title ""//TRIM(tit1e1)// ""  ! If one of the arrays is empty only one graph will be plotted.
If (n2>0.AND.n1==0) Write(10,* ) ' plot "xydata2.dat" using 1:3 with ' '//Layout2//' title ""//TRIM(tit1e2)// ""
Close(10, Status='KEEP')

! Plot curve with gnuplot and cleanup files
Call SYSTEM('gnuplot gp.txt')
! Pause ! To check the gnu file before deletion if the graph is not as expected.

Call SYSTEM('del gp.txt')
Call SYSTEM('del xydata1.dat')
Call SYSTEM('del xydata2.dat')

! Check message

Write(*, '(A)') 'File '//'trim(filename)//' .png'//'''//' is created.'

End Subroutine

Subroutine f2gpLagD(location, title_graph, xdata1, ydata1, plot_type, xlabel, ylabel, filename, title1, style1, style2, style3, style4)
! Writes file(s) with data for graph(s), writes a file to be read by gnuplot and calls
! gnuplot to read it and construct a graph according to the instructions in the file and
! the data in the first file(s).
! Source: http://implicitinone.com/fortran-gnuplot/, simple and more to the point than the
! PlotModule.f90
! SPECIAL CASE: there is only one data set which is divided into three parts, style as such
! is put in manually instead of with the subroutine.
! Plots the lag diagram.
Implicit None

Integer :: n1 ! number of data points
Real(8) :: xdata1(:) ! first x data array
Real(8) :: ydata1(:) ! first y data array
Integer :: plot_type ! 1 for linear plot, 2 for log
Character(*) :: xlabel, ylabel, filename, title1, title_graph
Integer :: TimeLine
Character(2) :: style1, style2, style3, style4
Character(25) :: Layout1, Layout2, Layout3, Layout4
Integer :: i, j, k
Character(*) :: Location

n1 = size(xdata1)

! -----------------------------
! Write data in files
! -----------------------------
Open(10, File = 'xydata0.dat')
Open(11, File = 'xydata1.dat')
Open(12, File = 'xydata2.dat')

Do i = 1, n1  ! ROP = 0, VT=VL
  Write(10, '(F9.3,1x,F9.3)') xdata1(i), xdata1(i)
End do

If (Location == 'texel') Then
  Do i = 1, 24
    Write(11, '(F9.3,1x,F9.3)') xdata1(i), ydata1(i)
  End Do
  Do i = 25, n1
    Write(12, '(F9.3,1x,F9.3)') xdata1(i), ydata1(i)
  End Do
Else If (Location == 'den helder') Then
  Do i = 1, 20
    Write(11, '(F9.3,1x,F9.3)') xdata1(i), ydata1(i)
  End Do
End if
End Do

Do i = 21, n1 - 1 ! Last sample in Den Helder is not valid.
    Write(12, '(F9.3,1x,F9.3)') xdata1(i), ydata1(i)
End Do
End If

Close(10)
Close(11)
Close(12)

! Create gnuplot command file
!
Open(10, ACCESS= 'SEQUENTIAL', FILE= 'gp.txt')

Write(10, '(A)') 'set terminal png size 1800,1800' ! Sets it to create .png
Write(10, '(A)') 'set output' '//' '' '/D:/hijnekam/Desktop/Program/Graphs/' '//' trim(filename)://' ! pgf is different from the default or fortran ('''')''''
Write(10, '(A)') 'set title' '/''''('''Bold''''/trim(title_graph))/'''' offset 0,1 font ,,32'' ! Graph Title, title in bold, works
Write(10, '(A)') 'set xlabel' '/''''('''/trim(xlabel))/'''' offset 0,-0.5 font ,,24'' ! Offset changes the position of the title, in this case lowers it, w/o it the title and tic labels overlap. First number offsets the horizontal position.
Write(10, '(A)') 'set ylabel' '/''''('''/trim(ylabel))/'''' offset 2,0 font ,,24''
Write(10, '(A)') 'set size 0.98'
Write(10, '(A)') 'set origin 0.01,0'
Write(10, '(A)') 'set yrange [0:*]' ! Forces the minimum of the y-axis to be 0 while the upper limit is still flexible.
Write(10, '(A)') 'set grid'
Write(10, '(A)') 'set key box opaque right bottom font ,,24 spacing 1.3' ! Box can be added to put a box around the key, it overlaps with the letters so it was removed
Call GraphStyle(Style1, Layout1)
Call GraphStyle(Style2, Layout2)
Call GraphStyle(Style3, Layout3)
Call GraphStyle(Style4, Layout4)

! Write(10,'(A)') 'plot "xydata1.dat" using 1:2 with ''//Layout1'' title ''''/NAP–85m Jetbit , ''TRIM(title1))/''', "xydata2.dat" using 1:2 with ''//Layout2'' title ''''/NAP–85m Mudmotor . ''TRIM(title1))/''', "xydata3.dat" using 1:2 with ''//Layout3'' title ''''/NAP–65m Jetbit , ''TRIM(title1))/''', "xydata0.dat" using 1:2 with ''//Layout4'' title ''''/ROP = 0 , ''TRIM(title1))/'''
          
If (Location == 'Texel') Then
    Write(10,'(A)') 'plot "xydata0.dat" using 1:2 with ''//Layout4'' title ''''/ROP = 0 , ''TRIM(title1))/''', "xydata1.dat" using 1:2 with ''//Layout2'' title ''''/NAP–65m , ''TRIM(title1))/''', "xydata2.dat" using 1:2 with ''//Layout3'' title ''''/NAP–85m , ''TRIM(title1))/'''
Else If (Location == 'Den Helder') Then
    Write(10,'(A)') 'plot "xydata0.dat" using 1:2 with ''//Layout4'' title ''''/ROP = 0 , ''TRIM(title1))/''', "xydata1.dat" using 1:2 with ''//Layout2'' title ''''/NAP–85m , ''TRIM(title1))/''', "xydata2.dat" using 1:2 with ''//Layout3'' title ''''/NAP–65m , ''TRIM(title1))/'''
End If

Close(10, STATUS= 'KEEP')

! Plot curve with gnuplot and cleanup files
!
Call SYSTEM('gnuplot gp.txt')

! Pause ! To check the gnup file before deletion if the graph is not as expected
Call SYSTEM('del gp.txt')
Call SYSTEM('del xydata0.dat')
Call SYSTEM('del xydata1.dat')
Call SYSTEM('del xydata2.dat')

! Check message

Write(*,'(A)') 'File '//'trim(filename)//'.png'//'' is created.'

End Subroutine

Subroutine f2gp5(ID, xdata1, ydata1, style1, plot_type, xlabel, ylabel, title, title1)
! Writes file(s) with data for graph(s), writes a file to be read by gnuplot and calls gnuplot to read it and construct a graph according to the instructions in the file and the data in the first file(s).
! Source: http://implicitnone.com/fortran–gnuplot/, simple and more to the point than the PlotModule.f90
! Alpha Delta, subroutine is not called anymore, it is included in f2gpSampleMulti.
Implicit None
 Character(*) :: ID ! Sample ID., used for title of graph and graphfilename.
 Integer :: n1 ! number of data points
 Real(8) :: xdata1(:) ! first x data array
 Real(8) :: ydata1(:) ! first y data array
 Integer :: plot_type ! 1 for linear plot, 2 for log plot, 3 for log-log plot
 Character(2) :: style1
 Character(len=*) :: xlabel, ylabel, title, title1 ! plot axis labels and title(s)
 Character(25) :: Layout1
 Integer :: i

n1 = size(xdata1)

!Write(*,*), n1

!------------------------------------------
! Write data in files
!------------------------------------------
Open(10, Access= 'SEQUENTIAL', File= 'xydata1.dat')
Do i=1,n1
   Write(10, '(F9.3,F9.3)') xdata1(i), ydata1(i)
End do
Close(10, Status= 'KEEP')

! Create gnuplot command file
!------------------------------------------
Open(10, ACCESS= 'SEQUENTIAL', FILE= 'gp.txt')
Write(10,*) 'set terminal png size 1200,1200'  ! Sets it to create .png file
Write(10, '(A)') 'set output '//'D:hijnekam/Desktop/Program/Graphs/Plot'/'ID//''/Title//'//'png'//''  ! Creates .png file and stores it in a different folder. The "/' used is different from the default or fortran ("\")!!!
Write(10,*) 'set label '//'"{:/Bold //Title//'///ID//}"'  ! Graph Title, title in bold
   works
Write(10,*) 'set xlabel '//'TRIM(xlabel)//'' offset 0,-1'  ! Offset changes the position of the title, in this case lowers it, w/o it the title and tic labels overlap.
First number offsets the horizontal position.
Write(10,*) 'set ylabel '//'TRIM(ylabel)//''
Write(10,*) 'set yrange [0:;] '  ! Forces the minimum of the y-axis to be 0 while the upper limit is still flexible.
WRITE (10, *) 'set key opaque'
WRITE (10, *) 'set key opaque right bottom box'
CALL GraphStyle(Style1, Layout1)
IF (plot_type==2) WRITE (10, *) 'set log y'  ! Logarithmic scale on the y-axis only
IF (plot_type==3) THEN  ! Logarithmic scale on both axis.
  WRITE (10, *) 'set log x'
  WRITE (10, *) 'set log y'
END IF
WRITE (10, '(A, I, A)') 'plot "xydata1.dat" using 1:2 with ' //Layout1' //title "' //TRIM(title1)/' "', ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '
CLOSE (10, STATUS='KEEP')
! Plot curve with gnuplot and cleanup files
!-------------------------------------------
CALL SYSTEM('gnuplot gp.txt')
! Pause  ! To check the gnu file before deletion if the graph is not as expected.
CALL SYSTEM('del gp.txt')
CALL SYSTEM('del xydata1.dat')
! Check message
!-------------------------------------------
WRITE(*, ' '(A)') 'File ' //''/''/Plot ' //ID//' ''/Title//' ''/ ''/is created.'
END Subroutine

 Subroutine f2g6(ID, xdata1, ydata1, style1, xdata2, ydata2, style2, plot_type, xlabel, ylabel, title, title1, title2)

! Writes file(s) with data for graph(s), writes a file to be read by gnuplot and calls gnuplot to read it and construct a graph according to the instructions in the file and the data in the first file(s).
! Source: http://implicitnone.com/fortran–gnuplot/, simple and more to the point than the PlotModule.f90
! Fluid measurements and calculation results, subroutine is not called anymore, it is included in f2gpSampleMulti.
Implicit None
!
Char (+) :: ID  ! Sample ID., used for title of graph and graphfilename.
Int :: n1  ! number of data points
Real(8) :: xdata1(:), xdata2(:)  ! first x data array
Real(8) :: ydata1(:), ydata2(:)  ! first y data array
Int :: plot_type  ! 1 for linear plot, 2 for log-log plot
Char(len=*) :: xlabel, ylabel, title, title1, title2  ! plot axis labels and title(s)
Char(2) :: style1, style2
Char(25) :: Layout1, Layout2
Int :: i
!
!WRITE(*, *) n1
!
!----- Write data in files
!-------------------------------------------
OPEN(10, Access='SEQUENTIAL', File='xydata1.dat')
DO i=1,n1
Write (10, '(F9.3, 1x, F9.3)') xdata1(i), ydata1(i)
End do

Close (10, 'Status = ' 'KEEP')

Open (10, 'Access = ' 'SEQUENTIAL', 'File = ' 'xydata2.dat')

Do i = 1, n1
Write (10, '(F9.3, 1x, F9.3)') xdata2(i), ydata2(i)
End do

Close (10, 'Status = ' 'KEEP')

! Create gnuplot command file

Open (10, 'ACCESS = ' 'SEQUENTIAL', 'FILE = ' 'gp.txt')

Write (10, '*') 'set terminal png size 1200, 1200' ! Sets it to create .png
Write (10, '(A)') 'set output ''''/''''/D:/hijnekam/Desktop/Program/Graphs/Plot ''''//ID''''//''''
Title ''''/''''/Title ''''/''''/ID'''''''' ! Graph Title, title in bold
Write (10, '*') 'set title ''''/''''/Bold ''''/''''/ID'''''''' ! Graph Title, title in bold
Write (10, '*') 'set xlabel ''''/''''/TRIM(xlabel)''''/'''' offset 0, -1' ! Offset changes the position of the title, in this case lowers it, w/o it the title and tic labels overlap.
First number offsets the horizontal position.
Write (10, '*') 'set ylabel ''''/''''/TRIM(ylabel)''''''''
Write (10, '*') 'set yrange [0:*]' ! Forces the minimum of the y-axis to be 0 while the upper limit is still flexible.
Write (10, '*') 'set key opaque right bottom box'
Call GraphStyle (Style1, Layout1)
Call GraphStyle (Style2, Layout2)

If (plot_type == 2) Write (10, '*') 'set log y' ! Logarithmic scale on the y-axis only
If (plot_type == 3) Then
Write (10, '*') 'set log x'
Write (10, '*') 'set log y'
End If

Write (10, '(A, A)') 'plot ''''/''''/xydata1.dat using 1:2 with ''''/''''/Layout1''''/''''/title ''''/''''/TRIM(title1)''''''''
''''/''''/xydata2.dat using 1:2 with ''''/''''/Layout2''''/''''/title ''''/''''/TRIM(title2)''''''''
Close (10, 'STATUS = ' 'KEEP')

! Plot curve with gnuplot and cleanup files

Call SYSTEM ('gnuplot gp.txt')

! Pause ! To check the gnu file before deletion if the graph is not as expected.
Call SYSTEM ('del gp.txt')
Call SYSTEM ('del xydata1.dat')
Call SYSTEM ('del xydata2.dat')

! Check message
Write (*, '(A)') 'File ''''/''''/Plot ''''/''''/ID''''/''''/Title''''/''''/png''''/'''' is created.'
End Subroutine

Subroutine f2gpSampleMulti (ID, n0, n5, n6, xdata0, ydata0, style0, xdata5, ydata5, style5, xdata1, ydata1, style1, xdata2, ydata2, style2, ydata3, style3, ydata4, style4, xlabel0, ylabel0, xlabel1, ylabel1, xlabel2, ylabel2, title, title0, title1, title2, g_title1, g_title2, g_title3, g_title4,
Character(*), Parameter :: filefolder_out = "D:\hijnekam\Desktop\Program\ProgramOutput" ! Output files

Character(*), Parameter :: filefolderScenarios_out = "D:\hijnekam\Desktop\Program\ProgramOutput\Scenarios\" ! Output files

Character(*) :: ID ! Sample ID., used for title of graph and graphfilename.

Integer :: n1, n2, n0, n5, n6 ! number of data points

Real(8) :: xdata1(:,), xdata2(:,), xdata0(1,:), ydata1(:,), ydata2(:,), ydata3(:,), ydata4(:,), ydata0(1,:), ydata5(1) ! first x data array, first y data array

Character(2) :: style1, style2, style3, style4, style0, style5

Character(len=**) :: xlabel0, ylabel0, xlabel1, ylabel1, xlabel2, ylabel2

Character(len=**) :: title, title0, title1, g_title1, title2, g_title2, g_title3, g_title4, g_title5, g_title6 ! plot axis labels and title(s)

Character(25) :: Layout1, Layout2, Layout3, Layout4, Layout0, Layout5

Character(19) :: xdata0(:,), xdata5(:,)

Character(19) :: DateTimeSource

Character(19) :: DateTimeSample

Character(19) :: DateTimeMin

Character(19) :: DateTimeMax

Integer :: TimeInt

Integer :: i, Y

Integer :: TimeIntGP

Integer(2) :: Day, Day_Min, Day_Max

Integer(2) :: Month, Month_Min, Month_Max

Integer(4) :: Year

Integer(2) :: Hours, Minutes, Seconds

Integer, optional :: Scenario

Character(100) :: Filename

n1 = size(xdata1)

n2 = size(xdata2)

DateTimeSource = xdata0(n6)

DateTimeSample = xdata0(n0)

Y = 6 ! Amount of hours that will be substracted from the source datetime and added to it, in order to respectivelly form the minimum and maximum of the zoomed plot

Call ZoomGPHours(DateTimeSource, DateTimeMin, DateTimeMax, Y)

! Write data in files

Open(10, Access = 'SEQUENTIAL', File = 'xydata0.dat') ! ProData data

Do i = 1, n0

Write(10, '(A19,1x,F9.3)') xdata0(i), ydata0(i)

End Do

Close(10, Status = 'KEEP')

If (Present(Scenario)) Then

Write(Filename, '(A,A,A,12.2,A)') 'PumpedVol', trim(ID), 'Scenario', trim(Scenario), '.txt'

Write(Filename, '(A,A,12.2,A)') trim(ID), 'PumpedVol Scenario', trim(Scenario), '.txt'

Call ZoomGPHours(DateTimeSource, DateTimeMin, DateTimeMax, Y)

Write data in files

12345
Open(11, Access='Sequential', File= 'filefolderScenarios_out//trim(filename))
Do i=1, n0
  Write(11, '(A19,1x,F9.3)') xdata0(i), ydata0(i)
End Do
Close(11, Status='Keep')
End If

Open(10, Access='SEQUENTIAL', File= 'xydata5.dat') ! Survey Data
Write(10, '(A19,1x,F9.3)') xdata5(1), ydata5(1) ! Required for if loop
Do i=2, n5
  If (ydata5(i) < ydata5(i-1)) Then
    Write(10,*)
  End If
  ! Call TimeIntervalTwoValues(TimeInt, xdata5(i), xdata5(i-1))
  ! If (TimeInt >= (24*60*60)) Then
    ! Creates gaps in the data which will cause the line to be broken in the graph.
    ! Write(10,*)
    ! Gaps are put in the data file if the time between two points is more than 24 hrs, this means that the lines will continue for the nightshift stops.
  End If
  Write(10, '(A19,1x,F9.3)') xdata5(i), ydata5(i)
End Do
Close(10, Status='KEEP')

If (Present(Scenario)) Then
  Write(Filename, '(A,A,12.2,A)') trim(ID), ' Away Scenario', Scenario, '.txt'
Open(11, Access='Sequential', File= 'filefolderScenarios_out//trim(filename))
Write(11, '(A19,F9.3)') xdata5(1), ydata5(1)
Do i=2, n5
  If (ydata5(i) < ydata5(i-1)) Then
    Write(11,*)
  End If
  Write(11, '(A19,1x,F9.3)') xdata5(i), ydata5(i)
End Do
Close(11, Status='Keep')
End If

Open(10, Access='SEQUENTIAL', File= 'xydata1.dat') ! Alpha/Delta
Do i=1, n1
  Write(10, '(F9.3,1x,F9.3)') xdata1(i), ydata1(i)
End do
Close(10, Status='KEEP')

Open(10, Access='SEQUENTIAL', File= 'xydata2.dat') ! HB/Bingham/Measured
Do i=1, n2
  Write(10, '(F9.3,1x,F9.3,1x,F9.3,1x,F9.3)') xdata2(i), ydata2(i), ydata3(i), ydata4(i)
End do
Close(10, Status='KEEP')

! Create gnuplot command file
Open(10, ACCESS='SEQUENTIAL', FILE= 'gp.txt')
Write(10,*) 'set terminal png size 2400,2400' ! Sets it to create .png
Write(Filename, '(A,12.2,A)') trim(ID)//' Plot Scenario ', Scenario, '.png'
Write(10, '(A)') 'set output ""/""/D:\hijnekam/Desktop/Program/Graphs/Samples/""/trim(', filename)/""' ! Creates .png file and stores it in a different folder. The "" used is different from the default or fortran ("\") !!!!
Write(10,*) 'set multiplot title ""/""/Bold ""/trim(filename)/"" font ",22""'}
Write(10,*) 'set title '//''/[Bold]''//''Alpha vs. Delta]' ! Graph Title, title in bold.
!Write(10,8) 'set xdata'
Write(10,*) 'set xlabel '//''/[TRIM(xlabel1)][''//''
Write(10,*) 'set ylabel '//''/[TRIM(ylabel1)][''//''
Write(10,*) 'set size 0.5,0.5'
Write(10,*) 'set origin 0.0,0.0'
Write(10,*) 'set key opaque right top box'
Write(10,*) 'set grid'
write(10,'(A1)') 'plot "xydata1.dat" using 1:2 with '//Layout1//''title''//''/[TRIM(g_title1)][''//''
', 'xydata1.dat" using 1:3 with '//Layout3//''title''//''/[TRIM(g_title3)][''//''
', 'xydata2.dat" using 1:4 with '//Layout4//''title''//''/[TRIM(g_title4)][''//''
Write(10,*) 'set xdata time'
Write(10,*) 'set timefmt ''%d-%m-%Y %H:%M:%S'''
Write(10,*) 'set title '//''/[Bold]''//''Intersection]' ! Graph Title, title in bold.
!Call GraphStyle(Style1,Layout1)
!Call GraphStyle(Style2,Layout2)
!Large plot with zoomed axis.
Write(10,*) 'set xlabel '//''/[TRIM(xlabel2)][''//''
Write(10,*) 'set ylabel '//''/[TRIM(ylabel2)][''//''
Write(10,*) 'set ytics 500'
Write(10,*) 'set mytics 5'
Write(10,*) 'set size 1.049'
Write(10,*) 'set origin 0.0,0.49'
Write(10,*) 'set grid mytics ytics xtics'
Write(10,*) 'set key opaque right bottom box'
Write(10,'(A,A,A,A,A)') 'set xrange ['' ,DateTimeMin ,'' ;'' ,DateTimeMax ,'' ]'
Write(10,'(A,A,A,A,A)') 'set yrange [0:*]' ! Forces the minimum
of the y-axis to be 0 while the upper limit is still flexible.

Write(10,'(A,A,A,A,A)') 'set xrange ['' ,DateTimeMin ,'' ;'' ,DateTimeMax ,'' ]'
Write(10,'(A,A,A,A,A)') 'set yrange [0:*]' ! Forces the minimum
of the y-axis to be 0 while the upper limit is still flexible.

Call GraphStyle(Style0,Layout0)

Call GraphStyle(Style5,Layout5)

Write(10,'(A)') 'plot "xydata0.dat" using 1:3 with ''//Layout0//'' title ''//TRIM(g_title5)//''
"xydata5.dat" using 1:3 with ''//Layout5//'' title ''//TRIM(g_title6)//''
Because Gnuplot reads time and data as two separate columns the y-data becomes column 3.

! Smaller plot
Write(10,'(A)') 'set size 0.3,0.145'
Write(10,'(A)') 'set origin 0.65,0.645'
Write(10,'(A)') 'set bmargin 3; set tmargin 3; set lmargin 6; set rmargin 2'

Determines the size of the canvas that is cleaned around the plot(excludes the title and
axis)
Write(10,'(A)') 'clear'
Write(10,'(A)') 'set title ''/" Bold ''/"Overview '''''

! Write(10,'(A,A,A,A,A)') 'set xrange ['' ,DateTimeMin ,'' ;'' ,DateTimeMax ,'' ]'
! Write(10,'(A,A,A,A,A)') 'set yrange [0:*]'

Write(10,'(A)') 'set xdata time'
Write(10,'(A)') 'set timefmt '%d-%m-%Y %H:%M:%S''

write(10,'(A)') 'set xrange [*:*]'
Write(10,'(A)') 'set yrange [0:*]'

write(10,'(A)') 'set ytics 5000'
Write(10,'(A)') 'unset mytics'

Write(10,'(A)') 'set key off'
Write(10,'(A)') 'unset grid'
Write(10,'(A)') 'set grid xtics ytics'

! Smaller plot
Write(10,'(A)') 'plot "xydata0.dat" using 1:3 with ''//Layout0//'' title ''//TRIM(t_title1)//''
"xydata5.dat" using 1:3 with ''//Layout5//'' title ''//TRIM(t_title2)//''

! It’s important to close the multiplot environment!!!
Write(10,'(A)') 'unset multiplot'

Close(10,STATUS='KEEP')

Open(10, file = 'gp2.txt')

Write(10,'(A)') 'set terminal pngcairo size 1100,1100'

Write(Filename, '(A,I2.2,A)') trim(ID) // 'Fluid_Models_' //Scenario //'.png'
Write(10,'(A)') 'set output ''//''//D:hjinekam/DesktopyProgram/Graphs/Samples//''//trim( filename)''//''
! Creates .png file and stores it in a different folder. The ''/'' used
is different from the default or fortran ("/")!!!

Write(10,'(A)') 'set title font",28" ''//Bold ''//Fluid Model Comparison ''//trim(ID)//'''

write(10,'(A)') 'set xlabel ''Shear Rate [1/s]'' font",24"
Write(10,'(A)') 'set ylabel ''Shear Stress [Pa]'' font",24"

! write(10,'(A)') 'set ylabel ''Shear Rate [1/s]''
Write(10,'(A)') 'set ylabel ''Shear Stress [Pa]''
Write(10,'(A)') 'set key bottom right box font".20"

Write(10,'(A)') 'set bmargin at screen 0.07'

Write(10,'(A)') 'set grid'

Call GraphStyle(Style2,Layout2)
Call GraphStyle(Style3,Layout3)
Call GraphStyle(Style4,Layout4)

Write(10,'(A)') 'plot "xydata2.dat" using 1:2 with "//Layout2//" title ""/TRIM(g_title2)/"", "xydata2.dat" using 1:3 with "//Layout3//" title ""/TRIM(g_title3)/"", "xydata2.dat" using 1:4 with "//Layout4//" title ""/TRIM(g_title4)/"

Close(10)

! Plot curve with gnuplot and cleanup files
!---------------------------------------------------------------
Call SYSTEM( 'gnuplot gp.txt' )
Call SYSTEM( 'gnuplot gp2.txt' )

! Pause ! To check the gnu file before deletion if the graph is not as expected.

Call SYSTEM( 'del gp.txt' )
Call SYSTEM( 'del gp2.txt' )
Call SYSTEM( 'del xydata0.dat' )
Call SYSTEM( 'del xydata1.dat' )
Call SYSTEM( 'del xydata2.dat' )
Call SYSTEM( 'del xydata5.dat' )

! Check message
!---------------------------------------------------------------
Write(*,'(A)') 'File ""/trim(filename)"" is created.'

End Subroutine

Subroutine GraphStyle(Style,Layout,Index)
!
! In order to create uniformity in the output, this is separated from the plot subroutines
! so it can be used by all of those.
!
Implicit None

Character(2) :: style
Character(25) :: Layout
Integer, Optional :: Index

Select Case(style)
Case('la')
  Layout = 'lines ls 1'
  If Present(Index) then
    Write(Layout,'(A,I2)') 'lines ls ',Index
  End If
  Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "red" lt 1 lw 2 pt 7 ps 1'
End Case

Case('lb')
  Layout = 'lines ls 2'
  If Present(Index) then
    Write(Layout,'(A,I2)') 'lines ls ',Index
  End If
  Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "dark-blue" lt 1 lw 2 pt 7 ps 1'
End Case

Case('lc')
  Layout = 'lines ls 3'
  If Present(Index) then
    Write(Layout,'(A,I2)') 'lines ls ',Index
  End If
End Case
If (Present(Index)) Then
Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "orange" lt 1 lw 2 pt 7 ps 1'
Else
Write(10, '(A)') 'set style line 3 lc rgb "orange" lt 1 lw 2 pt 7 ps 1'
End If

Case ('1d')
Layout = 'lines 1s 4'
If (Present(Index)) Write(Layout, '(A, I2)') 'lines 1s ', Index
If (Present(Index)) Then
Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "dark-green" lt 1 lw 2 pt 7 ps 1'
Else
Write(10, '(A)') 'set style line 3 lc rgb "dark-green" lt 1 lw 2 pt 7 ps 1'
End If

Case ('1e')
Layout = 'lines 1s 5'
If (Present(Index)) Write(Layout, '(A, I2)') 'lines 1s ', Index
If (Present(Index)) Then
Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "magenta" lt 1 lw 2 pt 7 ps 1'
Else
Write(10, '(A)') 'set style line 4 lc rgb "magenta" lt 1 lw 2 pt 7 ps 1'
End If

Case ('1f')
Layout = 'lines 1s 6'
If (Present(Index)) Write(Layout, '(A, I2)') 'lines 1s ', Index
If (Present(Index)) Then
Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "sandybrown" lt 1 lw 2 pt 7 ps 1'
Else
Write(10, '(A)') 'set style line 5 lc rgb "sandybrown" lt 1 lw 2 pt 7 ps 1'
End If

Case ('1g')
Layout = 'lines 1s 7'
If (Present(Index)) Write(Layout, '(A, I2)') 'lines 1s ', Index
If (Present(Index)) Then
Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "orange-red" lt 1 lw 2 pt 7 ps 1'
Else
Write(10, '(A)') 'set style line 6 lc rgb "orange-red" lt 1 lw 2 pt 7 ps 1'
End If

Case ('1h')
Layout = 'lines 1s 8'
If (Present(Index)) Write(Layout, '(A, I2)') 'lines 1s ', Index
If (Present(Index)) Then
Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "spring-green" lt 1 lw 2 pt 7 ps 1'
Else
Write(10, '(A)') 'set style line 7 lc rgb "spring-green" lt 1 lw 2 pt 7 ps 1'
End If

Case ('1i')
Layout = 'lines 1s 9'
If (Present(Index)) Write(Layout, '(A, I2)') 'lines 1s ', Index
If (Present(Index)) Then
Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "cyan" lt 1 lw 2 pt 7 ps 1'
Else
Write(10, '(A)') 'set style line 8 lc rgb "cyan" lt 1 lw 2 pt 7 ps 1'
End If

Case ('1j')
Layout = 'lines 1s 10'
If (Present(Index)) Write(Layout, '(A, I2)') 'lines 1s ', Index
If (Present(Index)) Then
Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "khaki" lt 1 lw 2 pt 7 ps 1'
Else
Write(10, '(A)') 'set style line 10 lc rgb "khaki" lt 1 lw 2 pt 7 ps 1'
Case ("1k")  ! Redish
   Layout = "lines ls 11"
   If (Present(Index)) Write(Layout,"(A,I2)")  'lines ls',Index
   If (Present(Index)) Then
      Write(10, '(A,I2,A)') 'set style line ',Index,' lc rgb "#dd181f" lt 1 lw 3 pt 7 ps 1','
   Else
      Write(10,'(A)') 'set style line 11 lc rgb "#dd181f" lt 1 lw 3 pt 7 ps 1'
   End If
Case ("1l")
   Layout = "lines ls 12"
   If (Present(Index)) Write(Layout,"(A,I2)")  'lines ls',Index
   If (Present(Index)) Then
      Write(10, '(A,I2,A)') 'set style line ',Index,' lc rgb "dark−blue" lt 1 lw 3 pt 7 ps 1','
   Else
      Write(10,'(A)') 'set style line 12 lc rgb "dark−blue" lt 1 lw 3 pt 7 ps 1'
   End If
Case ("1m")
   Layout = "lines ls 13"
   If (Present(Index)) Write(Layout,"(A,I2)")  'lines ls',Index
   If (Present(Index)) Then
      Write(10, '(A,I2,A)') 'set style line ',Index,' lc rgb "orange" lt 1 lw 3 pt 7 ps 1','
   Else
      Write(10,'(A)') 'set style line 13 lc rgb "orange" lt 1 lw 3 pt 7 ps 1'
   End If
Case ("1n")
   Layout = "lines ls 14"
   If (Present(Index)) Write(Layout,"(A,I2)")  'lines ls',Index
   If (Present(Index)) Then
      Write(10, '(A,I2,A)') 'set style line ',Index,' lc rgb "dark−green" lt 1 lw 3 pt 7 ps 1','
   Else
      Write(10,'(A)') 'set style line 14 lc rgb "dark−green" lt 1 lw 3 pt 7 ps 1'
   End If
Case ("1o")
   Layout = "lines ls 15"
   If (Present(Index)) Write(Layout,"(A,I2)")  'lines ls',Index
   If (Present(Index)) Then
      Write(10, '(A,I2,A)') 'set style line ',Index,' lc rgb "magenta" lt 1 lw 3 pt 7 ps 1','
   Else
      Write(10,'(A)') 'set style line 15 lc rgb "magenta" lt 1 lw 3 pt 7 ps 1'
   End If
Case ("1p")
   Layout = "lines ls 16"
   If (Present(Index)) Write(Layout,"(A,I2)")  'lines ls',Index
   If (Present(Index)) Then
      Write(10, '(A,I2,A)') 'set style line ',Index,' lc rgb "sandybrown" lt 1 lw 3 pt 7 ps 1','
   Else
      Write(10,'(A)') 'set style line 16 lc rgb "sandybrown" lt 1 lw 3 pt 7 ps 1'
   End If
Case ("1q")
   Layout = "lines ls 17"
   If (Present(Index)) Write(Layout,"(A,I2)")  'lines ls',Index
   If (Present(Index)) Then
      Write(10, '(A,I2,A)') 'set style line ',Index,' lc rgb "orange−red" lt 1 lw 3 pt 7 ps 1','
   Else
      Write(10,'(A)') 'set style line 17 lc rgb "orange−red" lt 1 lw 3 pt 7 ps 1'
   End If
Case('1r')
Layout = 'lines ls 18'
If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls ', Index
If (Present(Index)) Then
  Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "spring-green" lt 1 lw 3 pt 7 ps 1'
Else
  Write(10, '(A)') 'set style line 18 lc rgb "spring-green" lt 1 lw 3 pt 7 ps 1'
End If

Case('1s')
Layout = 'lines ls 19'
If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls ', Index
If (Present(Index)) Then
  Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "cyan" lt 1 lw 3 pt 7 ps 1'
Else
  Write(10, '(A)') 'set style line 19 lc rgb "cyan" lt 1 lw 3 pt 7 ps 1'
End If

Case('1t')
Layout = 'lines ls 20'
If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls ', Index
If (Present(Index)) Then
  Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "khaki" lt 1 lw 3 pt 7 ps 1'
Else
  Write(10, '(A)') 'set style line 20 lc rgb "khaki" lt 1 lw 3 pt 7 ps 1'
End If

Case('1u')
Layout = 'lines ls 21'
If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls ', Index
If (Present(Index)) Then
  Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "bisque" lt 1 lw 3 pt 7 ps 1'
Else
  Write(10, '(A)') 'set style line 21 lc rgb "bisque" lt 1 lw 3 pt 7 ps 1'
End If

Case('1v')
Layout = 'lines ls 22'
If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls ', Index
If (Present(Index)) Then
  Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "plum" lt 1 lw 3 pt 7 ps 1'
Else
  Write(10, '(A)') 'set style line 22 lc rgb "plum" lt 1 lw 3 pt 7 ps 1'
End If

Case('1w')
Layout = 'lines ls 23'
If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls ', Index
If (Present(Index)) Then
  Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "medium-blue" lt 1 lw 3 pt 7 ps 1'
Else
  Write(10, '(A)') 'set style line 23 lc rgb "medium-blue" lt 1 lw 3 pt 7 ps 1'
End If

Case('1x')
Layout = 'lines ls 24'
If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls ', Index
If (Present(Index)) Then
  Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "dark-orange" lt 1 lw 3 pt 7 ps 1'
Else
  Write(10, '(A)') 'set style line 24 lc rgb "dark-orange" lt 1 lw 3 pt 7 ps 1'
End If

Case('1y')
Layout = 'lines ls 25'
If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls ', Index
If (Present(Index)) Then
  Write(10, '(A, I2, A)') 'set style line ', Index, ', lc rgb "dark-plum" lt 1 lw 3 pt 7 ps 1'
Else
   Write(10,'(A)') 'set style line 25 lc rgb "dark-plum" lt 1 lw 3 pt 7 ps 1'
End If

Case ('1z')
   Layout = 'lines ls 26'
   If (Present(Index)) Write(Layout,'(A,I2)') 'lines ls ',Index
   If (Present(Index)) Then
      Write(10,'(A,I2,Index)') 'set style line ',Index,' lc rgb "dark-goldenrod" lt 1 lw 3 pt 7 ps 1'
   Else
      Write(10,'(A)') 'set style line 26 lc rgb "dark-goldenrod" lt 1 lw 3 pt 7 ps 1'
   End If

Case ('2a')
   Layout = 'lp ls 1'
   Write(10,'(A)') 'set style line 1 lc rgb "red" lt 1 lw 2 pt 7 ps 1'

Case ('2b')
   Layout = 'lp ls 2'
   Write(10,'(A)') 'set style line 2 lc rgb "dark-blue" lt 1 lw 2 pt 7 ps 1'

Case ('2c')
   Layout = 'lp ls 3'
   Write(10,'(A)') 'set style line 3 lc rgb "orange" lt 1 lw 2 pt 7 ps 1'

Case ('2d')
   Layout = 'lp ls 4'
   Write(10,'(A)') 'set style line 4 lc rgb "dark-blue" lt 1 lw 3 pt 7 ps 2'

Case ('2e')
   Layout = 'lp ls 5'
   Write(10,'(A)') 'set style line 5 lc rgb "dark-red" lt 1 lw 2 pt 7 ps 1'

Case ('2f')
   Layout = 'lp ls 6'
   Write(10,'(A)') 'set style line 6 lc rgb "orange-red" lt 1 lw 2 pt 7 ps 1'

Case ('2g')
   Layout = 'lp ls 7'
   Write(10,'(A)') 'set style line 7 lc rgb "tan" lt 1 lw 2 pt 7 ps 1'

Case ('2h')
   Layout = 'lp ls 8'
   Write(10,'(A)') 'set style line 8 lc rgb "dark-green" lt 1 lw 2 pt 7 ps 1'

Case ('2i')
   Layout = 'lp ls 9'
   Write(10,'(A)') 'set style line 9 lc rgb "green" lt 1 lw 2 pt 7 ps 1'

Case ('2j')
   Layout = 'lp ls 10'
   Write(10,'(A)') 'set style line 10 lc rgb "chartreuse" lt 1 lw 2 pt 7 ps 1'

Case ('2k')
   Layout = 'lp ls 11'
   Write(10,'(A)') 'set style line 11 lc rgb "dark-goldenrod" lt 1 lw 2 pt 7 ps 1'

Case ('2l')
   Layout = 'lp ls 12'
   Write(10,'(A)') 'set style line 12 lc rgb "orange" lt 1 lw 2 pt 7 ps 1'

Case ('2m')
   Layout = 'lp ls 13'
   Write(10,'(A)') 'set style line 13 lc rgb "gold" lt 1 lw 2 pt 7 ps 1'

Case ('2n')
   Layout = 'lp ls 13'
   Write(10,'(A)') 'set style line 13 lc rgb "orange-red" lt 1 lw 1 pt 7 ps 2'
Case ('2o')
  Layout = 'lp ls 12'
  Write(10, '(A)') 'set style line 12 lc rgb "web-green" lt 1 lw 1 pt 7 ps 2'

Case ('3a')
  Layout = 'points ls 1'
  Write(10, '(A)') 'set style line 1 lc rgb "red" lt 1 lw 3 pt 6 ps 2'

Case ('3b')
  Layout = 'points ls 2'
  Write(10, '(A)') 'set style line 2 lc rgb "dark-blue" lt 1 lw 3 pt 6 ps 2'

Case ('3c')
  Layout = 'points ls 3'
  Write(10, '(A)') 'set style line 3 lc rgb "orange" lt 1 lw 3 pt 6 ps 2'

Case ('3d')
  Layout = 'points ls 4'
  Write(10, '(A)') 'set style line 4 lc rgb "dark-green" lt 1 lw 1 pt 6 ps 2'

Case ('3e')
  Layout = 'points ls 5'
  Write(10, '(A)') 'set style line 5 lc rgb "magenta" lt 1 lw 1 pt 6 ps 2'

Case ('3f')
  Layout = 'points ls 6'
  Write(10, '(A)') 'set style line 6 lc rgb "sandybrown" lt 1 lw 1 pt 6 ps 2'

Case ('3g')
  Layout = 'points ls 7'
  Write(10, '(A)') 'set style line 7 lc rgb "orange-red" lt 1 lw 1 pt 6 ps 2'

Case ('3h')
  Layout = 'points ls 8'
  Write(10, '(A)') 'set style line 8 lc rgb "spring-green" lt 1 lw 1 pt 6 ps 2'

Case ('3i')
  Layout = 'points ls 9'
  Write(10, '(A)') 'set style line 9 lc rgb "cyan" lt 1 lw 1 pt 6 ps 2'

Case ('3j')
  Layout = 'points ls 10'
  Write(10, '(A)') 'set style line 10 lc rgb "khaki" lt 1 lw 1 pt 6 ps 2'

Case ('3k')
  Layout = 'points ls 11'
  Write(10, '(A)') 'set style line 11 lc rgb "red" lt 1 lw 1 pt 6 ps 2'

Case ('3l')
  Layout = 'points ls 12'
  Write(10, '(A)') 'set style line 12 lc rgb "dark-blue" lt 1 lw 1 pt 6 ps 2'

Case ('3m')
  Layout = 'points ls 13'
  Write(10, '(A)') 'set style line 13 lc rgb "red" lt 1 lw 1 pt 7 ps 3'

Case ('3n')
  Layout = 'points ls 14'
  Write(10, '(A)') 'set style line 14 lc rgb "dark-blue" lt 1 lw 1 pt 7 ps 3'

Case ('3o')
  Layout = 'points ls 15'
  Write(10, '(A)') 'set style line 15 lc rgb "chartreuse" lt 1 lw 1 pt 7 ps 3'

Case ('3p')
  Layout = 'points ls 16'
  Write(10, '(A)') 'set style line 16 lc rgb "dark-green" lt 1 lw 1 pt 7 ps 3'

Case ('3q')
E. Code of the SampleSource Program

```
Layout = 'points ls 17'
Write(10,'(A)') 'set style line 17 lc rgb "magenta" lt 1 lw 1 pt 7 ps 3'

Case('3r')
Layout = 'points ls 18'
Write(10,'(A)') 'set style line 18 lc rgb "sandybrown" lt 1 lw 1 pt 7 ps 3'

Case('3s')
Layout = 'points ls 19'
Write(10,'(A)') 'set style line 19 lc rgb "cyan" lt 1 lw 1 pt 7 ps 3'

Case('3t')
Layout = 'points ls 20'
Write(10,'(A)') 'set style line 20 lc rgb "orange" lt 1 lw 1 pt 7 ps 3'

Case('3u')
Layout = 'points ls 21'
Write(10,'(A)') 'set style line 21 lc rgb "dark-violet" lt 1 lw 1 pt 7 ps 3'

Case('3v')
Layout = 'points ls 22'
Write(10,'(A)') 'set style line 22 lc rgb "dark-green" lt 1 lw 3 pt 6 ps 2'

Case('3w')
Layout = 'points ls 23'
Write(10,'(A)') 'set style line 23 lc rgb "orange-red" lt 1 lw 1 pt 7 ps 3'

Case('3x')
Layout = 'points ls 24'
Write(10,'(A)') 'set style line 24 lc rgb "chartreuse" lt 1 lw 3 pt 6 ps 2'

Case('3y')
Layout = 'points ls 25'
Write(10,'(A)') 'set style line 25 lc rgb "dark-red" lt 1 lw 3 pt 6 ps 2'

Case('3z')
Layout = 'points ls 26'
Write(10,'(A)') 'set style line 26 lc rgb "dark-green" lt 1 lw 3 pt 6 ps 2'

Case('4a')
Layout = 'circles ls 1'
Write(10,'(A)') 'set style line 1 lc rgb "gray" lw 3'

Case('4b')
Layout = 'circles ls 2'
Write(10,'(A)') 'set style line 2 lc rgb "red" lw 3'

Case('4c')
Layout = 'circles ls 3'
Write(10,'(A)') 'set style line 3 lc rgb "light-green" lw 3'

Case('4d')
Layout = 'circles ls 4'
Write(10,'(A)') 'set style line 4 lc rgb "pink" lw 3'

Case('4e')
Layout = 'circles ls 5'
Write(10,'(A)') 'set style line 5 lc rgb "dark-gray" lw 3'

Case('4f')
Layout = 'circles ls 6'
Write(10,'(A)') 'set style line 6 lc rgb "#dark-blue" lw 3'

Case('4g')
Layout = 'circles ls 7'
Write(10,'(A)') 'set style line 7 lc rgb "dark-yellow" lw 3'

Case('4h')
Layout = 'circles ls 8'
Write(10,'(A)') 'set style line 8 lc rgb "orange" lw 3'
```
Case ('5a')
  Layout = 'boxes 1s 1'
  Write(10,'(A)') 'set style lines 1 lc rgb "red" lw 3'

Case ('5b')
  Layout = 'boxes 1s 2'
  Write(10,'(A)') 'set style lines 2 lc rgb "dark-blue" lw 3'

Case Default
  Write(*,'(A)') "Graph Style unknown, process aborted."
  Return

End Select

End Subroutine

!--------------------------------------

End Module SourceModule
This appendix contains the full code of the MarsdiepVisual program written for this thesis. The code is as used, which means that it is far from optimized and parts may be present but no longer in use.

```fortran
!! ***************************************************************************
!! PROGRAM: MarsdiepVisual
!!
!! PURPOSE: Different Options are available:
!! * Option 1 creates plots based on the fluid properties (measurements).
!! * Option 2 is based on the (calculated) results of both locations. Those
!! results are not calculated in this program.
!! * Option 3 is based on the sieve data gained in the Laboratory at Deltares
!! or on calculations made in the SampleSource program.
!! * Option 4 is based on the resulting data from the SampleSource program,
!! scenarios option.
!! * Option 5 is combines files from the different scenarios into 1 file for
!! all scenarios.
!! * Option 6 is creating a graph based on the different calculated locations
!! and the sample identification results.
!!
!!***************************************************************************

Program MarsdiepVisual

Use VisualModule

Implicit None

! Variables

---

! Run Variable(s)
Integer :: Option
Integer :: i, j, k, l, m, Index
Integer :: RunOption
Integer :: EndOption
Integer :: Counter
Integer :: WCounter

! File Variable(s)
Character(*), Parameter :: filefolder_in = "D:\hijnekam\Desktop\Program\ProgramInput\"
Character(*), Parameter :: filefolder_out = "D:\hijnekam\Desktop\Program\ProgramOutput\"

Integer :: InputCSV_Sur=1, SieveResults=3, SieveResults_V=4, InputCSV_Sur_DH=5,
InputCSV_Sur_T=6, SieveResults_Old=99, FormationsTNO=7
Integer :: OutputSamples_T=21, OutputSamples_DH=22, OutputFormSieve=23,
OutputSamplesLoc_T_Ptr=24, OutputSamplesLoc_DH_Ptr=25, OutputSamplesLoc_T_Ptr=27
Integer :: FluidPropTex = 30, FluidPropTex_T = 31, FluidPropTex_DH = 32
Integer :: SampleData = 40, SampleData_T = 41, SampleData_DH = 42
Integer :: InputPart = 50, OutputScenarios = 51, OutputScenarios1 = 52, OutputScenarios2
   = 53, OutputScenarios3 = 54

! Variables regarding reading of the input files
Integer :: Nlines_sur
Character(19), Allocatable :: DateTime_Sur(1)
```

206
Integer, Allocatable :: Joint (:), Btot (:)
Real, Allocatable :: CL (:), MD (:), InclinationRaw (:), AzimuthRaw (:)
Real, Allocatable :: HS (:), Dip (:), ElevationMGS (:), R_MGS (:)
Real (8), Allocatable :: Away (:)
Real (8), Allocatable :: ElevationCalc (:)
Real (8), Allocatable :: R_Calc (:)
Integer :: Nlines_sur_DH
Character (19), Allocatable :: DateTime_Sur_DH (:)
Integer, Allocatable :: Joint_DH (:), Btot_DH (:)
Real, Allocatable :: CL_DH (:), MD_DH (:), InclinationRaw_DH (:), AzimuthRaw_DH (:)
Real, Allocatable :: HS_DH (:), Dip_DH (:), ElevationMGS_DH (:), R_MGS_DH (:)
Real (8), Allocatable :: Away_DH (:), ElevationCalc_DH (:), R_Calc_DH (:)
Integer :: Nlines_sur_DH
Character (19), Allocatable :: DateTime_Sur_DH (:)
Character (19), Allocatable :: Joint_DH (:), Btot_DH (:)
Real, Allocatable :: CL_DH (:), MD_DH (:), InclinationRaw_DH (:), AzimuthRaw_DH (:)
Real, Allocatable :: HS_DH (:), Dip_DH (:), ElevationMGS_DH (:), R_MGS_DH (:)
Real (8), Allocatable :: Away_DH (:), ElevationCalc_DH (:), R_Calc_DH (:)
Character (:), Allocatable :: Sample_ID_DH (:)
Character (:), Allocatable :: SampleCode_DH (:) ! Length is variable depending on the location.
Integer :: No_Samples
Character (10) :: Location
Character (50) :: title1, title2, title3, title4, title0
Real (8), Allocatable :: PositionBitAll (:)
Integer, Allocatable :: Temp (:), EC (:), pH (:), MF (:), Rho (:), SC (:)
Integer, Allocatable :: RPM_600 (:), RPM_300 (:), RPM_200 (:), RPM_100 (:), RPM_60 (:), RPM_30 (:), RPM_6 (:), RPM_3 (:), Gel10s (:), Gel10min (:)
Character (19), Allocatable :: DateTime_Samples (:)
Character (19), Allocatable :: DateTimeSamples_T (:)
Character (19), Allocatable :: DateTimeSamples_DH (:)
Character (:), Allocatable :: SampleCode_T (:), SampleCode_DH (:) ! Length is variable depending on the location.
Real, Allocatable :: SamplingDepthTot_T (:), SamplingAwayTot_T (:), SamplingRightTot_T (:)
Real, Allocatable :: SamplingDepthTot_DH (:), SamplingAwayTot_DH (:), SamplingRightTot_DH (:)
Real, Allocatable :: SourceDepthTot_DH (:), SourceAwayTot_DH (:), SourceRightTot_DH (:)
Real, Allocatable :: SourceDepthTot_T (:), SourceAwayTot_T (:), SourceRightTot_T (:)
Character (50) :: Title
! Option 1
Real, Allocatable :: Alpha_T (:), Alpha_DH (:)
Real, Allocatable :: DeltaBH_T (:), Delta_HB_T (:), DeltaBH_DH (:), Delta_HB_DH (:)
Real, Allocatable :: HB_K_T (:), HB_n_T (:), Yield_HB_T (:), HB_K_DH (:), HB_n_DH (:), Yield_HB_DH (:)
Real, Allocatable :: PV_T (:), YP_T (:), PV_DH (:), YP_DH (:)
Real, Allocatable :: Alpha (:)
Real, Allocatable :: DeltaBH (:), Delta_HB (:)
Real, Allocatable :: HB_K (:), HB_n (:), Yield_HB (:)
Real, Allocatable :: PV (:), YP (:)
Integer :: No_Samples_T, No_Samples_DH
Character (19), Allocatable :: DateTime_Samples_T (:), DateTime_Samples_DH (:)
Character (:), Allocatable :: Sample_ID_T (:), Sample_ID_DH (:)
Real, Allocatable :: Temp_T (:), EC_T (:), pH_T (:), MF_T (:), Rho_T (:), SC_T (:)
Integer, Allocatable :: RPM_600_T (:), RPM_300_T (:), RPM_200_T (:), RPM_100_T (:), RPM_60_T (:), RPM_30 (:), RPM_6 (:), RPM_3 (:), Gel10s_T (:), Gel10min_T (:)
Real, Allocatable :: Temp_DH (:), EC_DH (:), pH_DH (:), MF_DH (:), Rho_DH (:), SC_DH (:)
Integer, Allocatable :: RPM_600_DH (:), RPM_300_DH (:), RPM_200_DH (:), RPM_100_DH (:), RPM_60_DH (:), RPM_30_DH (:), RPM_6 (:), RPM_3 (:), Gel10s_DH (:), Gel10min_DH (:)
Real, Allocatable :: DeltaWHB1 (:), DeltaWHB2 (:), DeltaWHB1 (:), DeltaWHB2 (:)
Character (50) :: Command
Character (40), Allocatable :: Var_T (:)
Character (40), Allocatable :: Var_DH (:)
! Option 2
Integer :: ScenarioNo

! Option (3)
Character (14), Allocatable :: Sample_Code (: ) ! Contains codes from both sides so the shorter ones will have spaces at the end.
Character (14), Allocatable :: Sample_Code_V (: )
Character (50) :: Graph_Title1
Integer :: GraphOption
Real, Allocatable :: D (: , : ), D_V (: , : )
Real :: Dev
Real, Allocatable :: DevTot (: ), DevMinTot (: )
Integer :: Index1, Index2
Integer :: MinValPos
Integer, Allocatable :: TempArr (: )

Character (50), Allocatable :: Formation_V (: )
Character (50), Allocatable :: Formation_Sampling_Time (: )
Character (50), Allocatable :: Formation_ResultGroup (: )

Real, Allocatable :: Char_NA_T (: ), Char_NA_DH (: )
Real, Allocatable :: Char_URTY_T (: ), Char_URTY_DH (: )
Real, Allocatable :: Char_EE_T (: ), Char_EE_DH (: )
Real, Allocatable :: Char_URK (: ), Char_URK_DH (: )
Real, Allocatable :: Char_PZ_DH (: )
Real, Allocatable :: Char_PE_DH (: )
Real, Allocatable :: Char_BX_DH (: )
Real, Allocatable :: Char_AP_DH (: )
Real, Allocatable :: Char_EEBB_T (: ), Char_EEBB_DH (: )

! Option 4
Character (14) :: ID

! Option 5
Character (150) :: Filename
Character (150) :: Filename_Out
Character (150) :: FileContent
Character (19), Allocatable :: DateTimeSamples (: )
Character (: ), Allocatable :: SampleCode (: )
Character (40), Allocatable :: Var (: ) ! Reads the headers and stores them in a vector, e.g. Var(2) is the name of the values stored within Var2 (: ).
Real, Allocatable :: Var1 (: ), Var2 (: ), Var3 (: ), Var4 (: ), Var5 (: ), Var6 (: ), Var7 (: ), Var8 (: ), Var9 (: ), Var10 (: ), Var11 (: ), Var12 (: ), Var13 (: ), Var14 (: ), Var15 (: ), Var16 (: ), Var17 (: ), Var18 (: )

Real, Allocatable :: Var1_Scenario01 (: ), Var1_Scenario02 (: ), Var1_Scenario03 (: ), Var1_Scenario04 (: ), Var1_Scenario05 (: ), Var1_Scenario06 (: ), Var1_Scenario07 (: ), Var1_Scenario08 (: ), Var1_Scenario09 (: )
Real, Allocatable :: Var2_Scenario01 (: ), Var2_Scenario02 (: ), Var2_Scenario03 (: ), Var2_Scenario04 (: ), Var2_Scenario05 (: ), Var2_Scenario06 (: ), Var2_Scenario07 (: ), Var2_Scenario08 (: ), Var2_Scenario09 (: )
Real, Allocatable :: Var3_Scenario01 (: ), Var3_Scenario02 (: ), Var3_Scenario03 (: ), Var3_Scenario04 (: ), Var3_Scenario05 (: ), Var3_Scenario06 (: ), Var3_Scenario07 (: ), Var3_Scenario08 (: ), Var3_Scenario09 (: )
Real, Allocatable :: Var4_Scenario01 (: ), Var4_Scenario02 (: ), Var4_Scenario03 (: ), Var4_Scenario04 (: ), Var4_Scenario05 (: ), Var4_Scenario06 (: ), Var4_Scenario07 (: ), Var4_Scenario08 (: ), Var4_Scenario09 (: )
Real, Allocatable :: Var5_Scenario01 (: ), Var5_Scenario02 (: ), Var5_Scenario03 (: ), Var5_Scenario04 (: ), Var5_Scenario05 (: ), Var5_Scenario06 (: ), Var5_Scenario07 (: ), Var5_Scenario08 (: ), Var5_Scenario09 (: )
Real, Allocatable :: Var6_Scenario01 (: ), Var6_Scenario02 (: ), Var6_Scenario03 (: ), Var6_Scenario04 (: ), Var6_Scenario05 (: ), Var6_Scenario06 (: ), Var6_Scenario07 (: ), Var6_Scenario08 (: ), Var6_Scenario09 (: )
Real, Allocatable :: Var7_Scenario01 (: ), Var7_Scenario02 (: ), Var7_Scenario03 (: ), Var7_Scenario04 (: ), Var7_Scenario05 (: ), Var7_Scenario06 (: ), Var7_Scenario07 (: ), Var7_Scenario08 (: ), Var7_Scenario09 (: )
Real, Allocatable :: Var8_Scenario01 (: ), Var8_Scenario02 (: ), Var8_Scenario03 (: ), Var8_Scenario04 (: ), Var8_Scenario05 (: ), Var8_Scenario06 (: ), Var8_Scenario07 (: ), Var8_Scenario08 (: ), Var8_Scenario09 (: )
Real, Allocatable :: Var9_Scenario01 (: ), Var9_Scenario02 (: ), Var9_Scenario03 (: ), Var9_Scenario04 (: ),Var9_Scenario05 (: ), Var9_Scenario06 (: ), Var9_Scenario07 (: ), Var9_Scenario08 (: ), Var9_Scenario09 (: )
Option 6 is creating a graph based on the different calculated locations and the sample identification results.
Option 5 is combining files from the different scenarios into 1 file for all scenarios.
Option 4 is based on the resulting data from the SampleSource program.
Option 3 is based on the sieve data gained in the Laboratory at Deltares or on calculations made in the SampleSource program.
Option 2 is based on the calculated results of both locations. Those results are not calculated in this program.
Option 1 creates plots based on the fluid properties (measurements).
Give an option or '0' for a list of options:
This program is meant for visualisation of raw data and data calculated elsewhere.

Continue ! Allows for a call from the end of the program in order to enable runs of the different parts without exiting the program.
Give an option or '0' for a list of options:

Case (Option)

Case (0)
Write(*,'(A)') "Option 1 creates plots based on the fluid properties (measurements)."
Write(*,'(A)') "Option 2 is based on the (calculated) results of both locations. Those results are not calculated in this program."
Write(*,'(A)') "Option 3 is based on the sieve data gained in the Laboratory at Deltares or on calculations made in the SampleSource program."
Write(*,'(A)') "Option 4 is based on the resulting data from the SampleSource program. scenarios option."
Write(*,'(A)') "Option 5 is combining files from the different scenarios into 1 file for all scenarios."
Write(*,'(A)') "Option 6 is creating a graph based on the different calculated locations and the sample identification results."
Write(*,'(A)') "This program is meant for visualisation of raw data and data calculated elsewhere."
Write(*,'(A)') "Give an option or '0' for a list of options:"

Case (1)
Counter = 0
Continue

If (Counter < 2) Then

If (Counter == 0) Then
    Location = 'T'
Else If (Counter == 1) Then
    Location = 'DH'
End If

Select Case (Location)

Case ('Den Helder', 'den heller', 'dh', 'DH', 'Dh')
    Location = 'Den Helder'
    Write (*, '(A,A)') 'The location is', trim(Location)
    Open (InputCSV_Sur, file = filefolder_in//"InputData_Sur_DH_Total.csv")
    Open (SampleData, file = filefolder_in//"SampleData_DH.csv")
    Open (FluidPropTex, file = filefolder_out//"Full01-03/DH_FluidPropTex_Scenario_01.txt")
    title1 = 'Den_Helder_Drilled_Paths'
    Case ('Texel', 'texel', 'txel', 'TXL', 'T', 'T')
    Location = 'Texel'
    Write (*, '(A,A)') 'The location is', trim(Location)
    Open (InputCSV_Sur, file = filefolder_in//"InputData_Sur_Texel_Total.csv")
    Open (SampleData, file = filefolder_in//"SampleData_Texel.csv")
    Open (FluidPropTex, file = filefolder_out//"Full01-03/T_FluidPropTex_Scenario_01.txt")
    title1 = 'Texel_Drilled_Paths'
End Select

title0 = trim(short(Location))// 'Drilled Paths'
title2 = trim(short(Location))// 'Rheologic Data'
title3 = trim(short(Location))// 'Density vs. Sand Content'
title4 = trim(short(Location))// 'Temperature vs. Marsh Funnel'

Call ReadingFileCSV_Sur (InputCSV_Sur, Nlines_sur, DateTime_Sur, Joint, CL, MD, InclinationRaw, AzimuthRaw, Btot, Dip, ElevationCalc, ElevationMGS, Away, R_Calc, R_MGS)

Call ReadingFileSampleData (Location, SampleData, No_Samples, DateTime_Samples, Sample_ID, Temp, EC, pH, Rho, SC, RPM_600, RPM_300, RPM_200, RPM_100, RPM_60, RPM_30, RPM_6, RPM_3, Gel10s, Gel10min)

Call ReadingFileOutputTxt (Location, FluidPropTex, Var, SampleCode, DateTimeSamples, Yield_HB, HB_K, HB_n, Alpha, Delta_HB, PV, YP, Delta_BH, DeltaWHB1, DeltaWHB2, DeltaW1, DeltaW2)

! Main Option 1
!---------------------------------------------

Call f2gp3D (Location, Away, R_Calc, ElevationCalc, title1, 'Away [m]', 'R_{Calculated} [m]', 'Elevation [m]', title0)

Call f2gpRhData (Location, Sample_ID, RPM_600, '3a', RPM_300, '3b', RPM_200, '3c', RPM_100, '3d', RPM_60, '3e', RPM_30, '3f', RPM_6, '3g', RPM_3, '3h', Gel10s, '3i', Gel10min, '3j', title2, 'Samples', 'Readings [{\{260\}}', 'RPM_{600} '[P]', 'RPM_{300} '[P]', 'RPM_{200} '[P]', 'RPM_{100} '[P]', 'RPM_{60} '[P]', 'RPM_{30} '[P]', 'RPM_{6} '[P]', 'RPM_{3} '[P]', 'Gel10s', 'Gel10min]')

Call f2gpSamplePar (Location, Sample_ID, Rho, '3a', '3b', SC, '3c', '3b', 'Samples', 'Density [kg/m^3]', 'Percentage [%]', 'Density', 'Sand Content')

Call f2gpData (Location, 'Yield Stress', 'Yield Stress', 'Sample Code', 'Yield Stress [Pa]', SampleCode, Yield_HB, 'HB Tau Yield', '3a', 'YP', 'YP', '3b')
'Call f2gpData(Location, 'Best_Fit', 'Best Fit', 'Sample Code', 'Deviation', SampleCode, Delta_HB, Delta_HB, '3k', Delta_BH, Delta_BH, '31')
Call f2gpYPData(Location, 'Best_Fit_YP', 'Best Fit', 'Sample Code', 'Deviation [-]', SampleCode, Delta_HB, 'Deviation HB', '3x', Delta_BH, 'Deviation BH', '3k', Yield_HB, 'HB Yield Stress', '3v', YP, 'BH Yield Stress', '3y')
Call f2gpYPData(Location, 'Best_Fit_YP_W1', 'Best Fit W1', 'Sample Code', 'Deviation [-]', SampleCode, DeltaWHB1, 'Deviation HB', '3x', DeltaWHB1, 'Deviation BH', '3k', Yield_HB, 'HB Yield Stress', '3v', YP, 'BH Yield Stress', '3y')
Call f2gpYPData(Location, 'Best_Fit_YP_W2', 'Best Fit W2', 'Sample Code', 'Deviation [-]', SampleCode, DeltaWHB2, 'Deviation HB', '3x', DeltaWHB2, 'Deviation BH', '3k', Yield_HB, 'HB Yield Stress', '3v', YP, 'BH Yield Stress', '3y')


Else If (Counter == 2) Then
Open(FluidPropTex_T, file = filefolder_in/"T_FluidPropTex_Scenario_01.txt")
Open(FluidPropTex_DH, file = filefolder_in/"DH_FluidPropTex_Scenario_01.txt")
Open(SampleData_T, file = filefolder_in/"SampleData_Texel.csv")
Open(SampleData_DH, file = filefolder_in/"SampleData_DH.csv")

Location = ' Texel'
Call ReadingFileOutputTxt(Location, FluidPropTex_T, Var_T, SampleCode_T, DateTimeSamples_T, Yield_HB_T, HB_K_T, HB_n_T, Alpha_T, Delta_HB_T, PV_T, YP_T, Delta_BH_T)
Call ReadingFileSampleData(Location, SampleData_T, No_Samples_T, DateTime_Samples_T, Sample_ID_T, Temp_T, EC_T, pH_T, MF_T, Rho_T, SC_T, RPM_600_T, RPM_300_T, RPM_200_T, RPM_100_T, RPM_60_T, RPM_30_T, RPM_6_T, RPM_3_T, Gel10s_T, Gel10min_T)

Location = ' Den Held er'
Call ReadingFileOutputTxt(Location, FluidPropTex_DH, Var_DH, SampleCode_DH, DateTimeSamples_DH, Yield_HB_DH, HB_K_DH, HB_n_DH, Alpha_DH, Delta_DH_DH, PV_DH, YP_DH, Delta_BH_DH)
Call ReadingFileSampleData(Location, SampleData_DH, No_Samples_DH, DateTime_Samples_DH, Sample_ID_DH, Temp_DH, EC_DH, pH_DH, MF_DH, Rho_DH, SC_DH, RPM_600_DH, RPM_300_DH, RPM_200_DH, RPM_100_DH, RPM_60_DH, RPM_30_DH, RPM_6_DH, RPM_3_DH, Gel10s_DH, Gel10min_DH)


Call f2gpDataHisto('Deviation from Data', 'Sample Code', 'Deviation [-]', SampleCode_T, Delta_BH_T, Delta_BH_DH, 'Texel', '5a', SampleCode_DH, Delta_BH_DH, Delta_BH_T, 'Den Helder', '5b')

End If

! Closing this Option
-------------------------------------------------------------
If (Counter < 2) Then
Close(InputCSV_Sur)
Close(SampleData)
Close(SampleData_T)
Close(SampleData_DH)
Close(FluidPropTex)
Deallocation(DateTime_Sur, Joint, CL, MD, InclinationRaw, AzimuthRaw, BTot, Dip, ElevationCalc,
ElevationMGS, Away, R_Calc, R_MGS)
Dealocate (DateTime_Samples, Sample_ID, Temp, EC, pH, Rho, SC, RPM_600, RPM_300, RPM_200,
RPM_100, RPM_60, RPM_30, RPM_6, RPM_3, Gel10s, Gel10min)
Dealocate (Var, SampleCode, DateTimeSamples, Yield_HB, HB_K, HB_n, Alpha, Delta_HB, PV,
YP, Delta_BH)
Dealocate (DeltaWHB1, DeltaWHB2, DeltaWBH1, DeltaWBH2)
Else If (Counter == 2) Then
Close (SampleData_T)
Close (SampleData_DH)
Close (FluidPropTex_T)
Close (FluidPropTex_DH)
Dealocate (Var_T, SampleCode_T, DateTime_Samples_T, Sample_ID_T, Temp_T, EC_T, pH_T, MF_T,
Rho_T, SC_T, RPM_600_T, RPM_300_T, RPM_200_T, RPM_100_T, RPM_60_T, RPM_30_T, RPM_6_T, RPM_3_T,
Gel10s_T, Gel10min_T)
Dealocate (Var_DH, SampleCode_DH, DateTime_Samples_DH, Sample_ID_DH, Temp_DH, EC_DH, pH_DH,
MF_DH, Rho_DH, SC_DH, RPM_600_DH, RPM_300_DH, RPM_200_DH, RPM_100_DH, RPM_60_DH, RPM_30_DH,
RPM_6_DH, RPM_3_DH, Gel10s_DH, Gel10min_DH)
End If
Counter = Counter + 1
If (Counter <= 2) Then
Write (*, '(A, I)') "Counter is ", Counter
Write (*, '+')
Write (*, '(A)"

Go To 200
End If
Case (2)
Write (*, '(A)') "Give the option for the graph on top of the cross-section, either '1' (Head Locations at Sampling Time), '2' (Calculated sample source), '3' (Drilled Paths), '4' (Option 1 + 3), '5' (Option 2 + 3) or '10' (All of the above):"
Read (*, '(I)') GraphOption
Write (*, '+')
Write (*, '(A)"
------------------------------------------------------------------------------------------
Write (*, '+')
ScenarioNo = 5
Open (InputCSV_Sur_DH, file = filefolder_in//"InputData_Sur_DH_Total.csv")
Open (InputCSV_Sur_T, file = filefolder_in//"InputData_Sur_Texel_Total.csv")
Write (Filename, '(A, 12, 2, A)') "T_OutputSamples_Scenario_" , ScenarioNo , " .csv"
Open (OutputSamples_T, file = filefolder_out//trim(Filename))
Write (Filename, '(A, 12, 2, A)') "T_OutputSamplesLoc_Scenario_" , ScenarioNo , " .txt"
Open (OutputSamplesLoc_T_txt, file = filefolder_out//trim(Filename))
Write (Filename, '(A, 12, 2, A)') "DH_OutputSamples_Scenario_" , ScenarioNo , " .csv"
Open (OutputSamples_DH, file = filefolder_out//trim(Filename))
Write (Filename, '(A, 12, 2, A)') "DH_OutputSamplesLoc_Scenario_" , ScenarioNo , " .txt"
Open (OutputSamplesLoc_DH_txt, file = filefolder_out//trim(Filename))
Location = 'Texel'
Call ReadingFileOutputSamplesLocTxt(Location, OutputSamplesLoc_T_txt, SampleCode_T,
DateTimeSamples_T, SamplingAwayTot_T, SamplingDepthTot_T, SamplingRightTot_T, SourceAwayTot_T,
SourceDepthTot_T, SourceRightTot_T)
Location = 'Den Helder'
Call ReadingFileOutputSamplesLocTxt(Location, OutputSamplesLoc_DH_txt, SampleCode_DH,
DateTimeSamples_DH, SamplingAwayTot_DH, SamplingDepthTot_DH, SamplingRightTot_DH,
SourceAwayTot_DH, SourceDepthTot_DH, SourceRightTot_DH)
Call ReadingFileCSV_Sur(InputCSV_Sur_T, Nlines_sur_T, DateTime_Sur_T, Joint_T, CL_T, MD_T,
InclinationRaw_T, AzimuthRaw_T, Btot_T, Dip_T, ElevationCalc_T, ElevationMGS_T, Away_T, R_Calc_T, R_MGS_T

Call ReadingFileCSV_Sur (InputCSV_Sur_DH, Nlines_sur_DH, DateTime_Sur_DH, Joint_DH, CL_DH, MD_DH, InclinationRaw_DH, AzimuthRaw_DH, Btot_DH, Dip_DH, ElevationCalc_DH, ElevationMGS_DH, Away_DH, R_Calc_DH, R_MGS_DH)

! Main Option 2

Select Case (GraphOption)

Case (1)
filename = 'Head_Location_at_Sampling_Time'
title = 'Head Location at Sampling Time'
Call f2gpwBG (GraphOption, filename, SampleCode_T, SamplingAwayTot_T, SamplingDepthTot_T, '3k', Away_T, ElevationCalc_T, '1a', SampleCode_DH, SamplingAwayTot_DH, SamplingDepthTot_DH, '3l', Away_DH, ElevationCalc_DH, '1b', title, 'Sample Code', 'Height [m NAP]', 'Samples Texel', 'Samples Den Helder', 'Drilled Paths Texel', 'Drilled Paths Den Helder')

Case (2)
filename = 'Sample_Source_Locations'
title = 'Sample Source Locations'
Call f2gpwBG (SampleCode_T, SourceAwayTot_T, SourceDepthTot_T, '3k', SampleCode_DH, SourceAwayTot_DH, SourceDepthTot_DH, '3l', title, 'Sample Code', 'Height [m NAP]', 'Samples Texel', 'Samples Den Helder')

Case (3)
filename = 'Drilled_Paths'
title = 'Drilled Paths'
Call f2gpwBG (GraphOption, filename, SampleCode_T, SamplingAwayTot_T, SamplingDepthTot_T, '3k', Away_T, ElevationCalc_T, '1k', SampleCode_DH, SamplingAwayTot_DH, SamplingDepthTot_DH, '3l', Away_DH, ElevationCalc_DH, '1l', title, 'Sample Code', 'Height [m NAP]', 'Samples Texel', 'Samples Den Helder', 'Drilled Paths Texel', 'Drilled Paths Den Helder')

Case (4)
filename = 'Drilled_Paths_and_Head_Location_at_Sampling_Time'
title = 'Drilled Paths and Head Location at Sampling Time'
Call f2gpwBG (GraphOption, filename, SampleCode_T, SamplingAwayTot_T, SamplingDepthTot_T, '3k', Away_T, ElevationCalc_T, '1a', SampleCode_DH, SamplingAwayTot_DH, SamplingDepthTot_DH, '3l', Away_DH, ElevationCalc_DH, '1b', title, 'Sample Code', 'Height [m NAP]', 'Samples Texel', 'Samples Den Helder', 'Drilled Paths Texel', 'Drilled Paths Den Helder')

Case (5)
Write (+, '(A)') "To Be Done..."
! title = 'Drilled Paths and Sample Source Locations'
! Call f2gpwBG (GraphOption, filename, SampleCode_T, SamplingAwayTot_T, SamplingDepthTot_T, '3l', Away_T, ElevationCalc_T, '1b', SampleCode_DH, SamplingAwayTot_DH, SamplingDepthTot_DH, '3k', Away_DH, ElevationCalc_DH, '1l', title, 'Sample Code', 'Height [m NAP]', 'Samples Texel', 'Samples Den Helder', 'Drilled Paths Texel', 'Drilled Paths Den Helder')

Case (6)
title = 'Formations TNO'
Call f2gpwBG (SampleCode_T, SourceAwayTot_T, SourceDepthTot_T, '3k', SampleCode_DH, SourceAwayTot_DH, SourceDepthTot_DH, '3l', title, 'Sample Code', 'Height [m NAP]', 'Samples Texel', 'Samples Den Helder')

Case (10)
filename = 'Head_Location_at_Sampling_Time'
title = 'Head Location at Sampling Time'
GraphOption = 1
Call f2gpwBG (GraphOption, filename, SampleCode_T, SamplingAwayTot_T, SamplingDepthTot_T, '3k', Away_T, ElevationCalc_T, '1a', SampleCode_DH, SamplingAwayTot_DH, SamplingDepthTot_DH, '3l', Away_DH, ElevationCalc_DH, '1b', title, 'Sample Code', 'Height [m NAP]', 'Samples Texel', 'Samples Den Helder', 'Drilled Paths Texel', 'Drilled Paths Den Helder')

Call f2gpwBG (SampleCode_T, SourceAwayTot_T, SourceDepthTot_T, '3k', SampleCode_DH, SourceAwayTot_DH, SourceDepthTot_DH, '3l', title, 'Sample Code', 'Height [m NAP]', 'Samples Texel', 'Samples Den Helder')

! Has no effect since it calls a different subroutine that has yet to be merged with the f2gpbw2 subroutine. TO BE DONE.
filename = 'Drilled_Paths'
title = 'Drilled Paths'

GraphOption = 3
Call f2gpwBG2(GraphOption , filename , SampleCode_T , SamplingAwayTot_T , SamplingDepthTot_T ,
'3k' , Away_T , ElevationCalc_T , '1a' , SampleCode_DH , SamplingAwayTot_DH , SamplingDepthTot_DH , '3l' , Away_DH , ElevationCalc_DH , '1b' , title , 'Sample Code ' , 'Height [m NAP] ' , 'Samples Texel ' ,
Samples Den Helder ' , ' Drilled Paths Texel ' , ' Drilled Paths Den Helder ' )

filename = 'Drilled_Paths_and_Head_Location_at_Sampling_Time'
title = 'Drilled Paths and Head Location at Sampling Time'

GraphOption = 4
Call f2gpwBG2(GraphOption , filename , SampleCode_T , SamplingAwayTot_T , SamplingDepthTot_T ,
'3k' , Away_T , ElevationCalc_T , '1a' , SampleCode_DH , SamplingAwayTot_DH , SamplingDepthTot_DH , '3l' , Away_DH , ElevationCalc_DH , '1b' , title , 'Sample Code ' , 'Height [m NAP] ' , 'Samples Texel ' ,
Samples Den Helder ' , ' Drilled Paths Texel ' , ' Drilled Paths Den Helder ' )

End Select

! Files for Hans – containing the location of the samples in tables
!---------------------------------------------------------------------
Open(10, File= filefolder_out //"LocationSampleTime_T . csv")
Write (10, ' (A) ' ) ' Texel '
Write (10, ' ( ? ( A ) ) ' ) ' Sample Code ' , ' ; ' , ' Away [ m ] ' , ' ; ' , ' Depth [ m ] ' , ' ; ' , ' Right [ m ] ' 
Do i = 1 , size ( SampleCode_T )
Write (10, ' ( A , 3 ( A , F7 . 2 ) ) ' ) SampleCode_T(i) , ' ; ' , SamplingAwayTot_T(i) , ' ; ' , SamplingDepthTot_T(i) , ' ; ' , SamplingRightTot_T(i)
End Do
Close (10, Status='KEEP')

Open(10, File= filefolder_out //"LocationSampleTime_DH . csv")
Write (10, ' (A) ' ) ' Den Helder '
Write (10, ' ( ? ( A ) ) ' ) ' Sample Code ' , ' ; ' , ' Away [ m ] ' , ' ; ' , ' Depth [ m ] ' , ' ; ' , ' Right [ m ] ' 
Do i = 1 , size ( SampleCode_DH )
Write (10, ' ( A , 3 ( A , F7 . 2 ) ) ' ) SampleCode_DH(i) , ' ; ' , SamplingAwayTot_DH(i) , ' ; ' , SamplingDepthTot_DH(i) , ' ; ' , SamplingRightTot_DH(i)
End Do
Close (10, Status='KEEP')

! Closing this Option
Close ( OutputSamples_T )
Close ( OutputSamples_DH )
Close ( OutputSamplesLoc_T_Txt )
Close ( InputCSV_Sur_DH )
Close ( InputCSV_Sur_T )

Deallocate ( DateTimeSamples_T , SampleCode_T , SamplingDepthTot_T , SamplingAwayTot_T ,
SamplingRightTot_T , SourceDepthTot_T , SourceAwayTot_T , SourceRightTot_T )
Deallocate ( DateTimeSamples_DH , SampleCode_DH , SamplingDepthTot_DH , SamplingAwayTot_DH ,
SamplingRightTot_DH , SourceDepthTot_DH , SourceAwayTot_DH , SourceRightTot_DH )
Deallocate ( DateTime_Sur_T , Joint_T , CL_T, MD_T, InclinationRaw_T , AzimuthRaw_T , Btot_T , Dip_T ,
ElevationCalc_T , ElevationMGS_T , Away_T , R_Calc_T , R_MGS_T )
Deallocate ( DateTime_Sur_DH , Joint_DH , CL_DH, MD_DH, InclinationRaw_DH , AzimuthRaw_DH , Btot_DH ,
Dip_DH, ElevationCalc_DH , ElevationMGS_DH , Away_DH , R_Calc_DH , R_MGS_DH )

Case (3)

! Opening, reading and allocating
Write (*, ' (A) ' ) "Give program run option , ' 1 ' = fast , ' 2 ' = full : "
Read (*, ' ( I ) ' ) RunOption

Write (*, *)
Write (*, ' (A) ' ) "---------------------------------------------------------------------"
Write (*)
Open(SieveResults_Old, file = filefolder_in//"SieveResults.csv")

Open(SieveResults, file = filefolder_in//"SieveResultsV3.csv")
  ...V1 is a file composed of different gef values, created by someone else, ...V2 composed of the required values but some .gef files had invalid data, ...V3 is corrected for the errors from the .gef files.

Open(SieveResults_V, file = filefolder_in//"SieveResults_V.csv")  ! Samples from the initial site investigations.

Open(OutputFormSieve, file = filefolder_out//"OutputFormatSieve.csv")
Write(OutputFormSieve, '(A) "Sample_Code ; Formation_Sampling_Time ; DevMinTot ;
Sample_Code_V ; Formation_V ; Formation_Grouped" ! Writes the headers into the file, the file is filled with data in a loop further on in the program.

! Call ReadingFileSieve(SieveResults_Old,Sample_Code,D)
Call ReadingFileSieve(SieveResults_V,Sample_Code_V,D_V)
Call ReadingFileSieveV2(SieveResults,Sample_Code,D)

Allocate(DevTot(size(D_V)/99))
Allocate(DevMinTot(size(D)/99))
Allocate(Formation_V(size(D_V)/99))
Allocate(Formation_Sampling_Time(size(D)/99))
Allocate(Formation_ResultGroup(size(D)/99))
Allocate(Char_NA_T(99))
Allocate(Char_NA_DH(99))
Allocate(Char_EEBB_T(99))
Allocate(Char_EEBB_DH(99))
Allocate(Char_URTY_T(99))
Allocate(Char_URTY_DH(99))
Allocate(Char_EE_T(99))
Allocate(Char_EE_DH(99))
Allocate(Char_URK_DH(99))
Allocate(Char_PZ_DH(99))
Allocate(Char_PE_DH(99))
Allocate(Char_BX_DH(99))
Allocate(Char_AP_DH(99))

! Assigning and calculations

! Based on the sieve curves created from the samples taken from the vertical drillings, as found in the Grononderzoek rapport.
Formation_V(1:5) = 'Formation of Naaldwijk T (NA)''
Char_NA_T = (D(1,:) + D(2,:) + D(3,:) + D(4,:) + D(5,:))/5

Formation_V(6:10) = 'Eem Formation, Bruine Bank T (EE−BB)''
Char_EEBB_T = (D(6,:) + D(7,:) + D(8,:) + D(9,:) + D(10,:))/5

Formation_V(11:14) = 'Eem Formation T (EE)''
Char_EE_T = (D(11,:) + D(12,:) + D(13,:) + D(14,:))/4

Formation_V(15:19) = 'Formation of Urk, Tynje T (URTY)''
Char_URTY_T = (D(15,:) + D(16,:) + D(17,:) + D(18,:) + D(19,:))/5

Formation_V(20) = 'Formation of Naaldwijk DH (NA)''
Char_NA_DH = D(20,:)

Formation_V(21) = 'Formation of Boxtel DH (BX)''
Char_BX_DH = D(21,:)

Formation_V(22:24) = 'Eem Formation, Bruine Bank DH (EE−BB)''
Char_EEBB_DH = (D(22,:) + D(23,:) + D(24,:))/3

Formation_V(25:29) = 'Eem Formation DH (EE)''
Char_EE_DH = (D(25,:) + D(26,:) + D(27,:) + D(28,:) + D(29,:))/5

Formation_V(30:32) = 'Formation of Urk, Tynje DH (URTY)''
Char_URTY_DH = (D(30,:) + D(31,:) + D(32,:))/3
Formation_V(33:36) = 'Peelo Formation DH (PE)'
Char_PE_DH = \(\frac{D(33,:)+D(34,:)+D(35,:)+D(36,:)}{4}\)

Formation_V(37) = 'Formation of Urk DH'
Char_URK_DH = D(37,:)

Formation_V(38:41) = 'Formation of Appelscha DH (AP)'
Char_AP_DH = \(\frac{D(38,:)+D(39,:)+D(40,:)+D(41,:)}{4}\)

Formation_V(42:43) = 'Formation of Peize DH (PZ)'
Char_PZ_DH = \(\frac{D(42,:)+D(43,:)}{2}\)

! Based on the cross-section made based on seismic data, CPT data and drillings.
Formation_Sampling_Time(1) = 'Eem Formation, Bruine Bank DH (EE-BB)'
Formation_Sampling_Time(2:11) = 'Peelo Formation DH (PE)'
Formation_Sampling_Time(12) = 'Formation of Boxtel DH (BX)'
Formation_Sampling_Time(13) = 'Eem Formation, Bruine Bank DH (EE-BB)'
Formation_Sampling_Time(14) = 'Peelo Formation DH (PE)'
Formation_Sampling_Time(15:31) = 'Formation of Appelscha DH (AP)'
Formation_Sampling_Time(32:33) = 'Eem Formation, Bruine Bank T (EE-BB)'
Formation_Sampling_Time(34) = 'Formation of Drenthe T (DR)'
Formation_Sampling_Time(35:36) = 'Formation of Eem/Urk T (EE-URTY)'
Formation_Sampling_Time(37:47) = 'Formation of Urk, Tyne T (URTY)'
Formation_Sampling_Time(48:55) = 'Valley Fill (val1_b)'
Formation_Sampling_Time(56:58) = 'Formation of Urk, Tyne T (URTY)'
Formation_Sampling_Time(59:69) = 'Peelo Formation, Nieuwolda T (PENI)'

! Based on groups made of the samples resulting from the sieve tests
Formation_ResultGroup(12) = "Group DH 1"
Formation_ResultGroup(13) = "Group DH 2"
Formation_ResultGroup(1:4) = "Group DH 3"
Formation_ResultGroup(14) = "Group DH 3"
Formation_ResultGroup(5:11) = "Group DH 4"
Formation_ResultGroup(15) = "Group DH 5"
Formation_ResultGroup(16:19) = "Group DH 6"
Formation_ResultGroup(20:31) = "Group DH 7"
Formation_ResultGroup(32) = "Group T 1"
Formation_ResultGroup(33:34) = "Group T 2"
Formation_ResultGroup(35:36) = "Group T 3"
Formation_ResultGroup(37:42) = "Group T 4"
Formation_ResultGroup(43) = "Group To Be Done!"
Formation_ResultGroup(44:46) = "Group T 5"
Formation_ResultGroup(47:50) = "Group T 6"
Formation_ResultGroup(51:55) = "Group T 7"
Formation_ResultGroup(56:58) = "Group T 8"
Formation_ResultGroup(59:64) = "Group T 9"
Formation_ResultGroup(65:69) = "Group T 10"

Do i = 1, size(D)/99
  Do j = 1, size(D_V)/99
    Dev = sum((D(i,:)−D_V(j,:))**2)
    DevTot(j) = Dev
  End Do
  !Write(*,'DevTot(j) = ',DevTot(j))
End Do
!Write(*,'(A,13,A,13,A,F)''i, j, Min Dev = ',i,' ',j,' ',minloc(DevTot),'' ',minval(DevTot))

Index1 = i
Allocate(TempArr(2))
TempArr = Minloc(DevTot)
MinValPos = TempArr(1)
Deallocate(TempArr)

If (RunOption == 2) Then
  title = 'Most Similar Samples / Sample_Code(i)'
  Call f2gpsieve2(title,'Diameter [mm]','Summation Percentages by Weight [%]'
Sample_Code,D_Sample_Code_V,D_V,Index1,'1k',MinValPos,'11')
End If
```
DevMinTot(i) = minval(DevTot)

Write(OutputFormSieve, '(4(A,F6.3,6(A))') Sample_Code(i), ' ; ', Formation_Sampling_Time(i), ' ; ', DevMinTot(i), ' ; ', trim(Sample_Code_V(MinValPos)), ' ; ', trim(Formation_V(MinValPos)), ' ; ', trim(Formation_ResultGroup(MinValPos))
!
End Do
!
Making the graphs

Call f2gpSieve('Sievcurve Sampling Time Boxtel Formation', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 12, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Kreftehenye Formation DH', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 13, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Eem Formation DH', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 14, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Urk Formation – Tynje DH', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 15, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Peelo Formation DH', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 16, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Appelscha Formation DH', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 17, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Eem Formation T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 18, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Kreftehenye Formation T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 19, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Eem Formation T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 20, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Urk Formation – Tynje T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 21, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Peelo Formation T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 22, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Appelscha Formation T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 23, '1k', Sample_Code_D)

Write(*, *)
!
If (RunOption == 2) Then
!
Recreates the curves from the vertical samples.

Call f2gpSieve('Sievcurve Texel Naaldwijk V', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 6, '1k', Sample_Code_V_D_V2, '11', '13', '1m', '4', '1n', '5', '1o')

Call f2gpSieve('Sievcurve Texel Eem Bruine Bank V', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 6, '1k', Sample_Code_V_D_V7, '11', '8', '1m', '9', '1n', '10', '1o')

Call f2gpSieve('Sievcurve Texel Eem V', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 6, '1k', Sample_Code_V_D_V12, '11', '13', '1m', '14', '1n')

Call f2gpSieve('Sievcurve Texel Urk Tynje V', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 15, '1k', Sample_Code_V_D_V16, '11', '17', '1m', '18', '1n', '19', '1o')

Write(*, *)
! Making the graphs

Call f2gpSieve('Sievcurve Sampling Time Boxtel Formation', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 12, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Kreftehenye Formation DH', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 13, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Eem Formation DH', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 14, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Urk Formation – Tynje DH', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 15, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Peelo Formation DH', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 16, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Appelscha Formation DH', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 17, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Eem Formation T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 18, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Kreftehenye Formation T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 19, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Eem Formation T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 20, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Urk Formation – Tynje T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 21, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Peelo Formation T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 22, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Appelscha Formation T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 23, '1k', Sample_Code_D)

Write(*, *)
! Making the graphs

Call f2gpSieve('Sievcurve Texel Naaldwijk V', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 6, '1k', Sample_Code_V_D_V2, '11', '13', '1m', '4', '1n', '5', '1o')

Call f2gpSieve('Sievcurve Texel Eem Bruine Bank V', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 6, '1k', Sample_Code_V_D_V7, '11', '8', '1m', '9', '1n', '10', '1o')

Call f2gpSieve('Sievcurve Texel Eem V', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 6, '1k', Sample_Code_V_D_V12, '11', '13', '1m', '14', '1n')

Call f2gpSieve('Sievcurve Texel Urk Tynje V', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 15, '1k', Sample_Code_V_D_V16, '11', '17', '1m', '18', '1n', '19', '1o')

Write(*, *)
! Making the graphs

Call f2gpSieve('Sievcurve Sampling Time Boxtel Formation', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 12, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Kreftehenye Formation DH', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 13, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Eem Formation DH', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 14, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Urk Formation – Tynje DH', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 15, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Peelo Formation DH', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 16, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Appelscha Formation DH', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 17, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Eem Formation T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 18, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Kreftehenye Formation T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 19, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Eem Formation T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 20, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Urk Formation – Tynje T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 21, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Peelo Formation T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 22, '1k', Sample_Code_D)

Call f2gpSieve('Sievcurve Sampling Time Appelscha Formation T', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 23, '1k', Sample_Code_D)

Write(*, *)
! Making the graphs

Call f2gpSieve('Sievcurve Texel Naaldwijk V', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 6, '1k', Sample_Code_V_D_V2, '11', '13', '1m', '4', '1n', '5', '1o')

Call f2gpSieve('Sievcurve Texel Eem Bruine Bank V', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 6, '1k', Sample_Code_V_D_V7, '11', '8', '1m', '9', '1n', '10', '1o')

Call f2gpSieve('Sievcurve Texel Eem V', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 6, '1k', Sample_Code_V_D_V12, '11', '13', '1m', '14', '1n')

Call f2gpSieve('Sievcurve Texel Urk Tynje V', 'Diameter [nm]', 'Summation Percentages by Weight [%]', 15, '1k', Sample_Code_V_D_V16, '11', '17', '1m', '18', '1n', '19', '1o')

Write(*, *)
```
```
Call f2gSieve('Sieve curve Den Helder Naaldwijk V', 'Diameter [mm]', 'Summation Percentages by Weight [%]', 20, '1k', Sample_Code_V, D_V)

Call f2gSieve('Sieve curve Den Helder Boxtel V', 'Diameter [mm]', 'Summation Percentages by Weight [%]', 21, '1k', Sample_Code_V, D_V)

Call f2gSieve('Sieve curve Den Helder Eem Bruine Bank V', 'Diameter [mm]', 'Summation Percentages by Weight [%]', 22, '1k', Sample_Code_V, D_V, 23, '1l', 24, '1m')

Call f2gSieve('Sieve curve Den Helder Eem V', 'Diameter [mm]', 'Summation Percentages by Weight [%]', 25, '1k', Sample_Code_V, D_V, 26, '1l', 27, '1m', 28, '1n', 29, '1o')

Call f2gSieve('Sieve curve Den Helder Urk Tynje V', 'Diameter [mm]', 'Summation Percentages by Weight [%]', 30, '1k', Sample_Code_V, D_V, 31, '1l', 32, '1m')

Call f2gSieve('Sieve curve Den Helder Peelo V', 'Diameter [mm]', 'Summation Percentages by Weight [%]', 33, '1k', Sample_Code_V, D_V, 34, '1l', 35, '1m', 36, '1n')

Call f2gSieve('Sieve curve Den Helder Urk V', 'Diameter [mm]', 'Summation Percentages by Weight [%]', 37, '1k', Sample_Code_V, D_V)

Call f2gSieve('Sieve curve Den Helder Appelscha V', 'Diameter [mm]', 'Summation Percentages by Weight [%]', 38, '1k', Sample_Code_V, D_V, 39, '1l', 40, '1m', 41, '1n')

Call f2gSieve('Sieve curve Den Helder Peize V', 'Diameter [mm]', 'Summation Percentages by Weight [%]', 42, '1k', Sample_Code_V, D_V, 42, '1l')

End If

! Closes this Option

Close(SieveResults)
! Close(SieveResults_Old)
Close(SieveResults_V)
Close(OutputFormSieve)

Deallocate(Sample_Code, D, Sample_Code_V, D_V)
Deallocate(DevTot)
Deallocate(DevMinTot)
Deallocate(Formation_V)
Deallocate(Formation_Sampling_Time)
Deallocate(Char_NA_T, Char_NA_DH)
Deallocate(Char_EEBB_T, Char_EEBB_DH)
Deallocate(Char_URTY_T, Char_URTY_DH)
Deallocate(Char_EE_T, Char_EE_DH)
Deallocate(Char_URK_DH)
Deallocate(Char_PZ_DH)
Deallocate(Char_PE_DH)
Deallocate(Char_BX_DH)
Deallocate(Char_AP_DH)

Case(4)
  ! Creates a single graph of the results of the different scenarios for one sample.

Open(SieveResults, file = filefolder_in/"SieveResultsV2.csv")
! This is read to get the Sample_Code vector, sieve data is not used.
Call ReadingFileSieveV2(SieveResults, Sample_Code, D)
No_Samples = Linecount(SieveResults)-1  ! -1 for headers

Do i=1, No_Samples
  Write(*, '(I2.2, A, I2.2)') i, "/", No_Samples
  ID = Sample_Code(i)
  title = 'Comparison Different Scenarios' // trim(ID)
  Call f2gScenarios(ID, title, 'Date Time', 'Length [m]', '2d', '2e', '2f', '2g', '2h', '2i', '2j', '2k', '2l', '2m')
  ! Data is read in the subroutine, so not data is passed to the subroutine.
End Do

Close(SieveResults)
Deallocate(Sample_Code)
```
Case (5) ! Combines results for the same calculation from the different scenarios.

Do k = 1, 2 ! Two files containing different types of data
  If (k == 1) Then
    FileContent = '_OutputSamplesLoc_Scenario_'
  Else If (k == 2) Then
    FileContent = '_TransportPropTex_Scenario_'
  End If

Do j = 1, 2 ! Two locations
  If (j == 1) Then
    Location = 'Texel'
  Else If (j == 2) Then
    Location = 'Den Helder'
  End If

Do i = 1, 9
  ! Reading files
  Write (Filename, '(A, A, I2.2, A)') trim (Short (Trim (Location))), trim (FileContent), i, '_.txt'
  Open (InputPart, file = filefolder_out//trim (Filename))

  If (k == 1) Then
    Call ReadingFileOutputTxt(Location, InputPart, Var, SampleCode, DateTimeSamples, Var1, Var2, Var3, Var4, Var5, Var6)
    ! For sample locations: Var1 = SamplingAwayTot, Var2 = SamplingDepthTot, Var3 = SamplingRightTot
    Var4 = SourceAwayTot, Var5 = SourceDepthTot, Var6 = SourceRightTot
  Else If (k == 2) Then
    Call ReadingFileOutputTxt(Location, InputPart, Var, SampleCode, DateTimeSamples, Var1, Var2, Var3, Var4, Var5, Var6, Var7, Var8, Var9, Var10, Var11, Var12, Var13, Var14, Var15, Var16, Var17, Var18)
    ! For transport properties: Var1 = MaxPLugR, Var2 = AverageV, Var3 = D_Max, Var4 = Re_BH, Var5 = Re_HB, Var6 = Re_HB2, Var7 = Shear_Rate1, Var8 = Shear_Rate2
    Var9 = Eff Viscosity, Var10 = RouseNo, Var11 = RouseNo2, Var12 = Re_Gen_BH, Var13 = Re_Gen_HB, Var14 = D90, Var15 = Dmax BH Min, Var16 = Dmax BH Max
    Var17 = Dmax HB Min, Var18 = Dmax HB Max
  End If

  If (i == 1) Then
    Allocate (Var1_Scenario01(size (SampleCode)))
    Allocate (Var2_Scenario01(size (SampleCode)))
    Allocate (Var3_Scenario01(size (SampleCode)))
    Allocate (Var4_Scenario01(size (SampleCode)))
    Allocate (Var5_Scenario01(size (SampleCode)))
    Allocate (Var6_Scenario01(size (SampleCode)))
    If (Allocated(Var7)) Allocate(Var7_Scenario01(size (SampleCode)))
    If (Allocated(Var8)) Allocate(Var8_Scenario01(size (SampleCode)))
    If (Allocated(Var9)) Allocate(Var9_Scenario01(size (SampleCode)))
    If (Allocated(Var10)) Allocate(Var10_Scenario01(size (SampleCode)))
    If (Allocated(Var11)) Allocate(Var11_Scenario01(size (SampleCode)))
    If (Allocated(Var12)) Allocate(Var12_Scenario01(size (SampleCode)))
    If (Allocated(Var13)) Allocate(Var13_Scenario01(size (SampleCode)))
    If (Allocated(Var14)) Allocate(Var14_Scenario01(size (SampleCode)))
    If (Allocated(Var15)) Allocate(Var15_Scenario01(size (SampleCode)))
    If (Allocated(Var16)) Allocate(Var16_Scenario01(size (SampleCode)))
    If (Allocated(Var17)) Allocate(Var17_Scenario01(size (SampleCode)))
    If (Allocated(Var18)) Allocate(Var18_Scenario01(size (SampleCode)))

    Var1_Scenario01 = Var1
    Var2_Scenario01 = Var2
    Var3_Scenario01 = Var3
    Var4_Scenario01 = Var4
    Var5_Scenario01 = Var5
    Var6_Scenario01 = Var6
    If (Allocated(Var7)) Var7_Scenario01 = Var7
    If (Allocated(Var8)) Var8_Scenario01 = Var8
    If (Allocated(Var9)) Var9_Scenario01 = Var9
If (Allocated (Var10)) Var10_Scenario01 = Var10
If (Allocated (Var11)) Var11_Scenario01 = Var11
If (Allocated (Var12)) Var12_Scenario01 = Var12
If (Allocated (Var13)) Var13_Scenario01 = Var13
If (Allocated (Var14)) Var14_Scenario01 = Var14
If (Allocated (Var15)) Var15_Scenario01 = Var15
If (Allocated (Var16)) Var16_Scenario01 = Var16
If (Allocated (Var17)) Var17_Scenario01 = Var17
If (Allocated (Var18)) Var18_Scenario01 = Var18

Else If (i == 2) Then
Allocate (Var1_Scenario02 (size (SampleCode)))
Allocate (Var2_Scenario02 (size (SampleCode)))
Allocate (Var3_Scenario02 (size (SampleCode)))
Allocate (Var4_Scenario02 (size (SampleCode)))
Allocate (Var5_Scenario02 (size (SampleCode)))
Allocate (Var6_Scenario02 (size (SampleCode)))
Allocate (Var7_Scenario02 (size (SampleCode)))
Allocate (Var8_Scenario02 (size (SampleCode)))
Allocate (Var9_Scenario02 (size (SampleCode)))
Allocate (Var10_Scenario02 (size (SampleCode)))
Allocate (Var11_Scenario02 (size (SampleCode)))
Allocate (Var12_Scenario02 (size (SampleCode)))
Allocate (Var13_Scenario02 (size (SampleCode)))
Allocate (Var14_Scenario02 (size (SampleCode)))
Allocate (Var15_Scenario02 (size (SampleCode)))
Allocate (Var16_Scenario02 (size (SampleCode)))
Allocate (Var17_Scenario02 (size (SampleCode)))
Allocate (Var18_Scenario02 (size (SampleCode)))

Var1_Scenario02 = Var1
Var2_Scenario02 = Var2
Var3_Scenario02 = Var3
Var4_Scenario02 = Var4
Var5_Scenario02 = Var5
Var6_Scenario02 = Var6
If (Allocated (Var7)) Var7_Scenario02 = Var7
If (Allocated (Var8)) Var8_Scenario02 = Var8
If (Allocated (Var9)) Var9_Scenario02 = Var9
If (Allocated (Var10)) Var10_Scenario02 = Var10
If (Allocated (Var11)) Var11_Scenario02 = Var11
If (Allocated (Var12)) Var12_Scenario02 = Var12
If (Allocated (Var13)) Var13_Scenario02 = Var13
If (Allocated (Var14)) Var14_Scenario02 = Var14
If (Allocated (Var15)) Var15_Scenario02 = Var15
If (Allocated (Var16)) Var16_Scenario02 = Var16
If (Allocated (Var17)) Var17_Scenario02 = Var17
If (Allocated (Var18)) Var18_Scenario02 = Var18

Else If (i == 3) Then
Allocate (Var1_Scenario03 (size (SampleCode)))
Allocate (Var2_Scenario03 (size (SampleCode)))
Allocate (Var3_Scenario03 (size (SampleCode)))
Allocate (Var4_Scenario03 (size (SampleCode)))
Allocate (Var5_Scenario03 (size (SampleCode)))
Allocate (Var6_Scenario03 (size (SampleCode)))
Allocate (Var7_Scenario03 (size (SampleCode)))
Allocate (Var8_Scenario03 (size (SampleCode)))
Allocate (Var9_Scenario03 (size (SampleCode)))
Allocate (Var10_Scenario03 (size (SampleCode)))
Allocate (Var11_Scenario03 (size (SampleCode)))
Allocate (Var12_Scenario03 (size (SampleCode)))
Allocate (Var13_Scenario03 (size (SampleCode)))
Allocate (Var14_Scenario03 (size (SampleCode)))
Allocate (Var15_Scenario03 (size (SampleCode)))
Allocate (Var16_Scenario03 (size (SampleCode)))
Allocate (Var17_Scenario03 (size (SampleCode)))
Allocate (Var18_Scenario03 (size (SampleCode)))

Var1_Scenario03 = Var1
Var2_Scenario03 = Var2
Var3_Scenario03 = Var3
Var4_Scenario03 = Var4
Var5_Scenario03 = Var5
Var6_Scenario03 = Var6

If (Allocated(Var7)) Var7_Scenario03 = Var7
If (Allocated(Var8)) Var8_Scenario03 = Var8
If (Allocated(Var9)) Var9_Scenario03 = Var9
If (Allocated(Var10)) Var10_Scenario03 = Var10
If (Allocated(Var11)) Var11_Scenario03 = Var11
If (Allocated(Var12)) Var12_Scenario03 = Var12
If (Allocated(Var13)) Var13_Scenario03 = Var13
If (Allocated(Var14)) Var14_Scenario03 = Var14
If (Allocated(Var15)) Var15_Scenario03 = Var15
If (Allocated(Var16)) Var16_Scenario03 = Var16
If (Allocated(Var17)) Var17_Scenario03 = Var17
If (Allocated(Var18)) Var18_Scenario03 = Var18

Else If (i == 4) Then
Allocate(Var1_Scenario04(size(SampleCode)))
Allocate(Var2_Scenario04(size(SampleCode)))
Allocate(Var3_Scenario04(size(SampleCode)))
Allocate(Var4_Scenario04(size(SampleCode)))
Allocate(Var5_Scenario04(size(SampleCode)))
Allocate(Var6_Scenario04(size(SampleCode)))
If (Allocated(Var7)) Allocate(Var7_Scenario04(size(SampleCode)))
If (Allocated(Var8)) Allocate(Var8_Scenario04(size(SampleCode)))
If (Allocated(Var9)) Allocate(Var9_Scenario04(size(SampleCode)))
If (Allocated(Var10)) Allocate(Var10_Scenario04(size(SampleCode)))
If (Allocated(Var11)) Allocate(Var11_Scenario04(size(SampleCode)))
If (Allocated(Var12)) Allocate(Var12_Scenario04(size(SampleCode)))
If (Allocated(Var13)) Allocate(Var13_Scenario04(size(SampleCode)))
If (Allocated(Var14)) Allocate(Var14_Scenario04(size(SampleCode)))
If (Allocated(Var15)) Allocate(Var15_Scenario04(size(SampleCode)))
If (Allocated(Var16)) Allocate(Var16_Scenario04(size(SampleCode)))
If (Allocated(Var17)) Allocate(Var17_Scenario04(size(SampleCode)))
If (Allocated(Var18)) Allocate(Var18_Scenario04(size(SampleCode)))

Var1_Scenario04 = Var1
Var2_Scenario04 = Var2
Var3_Scenario04 = Var3
Var4_Scenario04 = Var4
Var5_Scenario04 = Var5
Var6_Scenario04 = Var6

If (Allocated(Var7)) Var7_Scenario04 = Var7
If (Allocated(Var8)) Var8_Scenario04 = Var8
If (Allocated(Var9)) Var9_Scenario04 = Var9
If (Allocated(Var10)) Var10_Scenario04 = Var10
If (Allocated(Var11)) Var11_Scenario04 = Var11
If (Allocated(Var12)) Var12_Scenario04 = Var12
If (Allocated(Var13)) Var13_Scenario04 = Var13
If (Allocated(Var14)) Var14_Scenario04 = Var14
If (Allocated(Var15)) Var15_Scenario04 = Var15
If (Allocated(Var16)) Var16_Scenario04 = Var16
If (Allocated(Var17)) Var17_Scenario04 = Var17
If (Allocated(Var18)) Var18_Scenario04 = Var18

Else If (i == 5) Then
Allocate(Var1_Scenario05(size(SampleCode)))
Allocate(Var2_Scenario05(size(SampleCode)))
Allocate(Var3_Scenario05(size(SampleCode)))
Allocate(Var4_Scenario05(size(SampleCode)))
Allocate(Var5_Scenario05(size(SampleCode)))
Allocate(Var6_Scenario05(size(SampleCode)))
If (Allocated(Var7)) Allocate(Var7_Scenario05(size(SampleCode)))
If (Allocated(Var8)) Allocate(Var8_Scenario05(size(SampleCode)))
If (Allocated(Var9)) Allocate(Var9_Scenario05(size(SampleCode)))
If (Allocated(Var10)) Allocate(Var10_Scenario05(size(SampleCode)))
If (Allocated(Var11)) Allocate(Var11_Scenario05(size(SampleCode)))
If (Allocated(Var12)) Allocate(Var12_Scenario05(size(SampleCode)))
If (Allocated(Var13)) Allocate(Var13_Scenario05(size(SampleCode)))
If (Allocated(Var14)) Allocate(Var14_Scenario05(size(SampleCode)))
If (Allocated(Var15)) Allocate(Var15_scenario05(size(SampleCode)))
If (Allocated(Var16)) Allocate(Var16_scenario05(size(SampleCode)))
If (Allocated(Var17)) Allocate(Var17_scenario05(size(SampleCode)))
If (Allocated(Var18)) Allocate(Var18_scenario05(size(SampleCode)))

Var1_scenario05 = Var1
Var2_scenario05 = Var2
Var3_scenario05 = Var3
Var4_scenario05 = Var4
Var5_scenario05 = Var5
Var6_scenario05 = Var6
If (Allocated(Var7)) Var7_scenario05 = Var7
If (Allocated(Var8)) Var8_scenario05 = Var8
If (Allocated(Var9)) Var9_scenario05 = Var9
If (Allocated(Var10)) Var10_scenario05 = Var10
If (Allocated(Var11)) Var11_scenario05 = Var11
If (Allocated(Var12)) Var12_scenario05 = Var12
If (Allocated(Var13)) Var13_scenario05 = Var13
If (Allocated(Var14)) Var14_scenario05 = Var14
If (Allocated(Var15)) Var15_scenario05 = Var15
If (Allocated(Var16)) Var16_scenario05 = Var16
If (Allocated(Var17)) Var17_scenario05 = Var17
If (Allocated(Var18)) Var18_scenario05 = Var18

Else If (i == 6) Then
Allocate(Var1_scenario06(size(SampleCode)))
Allocate(Var2_scenario06(size(SampleCode)))
Allocate(Var3_scenario06(size(SampleCode)))
Allocate(Var4_scenario06(size(SampleCode)))
Allocate(Var5_scenario06(size(SampleCode)))
Allocate(Var6_scenario06(size(SampleCode)))
If (Allocated(Var7)) Allocate(Var7_scenario06(size(SampleCode)))
If (Allocated(Var8)) Allocate(Var8_scenario06(size(SampleCode)))
If (Allocated(Var9)) Allocate(Var9_scenario06(size(SampleCode)))
If (Allocated(Var10)) Allocate(Var10_scenario06(size(SampleCode)))
If (Allocated(Var11)) Allocate(Var11_scenario06(size(SampleCode)))
If (Allocated(Var12)) Allocate(Var12_scenario06(size(SampleCode)))
If (Allocated(Var13)) Allocate(Var13_scenario06(size(SampleCode)))
If (Allocated(Var14)) Allocate(Var14_scenario06(size(SampleCode)))
If (Allocated(Var15)) Allocate(Var15_scenario06(size(SampleCode)))
If (Allocated(Var16)) Allocate(Var16_scenario06(size(SampleCode)))
If (Allocated(Var17)) Allocate(Var17_scenario06(size(SampleCode)))
If (Allocated(Var18)) Allocate(Var18_scenario06(size(SampleCode)))

Var1_scenario06 = Var1
Var2_scenario06 = Var2
Var3_scenario06 = Var3
Var4_scenario06 = Var4
Var5_scenario06 = Var5
Var6_scenario06 = Var6
If (Allocated(Var7)) Var7_scenario06 = Var7
If (Allocated(Var8)) Var8_scenario06 = Var8
If (Allocated(Var9)) Var9_scenario06 = Var9
If (Allocated(Var10)) Var10_scenario06 = Var10
If (Allocated(Var11)) Var11_scenario06 = Var11
If (Allocated(Var12)) Var12_scenario06 = Var12
If (Allocated(Var13)) Var13_scenario06 = Var13
If (Allocated(Var14)) Var14_scenario06 = Var14
If (Allocated(Var15)) Var15_scenario06 = Var15
If (Allocated(Var16)) Var16_scenario06 = Var16
If (Allocated(Var17)) Var17_scenario06 = Var17
If (Allocated(Var18)) Var18_scenario06 = Var18

Else If (i == 7) Then
Allocate(Var1_scenario07(size(SampleCode)))
Allocate(Var2_scenario07(size(SampleCode)))
Allocate(Var3_scenario07(size(SampleCode)))
Allocate(Var4_scenario07(size(SampleCode)))
Allocate(Var5_scenario07(size(SampleCode)))
Allocate(Var6_scenario07(size(SampleCode)))
If (Allocated(Var7)) Allocate(Var7_scenario07(size(SampleCode)))
Else If (i == 9) Then

Allocate (Var8_Scenario01(size(SampleCode)))
Allocate (Var9_Scenario01(size(SampleCode)))
Allocate (Var10_Scenario01(size(SampleCode)))
Allocate (Var11_Scenario01(size(SampleCode)))
Allocate (Var12_Scenario01(size(SampleCode)))
Allocate (Var13_Scenario01(size(SampleCode)))
Allocate (Var14_Scenario01(size(SampleCode)))
Allocate (Var15_Scenario01(size(SampleCode)))
Allocate (Var16_Scenario01(size(SampleCode)))
Allocate (Var17_Scenario01(size(SampleCode)))
Allocate (Var18_Scenario01(size(SampleCode)))

Var1_Scenario01 = Var1
Var2_Scenario01 = Var2
Var3_Scenario01 = Var3
Var4_Scenario01 = Var4
Var5_Scenario01 = Var5
Var6_Scenario01 = Var6

End If

Else If (i == 8) Then

Allocate (Var1_Scenario07(size(SampleCode)))
Allocate (Var2_Scenario07(size(SampleCode)))
Allocate (Var3_Scenario07(size(SampleCode)))
Allocate (Var4_Scenario07(size(SampleCode)))
Allocate (Var5_Scenario07(size(SampleCode)))
Allocate (Var6_Scenario07(size(SampleCode)))

End If

Var1_Scenario07 = Var1
Var2_Scenario07 = Var2
Var3_Scenario07 = Var3
Var4_Scenario07 = Var4
Var5_Scenario07 = Var5
Var6_Scenario07 = Var6

Else If (i == 8) Then

Allocate (Var1_Scenario08(size(SampleCode)))
Allocate (Var2_Scenario08(size(SampleCode)))
Allocate (Var3_Scenario08(size(SampleCode)))
Allocate (Var4_Scenario08(size(SampleCode)))
Allocate (Var5_Scenario08(size(SampleCode)))
Allocate (Var6_Scenario08(size(SampleCode)))

End If

Var1_Scenario08 = Var1
Var2_Scenario08 = Var2
Var3_Scenario08 = Var3
Var4_Scenario08 = Var4
Var5_Scenario08 = Var5
Var6_Scenario08 = Var6

Else If (i == 9) Then

End If
Allocate (Var1_Scenario09 (size (SampleCode)))
Allocate (Var2_Scenario09 (size (SampleCode)))
Allocate (Var3_Scenario09 (size (SampleCode)))
Allocate (Var4_Scenario09 (size (SampleCode)))
Allocate (Var5_Scenario09 (size (SampleCode)))
Allocate (Var6_Scenario09 (size (SampleCode)))
If (Allocated (Var7)) Allocate (Var7_Scenario09 (size (SampleCode)))
If (Allocated (Var8)) Allocate (Var8_Scenario09 (size (SampleCode)))
If (Allocated (Var9)) Allocate (Var9_Scenario09 (size (SampleCode)))
If (Allocated (Var10)) Allocate (Var10_Scenario09 (size (SampleCode)))
If (Allocated (Var11)) Allocate (Var11_Scenario09 (size (SampleCode)))
If (Allocated (Var12)) Allocate (Var12_Scenario09 (size (SampleCode)))
If (Allocated (Var13)) Allocate (Var13_Scenario09 (size (SampleCode)))
If (Allocated (Var14)) Allocate (Var14_Scenario09 (size (SampleCode)))
If (Allocated (Var15)) Allocate (Var15_Scenario09 (size (SampleCode)))
If (Allocated (Var16)) Allocate (Var16_Scenario09 (size (SampleCode)))
If (Allocated (Var17)) Allocate (Var17_Scenario09 (size (SampleCode)))
If (Allocated (Var18)) Allocate (Var18_Scenario09 (size (SampleCode)))

Var1_Scenario09 = Var1
Var2_Scenario09 = Var2
Var3_Scenario09 = Var3
Var4_Scenario09 = Var4
Var5_Scenario09 = Var5
Var6_Scenario09 = Var6
If (Allocated (Var7)) Var7_Scenario09 = Var7
If (Allocated (Var8)) Var8_Scenario09 = Var8
If (Allocated (Var9)) Var9_Scenario09 = Var9
If (Allocated (Var10)) Var10_Scenario09 = Var10
If (Allocated (Var11)) Var11_Scenario09 = Var11
If (Allocated (Var12)) Var12_Scenario09 = Var12
If (Allocated (Var13)) Var13_Scenario09 = Var13
If (Allocated (Var14)) Var14_Scenario09 = Var14
If (Allocated (Var15)) Var15_Scenario09 = Var15
If (Allocated (Var16)) Var16_Scenario09 = Var16
If (Allocated (Var17)) Var17_Scenario09 = Var17
If (Allocated (Var18)) Var18_Scenario09 = Var18

End If

Close (Inputpart)
Deallocation (SampleCode, DateTimeSamples, Var, Var1, Var2, Var3, Var4, Var5, Var6)
If (Allocated (Var7)) Deallocate (Var7)
If (Allocated (Var8)) Deallocate (Var8)
If (Allocated (Var9)) Deallocate (Var9)
If (Allocated (Var10)) Deallocate (Var10)
If (Allocated (Var11)) Deallocate (Var11)
If (Allocated (Var12)) Deallocate (Var12)
If (Allocated (Var13)) Deallocate (Var13)
If (Allocated (Var14)) Deallocate (Var14)
If (Allocated (Var15)) Deallocate (Var15)
If (Allocated (Var16)) Deallocate (Var16)
If (Allocated (Var17)) Deallocate (Var17)
If (Allocated (Var18)) Deallocate (Var18)

End Do

! Writing Files

Open Inputpart, file = filefolder_out // trim (Filename)
If (k == 1) Then
Call ReadingFileOutputTxt (Location, InputPart, Var, SampleCode, DateTimeSamples, Var1, Var2, Var3, Var4, Var5, Var6)
Else If (k == 2) Then
Call ReadingFileOutputTxt (Location, InputPart, Var, SampleCode, DateTimeSamples, Var1, Var2, Var3, Var4, Var5, Var6, Var7, Var8, Var9, Var10, Var11, Var12, Var13, Var14, Var15, Var16, Var17, Var18)
End If

Close (Inputpart)
If (k == 1) Then
    Write (Filename_Out, '(2(A) )') trim (Short (trim (Location ))) , "_Locations_Scenarios .csv"
    Open (OutputScenarios , file = filefolder_out // Filename_Out)
    Write (OutputScenarios , '(A,27(A,A,A))') trim (Var(1)) , "; trim (Var(6)) , Scenario 1" , "; trim (Var(6)) , Scenario 2" , "; trim (Var(6)) , Scenario 3" , "; trim (Var(6)) , Scenario 4" , "; trim (Var(6)) , Scenario 5" , "; trim (Var(6)) , Scenario 6" , "; trim (Var(6)) , Scenario 7" , "; trim (Var(6)) , Scenario 8" , "; trim (Var(6)) , Scenario 9" , "; trim (Var(7)) , Scenario 1" , "; trim (Var(7)) , Scenario 2" , "; trim (Var(7)) , Scenario 3" , "; trim (Var(7)) , Scenario 4" , "; trim (Var(7)) , Scenario 5" , "; trim (Var(7)) , Scenario 6" , "; trim (Var(7)) , Scenario 7" , "; trim (Var(7)) , Scenario 8" , "; trim (Var(7)) , Scenario 9" , "; trim (Var(8)) , Scenario 1" , "; trim (Var(8)) , Scenario 2" , "; trim (Var(8)) , Scenario 3" , "; trim (Var(8)) , Scenario 4" , "; trim (Var(8)) , Scenario 5" , "; trim (Var(8)) , Scenario 6" , "; trim (Var(8)) , Scenario 7" , "; trim (Var(8)) , Scenario 8" , "; trim (Var(8)) , Scenario 9"
    Do i = 1 , size (SampleCode)
    Write (OutputScenarios , '(A,27(A,F))') SampleCode(i) , "; Var2_Scenario01(i) ;" , "Var4_Scenario01(i) ; Var4_Scenario02(i) ; Var4_Scenario03(i) ; Var4_Scenario04(i) ; Var4_Scenario05(i) ; Var4_Scenario06(i) ; Var4_Scenario07(i) ; Var4_Scenario08(i) ; Var4_Scenario09(i) ; Var5_Scenario01(i) ; Var5_Scenario02(i) ; Var5_Scenario03(i) ; Var5_Scenario04(i) ; Var5_Scenario05(i) ; Var5_Scenario06(i) ; Var5_Scenario07(i) ; Var5_Scenario08(i) ; Var5_Scenario09(i) ; Var6_Scenario01(i) ; Var6_Scenario02(i) ; Var6_Scenario03(i) ; Var6_Scenario04(i) ; Var6_Scenario05(i) ; Var6_Scenario06(i) ; Var6_Scenario07(i) ; Var6_Scenario08(i) ; Var6_Scenario09(i)
    End Do
    Else If (k == 2) Then
    Write (Filename_Out, '(2(A))') trim (Short (trim (Location ))) , "_Transport_1_Scenarios .csv"
    Open (OutputScenarios , file = filefolder_out // Filename_Out)
    Write (OutputScenarios , '(A,27(A,A))') trim (Var(1)) , "; trim (Var(4)) , Scenario 1" , "; trim (Var(4)) , Scenario 2" , "; trim (Var(4)) , Scenario 3" , "; trim (Var(4)) , Scenario 4" , "; trim (Var(4)) , Scenario 5" , "; trim (Var(4)) , Scenario 6" , "; trim (Var(4)) , Scenario 7" , "; trim (Var(4)) , Scenario 8" , "; trim (Var(4)) , Scenario 9" , "; trim (Var(9)) , Scenario 1" , "; trim (Var(9)) , Scenario 2" , "; trim (Var(9)) , Scenario 3" , "; trim (Var(9)) , Scenario 4" , "; trim (Var(9)) , Scenario 5" , "; trim (Var(9)) , Scenario 6" , "; trim (Var(9)) , Scenario 7" , "; trim (Var(9)) , Scenario 8" , "; trim (Var(9)) , Scenario 9" , "; trim (Var(10)) , Scenario 1" , "; trim (Var(10)) , Scenario 2" , "; trim (Var(10)) , Scenario 3" , "; trim (Var(10)) , Scenario 4" , "; trim (Var(10)) , Scenario 5" , "; trim (Var(10)) , Scenario 6" , "; trim (Var(10)) , Scenario 7" , "; trim (Var(10)) , Scenario 8" , "; trim (Var(10)) , Scenario 9"
    Do i = 1 , size (SampleCode)
    Write (OutputScenarios , '(A,27(A,F))') SampleCode(i) , "; Var2_Scenario01(i) ;" , "Var2_Scenario02(i) ; Var2_Scenario03(i) ; Var2_Scenario04(i) ; Var2_Scenario05(i) ; Var2_Scenario06(i) ; Var2_Scenario07(i) ; Var2_Scenario08(i) ; Var2_Scenario09(i) ; Var7_Scenario01(i) ; Var7_Scenario02(i) ; Var7_Scenario03(i) ; Var7_Scenario04(i) ; Var7_Scenario05(i) ; Var7_Scenario06(i) ; Var7_Scenario07(i) ; Var8_Scenario01(i) ; Var8_Scenario02(i) ; Var8_Scenario03(i) ; Var8_Scenario04(i) ; Var8_Scenario05(i) ; Var8_Scenario06(i) ; Var8_Scenario07(i) ; Var8_Scenario08(i) ; Var8_Scenario09(i)
    End Do
End If
Do i = 1, size (SampleCode) 
  Write (OutputScenarios2, '(A,18(A,F))') SampleCode(i), "," , Var14_Scenario01(i), " ; " , Average (Var15_Scenario01(i)) , " ; " , Average (Var13_Scenario01(i)) , " ; " 
  End Do 

Write (Filename_Out, (2(A))) , trim (Short (trim (Location))) , "," , "_Transport_3_Scenarios.csv" 
Open (OutputScenarios3, file = filefolder_out // Filename_Out) 

Do i = 1, size (SampleCode) 
  Write (OutputScenarios3, '(A,37(A,F))') SampleCode(i), "," , Var14_Scenario01(i), " ; " , Var15_Scenario02(i), " ; " , Var15_Scenario03(i), " ; " 
  End Do
Dealocate (SampleCode, DateTimeSamples, Var, Var1, Var2, Var3, Var4, Var5, Var6)
Case (6)

Open (FormationsTNO, file = filefolder_in // "TNOFormations.csv")
Call ReadingFileFormations (FormationsTNO, Sample_Code, Fm_No, Fm)
Do l=1,9
Write(*, '(A)') *
Write(*, *)
ScenarioNo = l
Write (Filename, '(A, I2, A)') "T_OutputSamplesLoc_Scenario_", ScenarioNo, ".txt"
Open (OutputSamplesLoc_T_TXT, file = filefolder_out // Full01-03// trim(Filename))
Location = 'Texel'
Call ReadingFileOutputSamplesLoc.Txt (Location, OutputSamplesLoc_T_TXT, SampleCode_T, 
DateTimeSamples_T, SamplingAwayTot_T, SamplingDepthTot_T, SamplingRightTot_T, SourceAwayTot_T, 
SourceDepthTot_T, SourceRightTot_T)
Write (Filename, '(A, I2, A)') "DH_OutputSamplesLoc_Scenario_", ScenarioNo, ".txt"
Open (OutputSamplesLoc_DH_TXT, file = filefolder_out // Full01-03// trim(Filename))
Location = 'Den Helder'
Call ReadingFileOutputSamplesLoc.Txt (Location, OutputSamplesLoc_DH_TXT, SampleCode_DH, 
DateTimeSamples_DH, SamplingAwayTot_DH, SamplingDepthTot_DH, SamplingRightTot_DH, 
SourceAwayTot_DH, SourceDepthTot_DH, SourceRightTot_DH)
Allocate (FileNames(8))
Do j=1,8
Write (FileNames(j), '(A, I2, A)') "Data_Fm", j, ".dat"
Open(100+j, file = FileNames(j))
End Do
Do i=1, size (Sample_Code)
   Do k=1, size (SampleCode_T)
      If (Sample_Code(i) == SampleCode_T(k)) Then
         Write(100+Fm_No(i), '(A, I1, F, 1x, F)') Sample_Code(i), 100+SourceAwayTot_T(k), 
SourceDepthTot_T(k)
      End If
   End Do
Do m=1, size (SampleCode_DH)
   If (Sample_Code(i) == SampleCode_DH(m)) Then
      Write(100+Fm_No(i), '(A, I1, F, 1x, F)') Sample_Code(i), 4700−SourceAwayTot_DH(m), 
SourceDepthTot_DH(m)
   End If
End Do
End Do
Do $j=1,8$
  Close(100+j)
End Do

BackgroundName = 'CrossSection' ! .png type is required, original cross-section
Write(title, '(A,12,2)') 'Formations Scenario', ScenarioNo
Call 2gpwBGfm(ScenarioNo, BackgroundName, trim(title), 'Sample Code', 'Height [m NAP]',
  FileNames(1), 'Boxtel Fm', '4a', FileNames(2), 'Kreftenheye Fm', '4b', FileNames(3), 'Eem Fm', '4c',
  FileNames(4), 'URTY', '4d', FileNames(5), 'Peelo Fm', '4e', FileNames(6), 'URVE', '4f',
  FileNames(7), 'Appelscha Fm', '4g', FileNames(8), 'Appelscha Weerdinghe', '4h')

BackgroundName = 'CrossSectionBlanco' ! .png type is required, fact map
Write(title, '(A,12,2)') 'Formations Scenario', ScenarioNo
Call 2gpwBGfm(ScenarioNo, BackgroundName, trim(title), 'Sample Code', 'Height [m NAP]',
  FileNames(1), 'Boxtel Fm', '4a', FileNames(2), 'Kreftenheye Fm', '4b', FileNames(3), 'Eem Fm', '4c',
  FileNames(4), 'URTY', '4d', FileNames(5), 'Peelo Fm', '4e', FileNames(6), 'URVE', '4f',
  FileNames(7), 'Appelscha Fm', '4g', FileNames(8), 'Appelscha Weerdinghe', '4h')

BackgroundName = 'CrossSectionV2' ! .png type is required, proposal V2
Write(title, '(A,12,2)') 'Formations Scenario', ScenarioNo
Call 2gpwBGfm(ScenarioNo, BackgroundName, trim(title), 'Sample Code', 'Height [m NAP]',
  FileNames(1), 'Boxtel Fm', '4a', FileNames(2), 'Kreftenheye Fm', '4b', FileNames(3), 'Eem Fm', '4c',
  FileNames(4), 'URTY', '4d', FileNames(5), 'Peelo Fm', '4e', FileNames(6), 'URVE', '4f',
  FileNames(7), 'Appelscha Fm', '4g', FileNames(8), 'Appelscha Weerdinghe', '4h')

BackgroundName = 'CrossSectionV3' ! .png type is required, proposal V3
Write(title, '(A,12,2)') 'Formations Scenario', ScenarioNo
Call 2gpwBGfm(ScenarioNo, BackgroundName, trim(title), 'Sample Code', 'Height [m NAP]',
  FileNames(1), 'Boxtel Fm', '4a', FileNames(2), 'Kreftenheye Fm', '4b', FileNames(3), 'Eem Fm', '4c',
  FileNames(4), 'URTY', '4d', FileNames(5), 'Peelo Fm', '4e', FileNames(6), 'URVE', '4f',
  FileNames(7), 'Appelscha Fm', '4g', FileNames(8), 'Appelscha Weerdinghe', '4h')

Do $j=1,8$
  Write(FileNames(j), '(A, I2,2,A)') 'Data_Fm', j, '.dat'
  Command = 'del' // trim(FileNames(j))
  Call SYSTEM(trim(Command))
End Do

Deallocate(FileNames)
Dealocate(DateAimeSamples_T, SamplingAwayTot_T, SamplingDepthTot_T, SamplingRightTot_T, T,
  SamplingAwayTot_DH, SamplingDepthTot_DH, SamplingRightTot_DH, T)
Dealocate(SampleCode_DH, DateTimeSamples_DH, SamplingAwayTot_DH, SamplingDepthTot_DH, SamplingRightTot_DH, T)
Close(OutputSamplesLoc_T_Txt)
Close(OutputSamplesLoc_DH_Txt)
End Do

Close(FormationsTNO)
Dealocate(Sample_Code, Fm_No, Fm)

Case Default
  Write(*, '(A)') 'Unknown Option, program is terminated.'
  Read(*, *)
  Stop
End Select

!---------------------------------------------------------------
Write(*,*)
Write(*, '(A)') "The chosen part of the program is done, to go back to the start type '1' or type '0' to close the program:"
Read(*,*) EndOption

Select Case(EndOption)
Case (0)
   Stop
Case (1)
   Write(*,*)
   Write(*,*(A)') "
   Write(*,*)
   Go to 100
Case Default
   Write(*,*(A)') "Unknown option program is terminated."
   Read(*,*)
   Stop
End Select
End Program MarsdiepVisual

Module VisualModule
   Implicit None
   Contains

   !---------------------------------------------------------------
   ! FUNCTIONS
   !---------------------------------------------------------------
   Integer Function LineCount(File)
   ! Calculates the amount of lines in a file
   ! Based on http://web.utah.edu/thorne/computing/Handy_Fortran_Tricks.pdf, bottom last page
   Implicit None
   Integer :: File,nlines
   Integer :: io
   nlines = 0 ! Number of lines needs a start value
   Do
      Read (File,*,IOSTAT=io)
      If (io > 0) Then
         Write(*,*) "Check input. Something is wrong..."
         Exit
      Else If (io < 0) Then ! lostat is negative for end of file.
         !WRITE(*,*) "End of file is reached." ! No longer necessary once it works.
         Exit
      Else
         nlines = nlines+1
      End If
   End Do
   Rewind(File) ! Reading of the data starts at the top of the file again.
   Linecount = nlines
   End Function

   !---------------------------------------------------------------
   Integer Function ColumnCount(File,Separator)

Counts the number of columns in a file by counting the separators.

Assumed is that the file has the same number of columns for every line and that the lines are separated by a symbol which is not equal to the decimal or thousand sign and not equal to the symbol for empty spaces if those are present.

Based on [http://rosetacode.org/wiki/Count_occurrences_of_a_substring#Fortran](http://rosetacode.org/wiki/Count_occurrences_of_a_substring#Fortran)

```fortran
Implicit None

Integer :: File
Integer :: nSeparators
Character(*) :: Separator
Character(2048) :: Line
Integer :: Pos, posmin

Read(File, '(A)') Line

nSeparators = 0
Posmin = 1

Do ! Infinite loop, loop runs till the command is reached that closes it.
    Pos = Index(Line(posmin:), Separator)
    If (Pos == 0) Then ! EOF is reached, index returns 0 if substring is not found.
        ColumnCount = nSeparators + 1 ! nColumns is actually the number of separators.
        Rewind(File) ! Reading of the data starts at the top of the file again.
        Return ! Closes the do-loop and thus the function.
    End If
    nSeparators = nSeparators + 1
    posmin = posmin + pos + len(Separator)
End Do

End Function

Character(100) Function Short(Long)

! Enables a shortcut to abbreviated words.

Implicit None

Character(*) :: Long

Select Case(Long)

Case('Den Helder')
    Short = 'DH'

Case('Texel')
    Short = 'T'

Case Default
    Write(*,'(A)') "There is no programmed short for this word."
End Select
End Function

Real Function Average(Vector)

! Calculates the average of a vector, excluding 0

Implicit None

Real :: Vector(:)

Average = Sum(Vector) / Count(Vector > 0)

End Function
```
SUBROUTINES

Reading in

Subroutine ReadingFileCSV_Sur(File, Nlines_sur, DateTime_Sur, Joint, CL, MD, InclinationRaw, AzimuthRaw, Btot, Dip, ElevationCalc, ElevationMGS, Away, R_Calc, R_MGS)

! Containing header, 1st column date (dd-mm-yyyy), 2nd column time (hh:mm), just real numbers the other columns, besides number 11 which contains integers. Columns separated by ",".

Implicit None

Integer :: File
Character(130) :: Header
Integer, allocatable :: Date_I(:)
Real, allocatable :: Time_R(:)
Integer, allocatable :: Joint(:,), Btot(:,)
Real, allocatable :: CL(:,), MD(:,), InclinationRaw(:,), AzimuthRaw(:,)
Real, allocatable :: Dip(:,), ElevationMGS(:,)
Real(:,), allocatable :: ElevationCalc(:,)
Real(:,), allocatable :: R_Calc(:,)
Real(:,), allocatable :: Away(:,)
Real, allocatable :: Dip(:,)
Real, allocatable :: ElevationMGS(:,)

Integer :: nlines, i, io, nlines_data, Nlines_sur
Character(19), Allocatable :: DateTime_Sur(:)

nlines = LINECOUNT(File)
nlines_data = nlines - 1 ! File contains headers.

Write( *, * ) "The Survey file is read and " , nlines_data , " lines containing data are found."

Allocate( Date_I(nlines_data) )
Allocate( Time_R(nlines_data) )
Allocate( Joint(nlines_data) )
Allocate( Btot(nlines_data) )
Allocate( CL(nlines_data) )
Allocate( MD(nlines_data) )
Allocate( InclinationRAW(nlines_data) )
Allocate( AzimuthRaw(nlines_data) )
Allocate( Dip(nlines_data) )
Allocate( ElevationCalc(nlines_data) )
Allocate( ElevationMGS(nlines_data) )
Allocate( Away(nlines_data) )
Allocate( R_Calc(nlines_data) )
Allocate( R_MGS(nlines_data) )

Read(file , '(A100)') Header

Do i=1,nlines_data-1 ! The "-1" accounts for the line read two lines above this sentence

Read(file , '(IS,1X,F15.1,F17.15,1X,1,1,9(F,4.3,F9.4,F)'), IOSTAT=i0) Date_I(i),Time_R(i),Joint(i), CL(i),MD(i),InclinationRaw(i),AzimuthRaw(i),Btot(i),Dip(i),ElevationCalc(i),ElevationMGS(i),Away(i),R_Calc(i),R_MGS(i)

If ( i0 > 0 ) Then
  write( *, * ) "Check input, something is wrong, unit wise..."
  exit
Else if ( i0 < 0 ) Then
  write( *, * ) "End of file is reached, before end of loop. Some lines may contain no data or rubbish."
  Exit
End If

End Do

Nlines_sur = nlines_data-1 ! The "-1" is to account for skipping the second line

Call DateTimeFromSur(DateTime_Sur, Date_I, Time_R, Nlines_sur)
Deallocate (Date_I, Time_R)
End Subroutine

Subroutine ReadingFileSampleData (Location, File, No_Samples, DateTime_Samples, Sample_ID, Temp, EC, pH, MF, Rho, SC, RPM_600, RPM_300, RPM_200, RPM_100, RPM_60, RPM_30, RPM_6, RPM_3, Gel10s, Gel10min)
! Containing header, 1st column date (dd-mm-yy), 2nd column time (hh:mm), just real numbers
the other columns, besides number 11 which contains integers. Columns separated by ",".
Implicit None

Integer :: File
Character (130) :: Header
Integer, Allocatable :: Date_I(:,)
Real, Allocatable :: Time_R(:,)
n integer, io, nlines_data, No_Samples
Character (19), Allocatable :: DateTime_Samples(:,)
Character (len=:) , Allocatable :: Sample_ID(:)
Real, Allocatable :: Temp(:,), EC(:,), pH(:,), MF(:,), Rho(:,), SC(:,)
Integer, Allocatable :: RPM_600(:,), RPM_300(:,), RPM_200(:,), RPM_100(:,), RPM_60(:,), RPM_30(:,), RPM_6(:,), RPM_3(:,), Gel10s(:,), Gel10min(:,)
Character (10) :: Location

nlines = LINECOUNT(File)
nlines_data = nlines-1 ! File contains headers.

Write(*,*)
Write(*, '(A,17,A)') "The Sample file is read and ",nlines_data," lines containing data are found."

Allocate (Date_I(nlines_data))
Allocate (Time_R(nlines_data))

Select Case (Location)
Case ('Den Helder')
Allocate (character (14) :: Sample_ID(nlines_data))
Case ('Texel')
Allocate (character (11) :: Sample_ID(nlines_data))
Case Default
Write(*, '(A)') "Location is unknown, program is terminated."
Pause
Stop
End Select

Allocate (Temp(nlines_data))
Allocate (EC(nlines_data))
Allocate (pH(nlines_data))
Allocate (MF(nlines_data))
Allocate (Rho(nlines_data))
Allocate (SC(nlines_data))
Allocate (RPM_600(nlines_data))
Allocate (RPM_300(nlines_data))
Allocate (RPM_200(nlines_data))
Allocate (RPM_100(nlines_data))
Allocate (RPM_60(nlines_data))
Allocate (RPM_30(nlines_data))
Allocate (RPM_6(nlines_data))
Allocate (RPM_3(nlines_data))
Allocate (Gel10s(nlines_data))
Allocate (Gel10min(nlines_data))

Read(file, '(A100)') Header

Do i=1,nlines_data
   Read(file, '(15,1x,F17.15,1x,A,1x,3(F5.2,1x),F6.2,1x,2(F5.2,1x),10(I))', JOSTAT=io) Date_I(i), Time_R(i), Sample_ID(i), Temp(i), EC(i), pH(i), MF(i), Rho(i), SC(i), RPM_600(i), RPM_300(i),
If (io > 0) Then
  Write(*,*),'Check input, something is wrong, unit wise...'
Exit
Else if (io < 0) Then
  Write(*,*),'End of file is reached, before end of loop. Some lines may contain no data or rubbish.'
Exit
End If
End Do
No_Samples = nlines_data
Call DateTimeFromSur(DateTime_Samples, Date_I, Time_R, No_Samples)
Deallocate(Date_I, Time_R)
End Subroutine

!---------------------------------------------------
Subroutine ReadingFileOutputSamplesLocTxt(Location, File, SampleCode, DateTimeSamples, SamplingAwayTot, SamplingDepthTot, SamplingRightTot, SourceAwayTot, SourceDepthTot, SourceRightTot)
! Reads the results from the samplesource program related to Texel.
Implicit None

Integer :: File
Character(19), Allocatable :: DateTimeSamples(:)
Character(len=:), Allocatable :: SampleCode(:)
Character(:) :: Location
Integer :: nlines, nlines_data, i, io
Integer, Allocatable :: MainI(:)
Real, Allocatable :: SourceLoc_Sur(:,), SourceLoc_PD(:,), SourceLocAv(:,)
Real, Allocatable :: SamplingDepthTot(:,), SamplingAwayTot(:,), SamplingRightTot(:,)
Real, Allocatable :: SourceDepthTot(:,), SourceAwayTot(:,), SourceRightTot(:,)
Real, Allocatable :: PositionBitAll(:)

nlines = Linecount(File)
nlines_data = nlines - 1 ! File contains headers.
Write(*,'(A,I3,A)') 'The output '//trim(Location)//' file is read and ',nlines_data,' lines containing data are found.'
Allocate(DateTimeSamples(nlines_data))
Select Case(Location)
  Case('Den Helder')
    Allocate(character(14) :: SampleCode(nlines_data))
  Case('Texel')
    Allocate(character(11) :: SampleCode(nlines_data))
  Case Default
    Write(*,'(A)') 'Location is unknown, program is terminated.'
    Pause
    Stop
End Select
Allocate(MainI(nlines_data))
Allocate(SourceLoc_Sur(nlines_data))
Allocate(SourceLoc_PD(nlines_data))
Allocate(SourceLocAv(nlines_data))
Allocate(SamplingDepthTot(nlines_data))
Allocate(SamplingAwayTot(nlines_data))
Allocate(SamplingRightTot(nlines_data))
Allocate(SourceDepthTot(nlines_data))
Allocate (SourceAwayTot(nlines_data))
Allocate (SourceRightTot(nlines_data))
Allocate (PositionBitAll(nlines_data))

Read(File,*) ! Skips headers.

Do i = 1, nlines_data
   Read (File, '"(2(A),6(F))"', IOSTAT=io) SampleCode(i), DateTimeSamples(i), SamplingAwayTot(i), SamplingDepthTot(i), SamplingRightTot(i), SourceAwayTot(i), SourceDepthTot(i), SourceRightTot(i)
   If (io > 0) Then
      Write(*, '"(A)"') "Check input, something is wrong, unit wise..."
      Exit
   Else if (io < 0) Then
      Write(*, '"(A)"') "End of file is reached, before end of loop. Some lines may contain no data or rubbish."
      Exit
   End if
End Do

End Subroutine

!---------------------------------------------------------------

Subroutine ReadingFileSieve(File, Sample_Code,D)
! Reads the sieve file, can be generalized to be applicable to files with variable parameter lengths.
Implicit None

Integer :: File
Integer :: nlines, nlines_data, i, io, j
Character(31), Allocatable :: GefFileName(:)
Character(7), Allocatable :: Boring_Code(:)
Character(3), Allocatable :: NEN5104_Code(:)
Character(7), Allocatable :: Sample_No(:)
Character(14), Allocatable :: Depth(:)
Character(1), Allocatable :: lineTemp
Character(2048) :: Line
Integer :: Posmin, Posmax, Pos
Character(40), Allocatable :: Var(:)
Character :: Separator
Integer :: nColumns
Character(2), Allocatable :: Number
Character(7), Allocatable :: Code
Real, Allocatable :: D(:, :)
Integer :: count

nlines = LineCount(File)
nlines_data = nlines - 1 ! File contains headers.
Allocate (D(nlines_data,99)) ! (y,x)

Separator = ","
nColumns = ColumnCount(File,Separator)

Write(*,*) "ColumnCount: ",nColumns
Write(*,*) "The Sieve file is read and ",nlines_data," lines containing data are found."
Allocate (Var(nColumns))
Allocate (GefFileName(nlines_data))
Allocate (Boring_Code(nlines_data))
Allocate (NEN5104_Code(nlines_data))
Allocate (Sample_No(nlines_data))
Allocate (Sample_Code(nlines_data))
Allocate (Depth(nlines_data))
Read(File,'(A)') ! Skips headers
Do j = 1,nLines_data
   Read(File,'(A)') Line
   posmin = 1
   posmax = -1
   Count = 0
   Do i = 1,nColumns
      Pos = index(Line(Posmin:),'*')
      If (Pos == 0) Then
         Var(i) = Line(Posmin:)
         ! Write(*,'(A,14,A,A,A,15,A,15)') "i, Var(i), Posmin, Posmax: ",i, ",", trim(Var(i)),",", Posmin,",",Posmax
         Exit
      End If
      Count = count + 1
      ! If (i == 10) Pause
      If (Count == 5) exit
      Posmax = Posmax + Pos
      Var(i) = Line(Posmin:Posmax)
      Posmin = Posmin + Pos
   End Do
   Read(Var(2),'(A)') Boring_Code(j)
   Read(Var(3),'(A)') NEN5104_Code(j)
   Read(Var(4),'(A)') Sample_No(j)
   ! Giving the gef results the same code as used in the other files.
   Code = Sample_No(j)
   Read(Code,(Len(trim(Code))-1:),'(A2)') Number
   If (Index(Boring_Code(j),'A1') > 0 .and. Index(Boring_Code(j),'DH') > 0) Then
     Sample_Code(j) = "PBIDH-A1--65--"//Number
   Else If (Index(Boring_Code(j),'A3') > 0 .and. Index(Boring_Code(j),'DH') > 0) Then
     Sample_Code(j) = "PBIDH-A3--85--"//Number
   Else If (Index(Boring_Code(j),'B65') > 0 .and. Index(Boring_Code(j),'TXL') > 0) Then
     Sample_Code(j) = "B--65--TXL--"//Number
   Else If (Index(Boring_Code(j),'C85') > 0 .and. Index(Boring_Code(j),'TXL') > 0) Then
     Sample_Code(j) = "C--85--TXL--"//Number
   Else ! The vertical file or other files.
     Sample_Code(j) = trim(Boring_Code(j)) //""//trim(Sample_No(j))
   ! Write(*,*') Sample_Code(j)
   End If
End Do
Rewind(File)
Read(File,'*') ! Skips headers
Do j = 1,nLines_data
   count = 0
   Do While (.TRUE.)
      Read(File,'(A)',Advance='No') lineTemp
      ! Write(*,'(A)') lineTemp
      If (lineTemp=='') count = count + 1
      If (count == 5) exit
   End Do
   Read(File,'*') D(j,:)
End Do
End Subroutine
Subroutine ReadingFileSieveV2 (File, Sample_Code, D)

! Reads the sieve file V2, can be generalized to be applicable to files with variable parameter lengths.

Implicit None

Integer :: File
Integer :: nlines, nlines_data, i, io, j
Character (31), Allocatable :: GetFileName(:)
Character (7), Allocatable :: Boring_Code(:)
Character (3), Allocatable :: NEN5104_Code(:)
Character (7), Allocatable :: Sample_No(:)
Character (14), Allocatable :: Sample_Code(:)
Character (17), Allocatable :: Depth(:)
Character (1), :: lineTemp
Character (2048) :: Line
Character (40), Allocatable :: Var(:)
Character :: Separator
Integer :: nColumns
Character (2) :: Number
Character (7) :: Code
Real, Allocatable :: D(:, :)
Integer :: count
Real, Allocatable :: Mass_Sand(:)
Character (6), Allocatable :: Sieve_Code(:)
Real, Allocatable :: Avg_GrainSize(:)
Character :: Temp
Character (2) :: Temp2

nlines = LineCount (File)
nlines_data = nlines - 1 ! File contains headers.
Allocate (D(nlines_data, 99)) ! (y, x)
Separator = "," nColumns = ColumnCount (File, Separator)
Write (*.*) Write(*, '(A, I7, A)') 'The Sieve file V2 is read and ', nlines_data, ' lines containing data are found.'
Allocate (Var(nColumns))
Allocate (Sieve_Code(nlines_data))
Allocate (Mass_Sand(nlines_data))
Allocate (Avg_GrainSize(nlines_data))
Allocate (Sample_No(nlines_data))
Allocate (Sample_Code(nlines_data))

Read (File, '(A)') ! Skips headers

Do j = 1, nlines_data
    Read (File, '(A)') Line
    posmin = 1
    posmax = -1
    Count = 0
    Do i = 1, 6
        Pos = index (Line (posmin:), ",")
        If (Pos == 0) Then
            Var(i) = Line (posmin:)
            !Write (*, '(A, 14, A, A, I5, A, I5)') ",", Var(i), posmin, posmax: ",", trim (Var(i))
        Else
            posmin = Pos + 1
            posmax = Pos + 1
        End If
    End Do
    Count = count + 1
Posmax = Posmax + Pos
Var(i) = Line(Posmin:Posmax)
Posmin = Posmin + Pos

If (Count == 4) exit
End Do

Read(Var(1),'(A)') Sieve_Code(j)
Read(Var(2),'(A)') Sample_Code(j)
Read(Var(3),'(F)') Mass_Sand(j)
Read(Var(4),'(F)') Avg_Grainsize(j)

End Do

Rewind(File)

Read(File,*) ! Skips headers
Do j = 1,nLines_data
count = 0
Do While (.TRUE.)
  Read(File,'(A)') lineTemp
  If (lineTemp == ',') count = count + 1
  If (count == 4) exit
End Do

Read(File,*) D(j,:)
End Do

Deallocation(Var)

End Subroutine

!----------------------------------------

Subroutine ReadingFileFormations(File, Sample_Code, Fm_No, Fm)
! Reads the file with formations assigned by TNO
Implicit None

Integer :: nlines, nlines_data, i, io, IOSTAT, j
Integer :: File
Integer, Allocatable :: Fm_No(:)
Character(100), Allocatable :: Fm(:)
Character(14), Allocatable :: Sample_Code(:)
Character :: Separator
Integer :: nColumns, Count
Character(1) :: lineTemp
Character(2048) :: Line
Integer :: Posmin, Posmax, Pos
Character(40), Allocatable :: Var(:)

nlines = LineCount(File)
nlines_data = nlines - 1 ! File contains headers.
Separator = ","
nColumns = ColumnCount(File,Separator)

Write(*,*)
Write(*,'(A,I3,A)') 'The Formation TNO file is read and ',nlines_data,' lines containing data are found.'
Allocate(Fm_No(nlines_data))
Allocate(Fm(nlines_data))
Allocate(Sample_Code(nlines_data))
Allocate(Var(nColumns))

Read(File,*) ! Skips headers.

Do j = 1,nLines_data
  Read(File,'(A)') Line
posmin = 1
posmax = -1
Count = 0

Do i = 1, 6
  Pos = index(Line(Posmin:), ",")
  If (Pos == 0) Then
    Var(i) = Line(Posmin:)
    Write(*, '(A, I4, A, A, A, I5, A, I5)') "i, Var(i), Posmin, Posmax: ", ", ", trim(Var(i))
  End If
  Count = count + 1
  Posmax = Posmax + Pos
  Var(i) = Line(Posmin: Posmax)
  Posmin = Posmin + Pos
  If (Count == 2) exit
End Do

Read(Var(1), '(A)') Sample_Code(j)
Read(Var(2), '(I)') Fm_No(j)

End Do

Rewind(File)

Read(File, *) ! Skips headers
Do j = 1, nLines_data
  count = 0
  Do While (.TRUE.)
    Read(File, '(A)', Advance = 'No') lineTemp
    If (lineTemp == ",") count = count + 1
    If (count == 2) exit
  End Do
  Read(File, *) Fm(j)
End Do

Deallocate(Var)

End Subroutine

Subroutine ReadingFileOutputTxt(Location, File, Var_SampleCode, DateTimeSamples, Col01, Col02, 
  Col03, Col04, Col05, Col06, Col07, Col08, Col09, Col10, Col11, Col12, Col13, Col14, Col15, Col16, Col17, 
  Col18)
  ! Generalized reading subroutine, read file with header and sample code in the 1st column, 
  datetime sample in the second column, other columns are unspecified but of the real type.
  Implicit None

  Integer :: File
  Character(19) , Allocatable :: DateTimeSamples(:)
  Character(len=:) , Allocatable :: SampleCode(:)
  Character(*) :: Location
  Integer :: nlines, nlines_data, i, io
  Real, Allocatable :: Col01(:,), Col02(:,), Col03(:,), Col04(:,), Col05(:,), Col06(:,)
  Real, Optional, Allocatable :: Col07(:,), Col08(:,), Col09(:,), Col10(:,), Col11(:,), Col12(:,)
  Real, Optional, Allocatable :: Col13(:,), Col14(:,), Col15(:,), Col16(:,), Col17(:,), Col18(:,)
  Character :: Separator
  Integer :: nColumns, Count
  Character(1) :: lineTemp
  Character(2048) :: Line
  Integer :: Posmin, Posmax, Pos
Character (40), Allocatable :: Var(:)
nlines = Linecount(File)
nlines_data = nlines - 1 ! File contains headers.
Separator = "," nColumns = ColumnCount(File, Separator)

Select Case(Location)
Case ("Den Helder")
  Allocate(character(14) :: SampleCode(nlines_data))
Case ("Texel")
  Allocate(character(11) :: SampleCode(nlines_data))
Case Default
  Write(*, '(A)') "Location is unknown, program is terminated."
  Pause
  Stop
End Select

Allocate(DateTimeSamples(nlines_data))
Allocate(Var(nColumns))
Allocate(Col01(nlines_data))
Allocate(Col02(nlines_data))
Allocate(Col03(nlines_data))
Allocate(Col04(nlines_data))
Allocate(Col05(nlines_data))
Allocate(Col06(nlines_data))
If(Present(Col07)) Allocate(Col07(nlines_data))
If(Present(Col08)) Allocate(Col08(nlines_data))
If(Present(Col09)) Allocate(Col09(nlines_data))
If(Present(Col10)) Allocate(Col10(nlines_data))
If(Present(Col11)) Allocate(Col11(nlines_data))
If(Present(Col12)) Allocate(Col12(nlines_data))
If(Present(Col13)) Allocate(Col13(nlines_data))
If(Present(Col14)) Allocate(Col14(nlines_data))
If(Present(Col15)) Allocate(Col15(nlines_data))
If(Present(Col16)) Allocate(Col16(nlines_data))
If(Present(Col17)) Allocate(Col17(nlines_data))
If(Present(Col18)) Allocate(Col18(nlines_data))

Read(File, '(A)') Line
posmin = 1
posmax = -1
Count = 0
Do i=1, nColumns
  Pos = index(Line(posmin:), ",")
  If (Pos == 0) Then
    Var(i) = Line(posmin:)
    !Write(*, '(A,14,A,A,15,A,15)') 'i', Var(i), posmin, posmax: ",","", trim(Var(i)), ",", posmin,"", ",", posmax
    Exit
  End If
  Count = count + 1
  Posmax = Posmax + Pos
  Var(i) = Line(posmin: posmax)
  Posmin = posmin + Pos
End Do
Do i = 1, nlines_data
  If (Present(Col18)) Then
    Read(File, "+JOSTAT=io) SampleCode(i), DateTimeSamples(i), Col01(i), Col02(i), Col03(i), 
    Col04(i), Col05(i), Col06(i), Col07(i), Col08(i), Col09(i), Col10(i), Col11(i), Col12(i), Col13(i)
The conversion from Excel no. to dd/mm/yyyy is partially based on http://www.codeproject.com/Articles/2750/Excel-serial-date-to-Day-Month-Year-and-vise-versa

Subroutine DateTimeFromSur(DateTime_Sur, Date_I, Time_R, Nlines_sur)

! Converts the Date_I en Time_R to one vector, which can be searched and has the same format as the one read from Prodata.
! The conversion from excel no. to dd/mm/yyyy is partially based on http://www.codeproject.com/Articles/2750/Excel-serial-date-to-Day-Month-Year-and-vise-versa

Implicit None
Character(19), Allocatable :: DateTime_Sur(:)
Integer(2), Allocatable :: Day_L(:), Month_L(:), Hours_L(:), Minutes_L(:), Seconds_L(:)
Integer(4), Allocatable :: Year_L(:)
Integer :: i, j, Nlines_sur
Real :: Date_I(:)
Real :: Time_R(:)
Real :: l, n, z, y

Allocate(Day_L(Nlines_sur))
Allocate(Month_L(Nlines_sur))
Allocate(Year_L(Nlines_sur))
Allocate(Hours_L(Nlines_sur))
Allocate(Minutes_L(Nlines_sur))
Allocate(Seconds_L(Nlines_sur))
Allocate(DateTime_Sur(Nlines_sur))

Do i=1,Nlines_Sur

If (Date_I(i) == 60) Then
  Day_L = 29
  Month_L = 2
  Year_L = 1900
End If

Else If (Date_I(i) < 60) Then
  Date_I(i) = Date_I(i) +1
End Else

l = Date_I(i) +68569 +2415019
n = floor(( 4 * l ) / 146097)
l = l - floor(( 146097 * n + 3 ) / 4)
z = floor(( 4000 * ( l + 1 ) ) / 1461001)
l = l - floor(( 1461 * z ) / 4) + 31
y = floor(( 80 * l ) / 2447)
Day_L = l - floor(( 2447 * y ) / 80)
l = floor(y / 11)
Month_L = y + 2 - ( 12 * l )
Year_L = 100 * ( n - 49 ) + z + 1
End If

Hours_L(i) = int(mod(Time_R(i)*24.0,60.0))
Minutes_L(i) = int(mod(Time_R(i)*1440.0,60.0))
Seconds_L(i) = nint(mod(Time_R(i)*86400.0,60.0)) ! Due to possible errors when
taking the only the integer part for seconds, the seconds are rounded towards the nearest
integer.

If (Seconds_L(i) == 60) Then ! Additional security measure, to
  assure realistic values.
  Minutes_L(i) = Minutes_L(i) +1
  Seconds_L(i) = 0
End If

Write(DateTime_Sur(i),'(I2,A1,I2,A1,I4,1X,I2,A1,I2,A1,I2,A1)') Day_L(i),"-",Month_L(i)
  ,"-",Year_L(i),Hours_L(i),":","Minutes_L(i),":","Seconds_L(i)
End Do

End Subroutine

Subroutine ZoomGPHours(DateTimeSource,DateTimeMin,DateTimeMax,Y)
! Calculates the time in seconds to HH:MM:SS
Implicit None
Character(19) :: DateTimeSource,DateTimeMin,DateTimeMax
Read(DateTimeSource, ' ( I2, 1x, I2, 1x, I4, 1x, I2, 1x, I2, 1x, I2 ) ') Day, Month, Year, Hours, Minutes, Seconds

Hours_Min = Hours - 2*Y
Day_Min = Day
Month_Min = Month

If (Hours_Min < 0) Then
Hours_Min = 24 + Hours_min
Day_Min = Day_Min - 1
End If

! Day_Min can become 0 or less so than the month has to be reduced and day_min becomes the
last or the x to last day of that month.
If (Day_Min < 1) Then
If ((Month - 1) == 1 .or. (Month - 1) == 3 .or. (Month - 1) == 5 .or. (Month - 1) == 7 .or. (Month - 1) == 8 .or. (Month - 1) == 10 .or. (Month - 1) == 12) Then
Day_Min = 31 + Day_Min
Month_Min = Month - 1
Else If ((Month - 1) == 4 .or. (Month - 1) == 6 .or. (Month - 1) == 9 .or. (Month - 1) == 11) Then
Day_Min = 30 + Day_Min
Month_Min = Month - 1
Else If ((Month - 1) == 2) Then
Day_Min = 28 + Day_Min
Month_Min = Month - 1
End If
End If

Write(DateTimeMin, ' (2(12,2,A1),14,1x,2(12,2,A1),12,.2) ') Day_Min,"-",Month_Min,"-",Year, Hours_Min,"-",Minutes,"-",Seconds

Hours_Max = Hours + Y
Day_Max = Day
Month_Max = Month

If (Hours_Max > 23) Then
! 23:59 becomes 00:00 the next day
Hours_Max = Hours_Max - 24
Day_Max = Day_Max + 1
End If

If (Day_Max > 31) Then
If ((Month) == 1 .or. (Month) == 3 .or. (Month) == 5 .or. (Month) == 7 .or. (Month) == 8 .or. (Month) == 10 .or. (Month) == 12) Then
Day_Max = Day_Max - 31
Month_Max = Month + 1
Else If ((Month) == 4 .or. (Month) == 6 .or. (Month) == 9 .or. (Month) == 11) Then
Day_Max = Day_Max - 30
Month_Max = Month + 1
Else If ((Month) == 2) Then
Day_Max = Day_Max - 28
Month_Max = Month + 1
End If
Else If (Day_Max > 30) Then
If ((Month) == 4 .or. (Month) == 6 .or. (Month) == 9 .or. (Month) == 11) Then
Day_Max = Day_Max - 30
Month_Max = Month + 1
Else If ((Month) == 2) Then
Day_Max = Day_Max - 28
Month_Max = Month + 1
End If
Else If (Day_Max > 28) Then
If (Month == 2) Then
    Day_Max = Day_Max - 28
    Month_Max = Month + 1
End If
End If

Write (DateTimeMax, '(2(12.2,A1),I4,1x,2(12.2,A1),12.2)') Day_Max,"-",Month_Max,"-",Year,
        Hours_Max,"-",Minutes,"-",Seconds

End Subroutine

! Plotting the data
!
! Subroutine f2gp3D(location, xdata1, ydata1, zdata1, title0, xlabel, ylabel, zlabel, title1)
! Subroutine f2gp3D(Away, R_Calc, ElevationCalc, 'Drilled Paths', 'Away [m]', 'R_{Calculated} [m]
     ', 'Elevation [m]')
Implicit None

Integer :: n1 ! number of data points
Real(8) :: xdata1(:) ! first x data array
Real(8) :: ydata1(:) ! first y data array
Real(8) :: zdata1(:)
Character(len=*) :: xlabel, ylabel, zlabel, title1, title0 ! plot axis labels and
title(s)
Character(10) :: Location
Integer :: i, End1st, End2nd, End3rd, End4th
Character(50) :: graphtitle1, graphtitle2, graphtitle3, graphtitle4

n1 = size(xdata1)

! Write data in files
!
OPEN(10, Access='SEQUENTIAL', File='xyzdata1.dat')
OPEN(11, Access='SEQUENTIAL', File='xyzdata2.dat')
OPEN(12, Access='SEQUENTIAL', File='xyzdata3.dat')
OPEN(13, Access='SEQUENTIAL', File='xyzdata4.dat')
OPEN(14, Access='SEQUENTIAL', File='xyzdata5.dat')

End4th = 0
Do i = 2, n1-1
    If (xdata1(i+1) == 0) Then
        End1st = i +1
        Exit
    End If
    Write (10, '(F9.3,2(1x,F9.3))') xdata1(i), ydata1(i), zdata1(i)
End Do

Do i = End1st, n1-1
    If (xdata1(i+1) < xdata1(i)) Then
        End2nd = i
        Exit
    End If
    Write (11, '(F9.3,2(1x,F9.3))') xdata1(i), ydata1(i), zdata1(i)
End Do

Do i = End2nd+1, n1-1
    If (xdata1(i+1) < xdata1(i)) Then
        End3rd = i
        Exit
    End If
    Write (12, '(F9.3,2(1x,F9.3))') xdata1(i), ydata1(i), zdata1(i)
End Do

Do i = End3rd+1, n1-1
    If (xdata1(i+1) < xdata1(i)) Then

End4th = i
End If
Write(13,'(F9.3,2(1x,F9.3))') xdata1(i),ydata1(i),zdata1(i)
End Do

If (End4th > 0) Then
Do i = End4th+1,n1-1
Write(14,'(F9.3,2(1x,F9.3))') xdata1(i),ydata1(i),zdata1(i)
End do
End If

CLOSE(10, Status='KEEP')
CLOSE(11, Status='KEEP')
CLOSE(12, Status='KEEP')
CLOSE(13, Status='KEEP')
CLOSE(14, Status='KEEP')

! Create gnuplot command file

OPEN(10, ACCESS='SEQUENTIAL', FILE= 'gp.txt')
Write(10,'(A)') 'set terminal png size 2400,1600'
Write(10,'(A)') 'set output '// '/D:/hijnekam/Desktop/Program/Graphs//'trim(title0)//'.png'

Graph Title , title in bold, works
Write(10,'(A)') 'set title font " , 24" //"Bold "//trim(title1)//"'
Write(10,'(A)') 'set xlabel font " , 20" //trim(xlabel)//"'
Write(10,'(A)') 'set ylabel font " , 20" //trim(ylabel)//"'
Write(10,'(A)') 'set zlabel font " , 20" //trim(zlabel)//" offset -1,-0.4'

Select Case(Location)

Case('Texel')
Write(10,'(A)') 'set yrange [10:-10] reverse'
Write(10,'(A)') 'set for [x = 0:1800:200] arrow from x,-10,-90 to x,-10,5 nohead lt 0 lw 2'
Write(10,'(A)') 'set for [y = -10:10:1] arrow from 0,y,-90 to 0,y,5 nohead lt 0 lw 2'
Write(10,'(A)') 'set for [y = -10:10:1] arrow from 0,y,-90 to 1800,y,-90 nohead lt 0 lw 2'
Write(10,'(A)') 'set ytics font ",14" offset 0,-0.5'

Case('Den Helder')
Write(10,'(A)') 'set yrange [10:-10] reverse'
Write(10,'(A)') 'set for [x = 0:2000:200] arrow from x,-10,-90 to x,-10,5 nohead lt 0 lw 2'
Write(10,'(A)') 'set for [y = -10:10:1] arrow from 0,y,-90 to 0,y,5 nohead lt 0 lw 2'
Write(10,'(A)') 'set for [y = -10:10:1] arrow from 0,y,-90 to 2000,y,-90 nohead lt 0 lw 2'
Write(10,'(A)') 'set ytics font ",14" offset 0,-0.5'

End Select

Write(10,'(A)') 'set xtics font ",14"
Write(10,'(A)') 'set ztics font ",14"
Write(10,'(A)') 'set key font ",20" opaque right spacing 2'
Write(10,'(A)') 'set grid xtics ztics lw 2'
Write(10,'(A)') 'set view 70,60'
Write(10,'(A)') 'set xyplane at -90'
Write(10,'(A)') 'set style line 1 lc rgb "#dd181f" lt 1 lw 3 pt 7 ps 1'
Write(10,'(A)') 'set style line 2 lc rgb "dark-blue" lt 1 lw 3 pt 7 ps 1'
Write(10,'(A)') 'set style line 3 lc rgb "#ffa500" lt 1 lw 3 pt 7 ps 2'
Write(10,'(A)') 'set style line 4 lc rgb "dark-green" lt 1 lw 3 pt 7 ps 2'
Select Case (Location)

Case ('Den Helder')
  graphtitle1 = trim(Title1) // 'NAP−85m Jet bit'
graphtitle2 = trim(Title1) // 'NAP−85m MudMotor'
graphtitle3 = trim(Title1) // 'NAP−65m Jet bit'

Case ('Texel')
  graphtitle1 = trim(Title1) // 'NAP−65m Jet bit'
graphtitle2 = trim(Title1) // 'NAP−65m MudMotor'
graphtitle3 = trim(Title1) // 'NAP−85m Jet bit'
graphtitle4 = trim(Title1) // 'NAP−85m Mudmotor'

Case Default
  Write(10, '(A)') "Location unknown, no plot is made."
Read(*,*)
Return

End Select

Write(10, '(A)') 'splot "xyzdata1.dat" using 1:2:3 with lines ls 1 title ""/trim(graphtitle1)"
""/""/xyzdata2.dat" using 1:2:3 with lines ls 2 title ""/trim(graphtitle2)""
""/""/xyzdata3.dat" using 1:2:3 with lines ls 3 title ""/trim(graphtitle3)""
""/""/xyzdata4.dat" using 1:2:3 with lines ls 4 title ""/trim(graphtitle4)"
""/""/xyzdata5.dat" using 1:2:3 with lines ls 1 title ""/trim(graphtitle3)"

CLOSE(10, STATUS='KEEP')

! Plot curve with gnuplot and cleanup files

Call SYSTEM('gnuplot gp.txt')

! Pause ! To check the gnu file before deletion if the graph is not as expected.

Call SYSTEM('del gp.txt')
Call SYSTEM('del xyzdata1.dat')
Call SYSTEM('del xyzdata2.dat')
Call SYSTEM('del xyzdata3.dat')
Call SYSTEM('del xyzdata4.dat')

! Check message
Write(*, '(A)') 'File ' // /// trim(title0) // /// .png' // /// ' is created.'

End Subroutine

! Subroutine f2gpRhData plots the results of the viscometer

Explicit None

Integer :: nl ! number of data points
Character(*) :: xdata(-) ! first x data array
Integer :: ydata1(:), ydata2(:), ydata3(:), ydata4(:), ydata5(:), ydata6(:), ydata7(:), ydata8(:), ydata9(:), ydata10(:)
Character(len=*) :: xlabel, ylabel, title ! plot axis labels and title(s)
Character(10) :: Location
Integer :: i
Character(*) :: graph_title1, graph_title2, graph_title3, graph_title4, graph_title5, graph_title6, graph_title7, graph_title8, graph_title9, graph_title10
Character(2) :: style1, style2, style3, style4, style5, style6, style7, style8, style9, style10
Character(25) :: Layout1, Layout2, Layout3, Layout4, Layout5, Layout6, Layout7,
n1 = size(xdata)

! Write data in files

OPEN(10, Access='SEQUENTIAL', File='xydata1.dat')

Do i = 1, n1
    If (ydata1(i) == 0 . or. ydata2(i) == 0 . or. ydata3(i) == 0 . or. ydata5(i) == 0 . or. ydata6(i) == 0 . or. ydata7(i) == 0 . or. ydata8(i) == 0) Then
        Write(10,*)
    Else
        Write(10,'(A,1x,10(1x,13.3))') xdata(i),i,ydata1(i),ydata2(i),ydata3(i),ydata4(i),ydata5(i),ydata6(i),ydata7(i),ydata8(i),ydata9(i),ydata10(i)
    End If
End Do

CLOSE(10, Status='KEEP')

! Create gnuplot command file

OPEN(10,ACCESS='SEQUENTIAL',FILE='gp.txt')

Write(10,'(A)' )' set terminal png size 2400,800'  ! Sets it to create .png
Write(10,'(A)' )' set output '//@' 'D:hijnekam/Desktop/Program/Graphs/Plot '/@' trim(title)//@' .png'//@' !' Creates .png file and stores it in a different folder. The "@" used is different from the default or fortran ("\")!!!
Write(10,'(A)' )' set title '//@' '{:/Bold //Title//' }'  ! Graph Title, title in bold, works
Write(10,'(A)' )' set xlabel '//@' 'Trim(xlabel)//@' ' '  
Write(10,'(A)' )' set ylabel '//@' 'Trim(ylabel)//@' ' '  
Write(10,'(A)' )' set key Left right outside '  
Write(10,'(A)' )' set xtics rotate by -45'  ! show gridlines in the xy plane
Write(10,'(A)' )' set grid'  

Call GraphStyle(Style1,Layout1)
Call GraphStyle(Style2,Layout2)
Call GraphStyle(Style3,Layout3)
Call GraphStyle(Style4,Layout4)
Call GraphStyle(Style5,Layout5)
Call GraphStyle(Style6,Layout6)
Call GraphStyle(Style7,Layout7)
Call GraphStyle(Style8,Layout8)
Call GraphStyle(Style9,Layout9)
Call GraphStyle(Style10,Layout10)

Write(10,'(A)' )' plot "xydata1.dat" using 2:3:xticlabels(1) with '//@' trim(Layout1)//' title " //TRIM(graph_title1)="/", "xydata1.dat" using 2:4:xticlabels(1) with '//@' trim(Layout2)//' title " //TRIM(graph_title2)="/", "xydata1.dat" using 2:5:xticlabels(1) with '//@' trim(Layout3)//' title " //TRIM(graph_title3)="/", "xydata1.dat" using 2:6:xticlabels(1) with '//@' trim(Layout4)//' title " //TRIM(graph_title4)="/", "xydata1.dat" using 2:7:xticlabels(1) with '//@' trim(Layout5)//' title " //TRIM(graph_title5)="/", "xydata1.dat" using 2:8:xticlabels(1) with '//@' trim(Layout6)//' title " //TRIM(graph_title6)="/", "xydata1.dat" using 2:9:xticlabels(1) with '//@' trim(Layout7)//' title " //TRIM(graph_title7)="/", "xydata1.dat" using 2:10:xticlabels(1) with '//@' trim(Layout8)//' title " //TRIM(graph_title8)="/", "xydata1.dat" using 2:11:xticlabels(1) with '//@' trim(Layout9)//' title " //TRIM(graph_title9)="/", "xydata1.dat" using 2:12:xticlabels(1) with '//@' trim(Layout10)//' title " //TRIM(graph_title10)="/"

CLOSE(10,STATUS='KEEP')

! Plot curve with gnuplot and cleanup files

Call SYSTEM('gnuplot gp.txt')

! Pause ! To check the gnu file before deletion if the graph is not as expected

Call SYSTEM('del gp.txt')
Call SYSTEM('del xydata1.dat')
! Check message
---
Write(*,'(A)') 'File '//'//'Plot '//'trim(Title)//'.png'//''''' is created.'
End Subroutine
---

Subroutine f2gpSamplePar(Location, xdata , ydata1 , style1 , ydata2 , style2 , title , xlabel, ylabel1, ylabel2, graph_title1, graph_title2)
! Plots two sample parameters in one graph with different y-axis.
Implicit None
Character(10) :: Location
Character(*) :: xdata(:)
Real :: ydata1(:), ydata2(:)
Character(len=*) :: xlabel, ylabel1, ylabel2, title
Character(*) :: graph_title1, graph_title2
Character(2) :: style1, style2
Character(25) :: Layout1, Layout2
Integer :: i, n1

n1 = size(xdata)
---
! Write data in files
---
OPEN(10, Access='SEQUENTIAL', File='xydata1.dat')
Do i = 1, n1
   If (ydata1(i) == 0 .or. ydata2(i) == 0) Then
      Write(10,'(A)')
   Else
      Write(10,'(A,1x,1x,F6.3,1x,F6.2)') xdata(i),i,ydata1(i),ydata2(i)
   End If
End Do
CLOSE(10, Status='KEEP')
---
! Create gnuplot command file
---
OPEN(10, ACCESS='SEQUENTIAL', FILE='gp.txt')
Write(10,'(A)') 'set terminal png size 2400,800' ! Sets it to create .png
Write(10,'(A)') 'set output '//'//'D:/hijnekam/Desktop/Program/Graphs/Plot '//'trim(title)//' .png'//'''''
   ! Creates .png file and stores it in a different folder. The '/' used is different from the default or fortran ("\")!!!
Write(10,'(A)') 'set title '//'//'Bold '//'Title //''''
   ! Graph Title, title in bold, works
Write(10,'(A)') 'set xlabel '//'//'Trim(xlabel)//'''''
Write(10,'(A)') 'set ylabel '//'//'Trim(ylabel1)//'''''
Write(10,'(A)') 'set y2label '//'//'TRIM(ylabel2)//'''''
Write(10,'(A)') 'set key Left right outside'
Write(10,'(A)') 'set xtics rotate by -45'
Write(10,'(A)') 'set grid' ! show gridlines in the xy plane
Select Case(title) ! Fixes the axis so that the grid is correct on both axis, this of course can only be done for known cases.
Case ('DH Density vs. Sand Content')
   Write(10,'(A)') 'set ytics 0,0.3,1.5'
   Write(10,'(A)') 'set y2tics 0,0.05,0.25'
   Write(10,'(A)') 'set yrange [0:1.5]'
   Write(10,'(A)') 'set y2range [0:0.25]'
Case ('T Density vs. Sand Content')
   Write(10,'(A)') 'set ytics 0,0.3,1.5'
   Write(10,'(A)') 'set y2tics 0,0.02,0.1'
   Write(10,'(A)') 'set yrange [0:1.5]'
   Write(10,'(A)') 'set y2range [0:0.1]'
End Select
Case ('DH Temperature vs. Marsh Funnel')
  Write(10,*) 'set ytics 0, 0.25'
  Write(10,*) 'set y2tics 0.40, 200'
  Write(10,*) 'set yrange [0:25]'
  Write(10,*) 'set y2range [0:200]'

Case ('T Temperature vs. Marsh Funnel')
  Write(10,*) 'set ytics 0, 0.30'
  Write(10,*) 'set y2tics 0.20, 120'
  Write(10,*) 'set yrange [0:30]'
  Write(10,*) 'set y2range [0:120]'

Case Default
  Write(10,*) 'set ytics 0: *'
  Write(10,*) 'set y2tics 0: *'
  Write(10,*) 'set yrange [0:*]'
  Write(10,*) 'set y2range [0:*]'
  Write(10,*) 'set ytics 0, 5, 25'
  Write(10,*) 'set y2tics 0, 40, 200'
  Write(10,*) 'set yrange [0:25]'
  Write(10,*) 'set y2range [0:200]'
while the upper limit is still flexible.
  Write(10,*) 'set ytics 0, 5, 30'
  Write(10,*) 'set y2tics 0, 20, 120'
  Write(10,*) 'set yrange [0:30]'
  Write(10,*) 'set y2range [0:120]'

Forcing the minimum of the y-axis to be 0
while the upper limit is still flexible.

End Select

Call GraphStyle (Style1, Layout1)
Call GraphStyle (Style2, Layout2)

Write(10, '(A)') 'Graph is not specified, the tics on the axes and its values can look
weird but are correct.'

End Subroutine

! Plot curve with gnuplot and cleanup files
!-------------------------------------------------------------
Call SYSTEM('gnuplot gp.txt')

! Pause ! To check the gnu file before deletion if the graph is not as expected.

Call SYSTEM('del gp.txt')
Call SYSTEM('del xdata1.dat')

! Check message
!-------------------------------------------------------------
Write(10, '(A)') 'File '//trim(graph_title1)///'.png'//''//is created.'

End Subroutine

!-------------------------------------------------------------
Subroutine f2gpwBG(wdata1, xdata1, ydata1, Style1, wdata2, xdata2, ydata2, Style2, title, xlabel, ylabel, graph_title1, graph_title2)
! Implement a picture for the background. Based on amongst others; http://www.gnuplotting.org/tag/image/.
! And plots results of the Sample Source program on top of it.
Implicit None

Character(*) :: wdata1(:), wdata2(:)
Real :: xdata1(:), ydata1(:), xdata2(:, ydata2(:)
Character(2) :: Style1, Style2
Character(25) :: Layout1, Layout2
Integer :: i, n1, n2
Character(50) :: title
Character(len=**) :: xlabel, ylabel, graph_title1, graph_title2

n1 = size(xdata1)

n2 = size(xdata2)

write(*,*) xdata1
OPEN(10, Access = 'SEQUENTIAL', File = 'xydata1.dat')

Do i = 1, n1
    WRITE(10, '(A, 2*(1.x, F7.2))') wdata1(i), 100 + xdata1(i), ydata1(i)
END DO

CLOSE(10, Status = 'KEEP')

OPEN(10, Access = 'SEQUENTIAL', File = 'xydata2.dat')

Do i = 1, n2
    WRITE(10, '(A, 2*(1.x, F7.2))') wdata2(i), 4675 - xdata2(i), ydata2(i)
END DO

CLOSE(10, Status = 'KEEP')

CREATE gnuplot command file

OPEN(10, Access = 'SEQUENTIAL', File = 'gp.txt')

WRITE(10, '(A)') 'set terminal pngcairo size 4800,2400'
WRITE(10, '(A)') 'set output ' '/D:/hijnekam/Desktop/Program/Graphs/Plot_' / / trim(titile) ' .png' '/'
WRITE(10, '(A)') 'set title ' '//trim(titile);// font",.32"'
WRITE(10, '(A)') 'set size 0.9'
WRITE(10, '(A)') 'set origin 0.05,0.05'
WRITE(10, '(A)') 'set xrange [-650 : 6110 ]'
WRITE(10, '(A)') 'set yrange [-153.5 : 32 ]'
WRITE(10, '(A)') 'set xlabel ' '//trim(xlabel);// font",.28" offset -4.0'
WRITE(10, '(A)') 'set ylabel ' '//trim(ylabel);// font",.28"'
WRITE(10, '(A)') 'set grid front lw 3'
WRITE(10, '(A)') 'set key at 5950,-140 Left reverse font ",.24"
WRITE(10, '(A)') 'set x2range [ 0 : * ]!! noreverse nowrnoteback'
WRITE(10, '(A)') 'set y2range [ 0 : * ]!! noreverse nowrnoteback'
WRITE(10, '(A)') 'unset xtics'
WRITE(10, '(A)') 'unset y2tics'

Call GraphStyle(Style1, Layout1)
Call GraphStyle(Style2, Layout2)

WRITE(10, '(A)') 'set lmargin at screen 0.05'
WRITE(10, '(A)') 'set rmargin at screen 0.85'
WRITE(10, '(A)') 'set tmargin at screen 0.05'
WRITE(10, '(A)') 'set bmargin at screen 0.95'
WRITE(10, '(A)') 'plot "Crosssection.png" binary filetype=png using 1:2:3:(225) with rgbalphas no title axes xy2, "xydata1.dat" using 2:3:xticlabels(l) with ' '/trim(Layout1)///' axes xyl title ""'/trim(graphtitle1)//" "xydata2.dat" using 2:3:xticlabels(l) with ' '/trim(Layout2)///' axes xyl title ""'/trim(graphtitle2)//"'

The value between brackets in the 1:2:3:(X) part needs to be between 0 (fully transparent) and 255 (fully opaque).

CLOSE(10, Status = 'KEEP')

Plot curve with gnuplot and cleanup files
Call SYSTEM('gnuplot gp.txt')

! Pause ! To check the gnu file before deletion if the graph is not as expected.

Call SYSTEM('del gp.txt')
Call SYSTEM('del xydata1.dat')
Call SYSTEM('del xydata2.dat')

! Check message

Write(*,'(A)') 'File '//'Plot '//'trim(title)//'.png'//''' is created.'

End Subroutine

Subroutine f2gwpBG2(GraphOption, filename, wdata1, xdata1, ydata1, Style1, xdata3, ydata3, style3,
                        wdata2, xdata2, ydata2, Style2, xdata4, ydata4, style4, title, xlabel, ylabel, graphtitle1,
                        graphtitle2, graphtitle3, graphtitle4)

! Implement a picture for the background. Based on amongst others; http://www.gnuplotting.org/tag/image/.
! And plots results of the Sample Source program on top of it.
Implicit None

Character(*) :: wdata1(:), wdata2(:)
Real :: xdata1(:,), ydata1(:,), xdata2(:,), ydata2(:,)
Real(8) :: xdata3(:,), ydata3(:,), xdata4(:,), ydata4(:,)
Character(2) :: Style1, Style2
Character(2) :: Style3, Style4
Character(25) :: Layout1, Layout2
Character(25) :: Layout3, Layout4
Integer :: i, n1, n2, n3, n4
Character(50) :: title
Character(50) :: GraphOption
Character(len=*) :: xlabel, ylabel, graphtitle1, graphtitle2, graphtitle3, graphtitle4
Integer :: GraphOption

n1 = size(xdata1)
n2 = size(xdata2)
n3 = size(xdata3)
n4 = size(xdata4)

! Write data in files

OPEN(10, Access='SEQUENTIAL', File='xydata1.dat')

Do i = 1, n1
    Write(10, '(A,1x,F7.2,1x,F7.2)') wdata1(i), 100+xdata1(i), ydata1(i) ! The 100 is because the input data in this case is relative to the origin, the origin is at approx. 100m.
End Do

CLOSE(10, Status='KEEP')

OPEN(10, Access='SEQUENTIAL', File='xydata2.dat')

Do i = 1, n2
    Write(10, '(A,1x,F7.2,1x,F7.2)') wdata2(i), 4675-xdata2(i), ydata2(i) ! The 4675 is because the input data in this case is relative to the origin, the origin is at approx. 4675m.
End Do

CLOSE(10, Status='KEEP')

OPEN(10, Access='SEQUENTIAL', File='xydata3.dat')

Write(10, '(F7.2,1x,F7.2)') 100+xdata3(1), ydata3(1)
Do i = 2, n3
  If ( xdata3 ( i ) < xdata3 ( i − 1) ) Write(10, *)
    Write (10, '(F7.2,1x,F7.2)') 100+xdata3 ( i ) , ydata3 ( i ) ! The 100 is because the input
  data in this case is relative to the origin, the origin is at approx. 100m.
End Do
CLOSE(10, Status='KEEP')

OPEN(10, Access='SEQUENTIAL', File= 'xydata4.dat')
Do i = 1, n4
  If ( xdata4 ( i ) < xdata4 ( i − 1) ) Write (10, * )
    Write (10, ' (F7.2,1x,F7.2)') 4675−xdata4 ( i ) , ydata4 ( i ) ! The 4675 is because the
  input data in this case is relative to the origin, the origin is at approx. 4675m.
End Do
CLOSE(10, Status='KEEP')

! Create gnuplot command file

OPEN(10, ACCESS= 'SEQUENTIAL', FILE= 'gp.txt')
Write (10, ' (A) ' ) ' set terminal pngcairo size 4800,2400 ' ! linewidth 2.0 font ",.40"
  size 2000,10000
Write (10, ' (A) ' ) ' set output ' '/D:/hijnekam/Desktop/Program/Graphs/Plot_' / trim (filename ) "/png" "/"
Write (10, ' (A) ' ) ' set title " / trim ( title )/" font ",.32"
Write (10, ' (A) ' ) ' set size 0.9 ' 
Write (10, ' (A) ' ) ' set origin 0.05,0.05 ' 
Write (10, ' (A) ' ) ' set xrange [−650 : 6110 ] ' ! Found manually based on
  the picture used, since it already contains a scale...
Write (10, ' (A) ' ) ' set xlabel ' '/" Trim ( xlabel )" font ",.28"
Write (10, ' (A) ' ) ' set ytics 0.500,5000 out nomirror rotate by −45 ' ! the values are
  OVERWRITTEN due to use of sample names on x–axis, the other options still work.
Write (10, ' (A) ' ) ' set yrange [−153.5 : 32 ] ' ! Idem dito...
Write (10, ' (A) ' ) ' unset ytics ' 
! Write (10, ' (A) ' ) ' set ytics −150,50,50 out nomirror font",.22" ' ! Replaces by arrows since
  the horizontal grid lines went into the key at the side.
Write (10, ' (A) ' ) ' set ylabel ' '/" Trim ( ylabel )"/" font ",.28" offset −4.0'
Write (10, ' (A) ' ) ' set grid front lw 3 ' 
Write (10, ' (A) ' ) ' set key at 6100,−135 Left reverse font ",.24"
Write (10, ' (A) ' ) ' set x2range [ 0 : * ] ' ! noreverse nowriteback '
Write (10, ' (A) ' ) ' set y2range [ 0 : * ] ' ! noreverse nowriteback '
Write (10, ' (A) ' ) ' unset x2tics ' 
Write (10, ' (A) ' ) ' unset y2tics ' 
Call GraphStyle (Style1, Layout1)
Call GraphStyle (Style2, Layout2)
Call GraphStyle (Style3, Layout3)
Call GraphStyle (Style4, Layout4)
Write (10, ' (A) ' ) ' set lmargin at screen 0.05 ' 
! Write (10, ' (A) ' ) ' set tmargin at screen 0.85 ' 
Write (10, ' (A) ' ) ' set bmargin at screen 0.05 ' 
Write (10, ' (A) ' ) ' set tmargin at screen 0.95 ' 
Select Case (GraphOption)
Case (1) ! Head Location sampling time
  Write (10, ' (A) ' ) ' plot "Crosssection.png" binary filetype/png using 1:2:3:(225) with
    rgbalpha notitle axes x2y2. "xydata1.dat" using 2:3:xticlabels (1) with ' '/trim(Layout1)//
    "axes xyl title " '/Trim ( graphtitle1 ) //". "xydata2.dat" using 2:3:xticlabels (1) with ' 
    '/trim(Layout2)//" axes xyl title " '/Trim ( graphtitle2 ) //"'
! The value between brackets in the 1:2:3:(X) part needs to be between 0 (fully
  transparent) and 255 (fully opaque).
Case (3) ! Drilled Paths
Write(10,'(A)') 'plot "Crosssection.png" binary filetype=png using 1:2:3:(225) with
rgbalpha notitle axes x2y2, "xydata3.dat" using 1:2 with '//'trim(Layout3)//' axes xyl
itle ""///TRIM(graphtitle3)//"",'xydata4.dat" using 1:2 with '//'trim(Layout4)//' axes
xyl title ""///TRIM(graphtitle4)//""

Case (4) ! Drilled Paths and Head Location at Sampling Time
Write(10,'(A)') 'plot "Crosssection.png" binary filetype=png using 1:2:3:(225) with
rgbalpha notitle axes x2y2, "xydata1.dat" using 2:3:x tic labels(1) with '//'trim(Layout1)//' axes xyl title ""///TRIM(graphtitle2)//"",'xydata2.dat" using 2:3:x tic labels(1) with '//'trim(Layout2)//' axes xyl title ""///TRIM(graphtitle3)//"",'xydata3.dat" using 1:2 with '//'trim(Layout3)//' axes xyl title ""///TRIM(graphtitle4)//""

Case (5) ! Drilled paths and source location
! Write(10,'(A)') 'plot "Crosssection.png" binary filetype=png using 1:2:3:(225) with
rgbalpha notitle axes x2y2, "xydata3.dat" using 1:2 with '//'trim(Layout3)//' axes xyl
itle ""///TRIM(graphtitle4)//""
End Select

CLOSE(10,STATUS= 'KEEP')

! Plot curve with gnuplot and cleanup files
!--------------------------------------------------------------
Call SYSTEM('gnuplot gp.txt')
!
! Pause ! To check the gnu file before deletion if the graph is not as expected.
!
Call SYSTEM('del gp.txt')
Call SYSTEM('del xydata1.dat')
Call SYSTEM('del xydata2.dat')
Call SYSTEM('del xydata3.dat')
Call SYSTEM('del xydata4.dat')
!
! Check message
!--------------------------------------------------------------
Write(*,'(A)') 'File ""///Plot_""///trim(filenam)"///.png"///""/// is created.'
End Subroutine

!--------------------------------------------------------------
Subroutine f2gpSieve(title, xlabel, ylabel, Index1, Style1, Sample_Code, D, Index2, Style2, Index3, Style3, Index4, Style4, Index5, Style5, Index6, Style6, Index7, Style7, Index8, Style8, Index9, style9, Index10, Style10, Index11, Style11, Index12, Style12, Index13, Style13, Index14, Style14, Index15, Style15, Index16, Style16)
! Plots different sand curves, up to 5 in the same graph.
Implicit None

Character(*) :: title
Character(*) :: xlabel, ylabel
Character(14) :: Sample_Code(:)
Integer :: Index1
Integer, Optional :: Index2, Index3, Index4, Index5, Index6, Index7, Index8, Index9, Index10, Index11, Index12, Index13, Index14, Index15, Index16
!
Optiona International only works for units that are called or exported, internal units are defined in any case. i.e. layout is internal and thus lacks the optional argument.

Character(2) :: Style1
Character(2), Optional :: Style2, Style3, Style4, Style5, Style6, Style7, Style8, Style9, Style10, Style11, Style12, Style13, Style14, Style15, Style16
Character(25) :: Layout1
Real :: D(:,:)
Integer :: i, j

!--------------------------------------------------------------
! Write data in files
OPEN(10, Access='SEQUENTIAL', File='xydata1.dat')

If ((size(D)/99) == 69) Write(10,'(A,69(1x,A))') 'Ydata', Sample_Code(:)
If ((size(D)/99) == 43) Write(10,'(A,43(1x,A))') 'Ydata', Sample_Code(:)

Do j = 1, 99
Write(10,'(1,1x,98(F,1x,F))') j, D(:,j)
End Do

CLOSE(10, Status='KEEP')

! Create gnuplot command file
!---------------------------------------------------------------
OPEN(10, ACCESS='SEQUENTIAL', FILE='gp.txt')
Write(10,'(A)') 'set terminal postscript size 2400,2400' ! linewidth 2.0 font ",40" size 2000,1000
Write(10,'(A)') 'set output ' ' ' '/D:/hijnekam/Desktop/Program/Graphs/SieveCurves/' 'trim(title)'/''
Write(10,'(A)') 'set title ' ' ' '/trim(title)''/'' font ",32''
Write(10,'(A)') 'set xrange [ 0.01 : 10 ]' ! Found manually based on the picture used, since it already contains a scale...
Write(10,'(A)') 'set xtics'
Write(10,'(A)') 'set mxtics'
Write(10,'(A)') 'set xlabel ' ' ' '/trim(xlabel)''/'' font ",28''
Write(10,'(A)') 'set yrange [ 0 : 100 ]' ! Idem dito...
Write(10,'(A)') 'set ytics 0,10,100'
Write(10,'(A)') 'set ylabel ' ' ' '/trim(ylabel)''/'' font ",28'' offset -4.0''
Write(10,'(A)') 'set grid xtics mxtics ytics lc rgb "black" lt 1 lw 1''
Write(10,'(A)') 'set key Left reverse font "",24''
Write(10,'(A)') 'set tmargin at screen 0.95'
Write(10,'(A)') 'set bmargin at screen 0.05'
Write(10,'(A)') 'set rmargin at screen 0.95'
Write(10,'(A)') 'set lmargin at screen 0.05'

Call GraphStyle(Style1,Layout1,Index1)
If (Present(Index2)) Call GraphStyle(Style2,Layout2,Index2)
If (Present(Index3)) Call GraphStyle(Style3,Layout3,Index3)
If (Present(Index4)) Call GraphStyle(Style4,Layout4,Index4)
If (Present(Index5)) Call GraphStyle(Style5,Layout5,Index5)
If (Present(Index6)) Call GraphStyle(Style6,Layout6,Index6)
If (Present(Index7)) Call GraphStyle(Style7,Layout7,Index7)
If (Present(Index8)) Call GraphStyle(Style8,Layout8,Index8)
If (Present(Index9)) Call GraphStyle(Style9,Layout9,Index9)
If (Present(Index10)) Call GraphStyle(Style10,Layout10,Index10)
If (Present(Index11)) Call GraphStyle(Style11,Layout11,Index11)
If (Present(Index12)) Call GraphStyle(Style12,Layout12,Index12)
If (Present(Index13)) Call GraphStyle(Style13,Layout13,Index13)
If (Present(Index14)) Call GraphStyle(Style14,Layout14,Index14)
If (Present(Index15)) Call GraphStyle(Style15,Layout15,Index15)
If (Present(Index16)) Call GraphStyle(Style16,Layout16,Index16)

Write(10,'(A)') 'set log x'
If (Present(Index16)) Then
Write(10,'(A,16(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,,Index16,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else If (Present(Index15)) Then
Write(10,'(A,15(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else If (Present(Index14)) Then
Write(10,'(A,14(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else If (Present(Index13)) Then
Write(10,'(A,13(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else If (Present(Index12)) Then
Write(10,'(A,12(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else If (Present(Index11)) Then
Write(10,'(A,11(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else If (Present(Index10)) Then
Write(10,'(A,10(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else If (Present(Index9)) Then
Write(10,'(A,9(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else If (Present(Index8)) Then
Write(10,'(A,8(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else If (Present(Index7)) Then
Write(10,'(A,7(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else If (Present(Index6)) Then
Write(10,'(A,6(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else If (Present(Index5)) Then
Write(10,'(A,5(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else If (Present(Index4)) Then
Write(10,'(A,4(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else If (Present(Index3)) Then
Write(10,'(A,3(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else If (Present(Index2)) Then
Write(10,'(A,2(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else If (Present(Index1)) Then
Write(10,'(A,1(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
Else
Write(10,'(A,0(13,A))') 'plot for [n in ",Index1,,Index2,,Index3,,Index4,,Index5,,Index6,,Index7,,Index8,,Index9,,Index10,,Index11,,Index12,,Index13,,Index14,,Index15,"] "xydata1.dat" using (column(n+1)):1 with lines ls (n) title columnheader(n+1)'
End If

OPEN(10, Access='SEQUENTIAL', File='xydata1.dat')
End Subroutine

Else If (Present(Index13)) Then
    Write(10,'("A",13(I3,A))') 'plot for [ in "",Index1,"",Index2,"",Index3,"",Index4,"",Index5,"",Index6,"",Index7,"",Index8,"",Index9,"",Index10,"",Index11,"",Index12,"",Index13,"",Index14,""] "xydata1.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)

Else If (Present(Index12)) Then
    Write(10,'("A",12(I3,A))') 'plot for [ in "",Index1,"",Index2,"",Index3,"",Index4,"",Index5,"",Index6,"",Index7,"",Index8,"",Index9,"",Index10,"",Index11,"",Index12,""] "xydata1.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)

Else If (Present(Index11)) Then
    Write(10,'("A",11(I3,A))') 'plot for [ in "",Index1,"",Index2,"",Index3,"",Index4,"",Index5,"",Index6,"",Index7,"",Index8,"",Index9,"",Index10,"",Index11,""] "xydata1.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)

Else If (Present(Index10)) Then
    Write(10,'("A",10(I3,A))') 'plot for [ in "",Index1,"",Index2,"",Index3,"",Index4,"",Index5,"",Index6,"",Index7,"",Index8,"",Index9,"",Index10,""] "xydata1.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)

Else If (Present(Index9)) Then
    Write(10,'("A",9(I3,A))') 'plot for [ in "",Index1,"",Index2,"",Index3,"",Index4,"",Index5,"",Index6,"",Index7,"",Index8,"",Index9,""] "xydata1.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)

Else If (Present(Index8)) Then
    Write(10,'("A",8(I3,A))') 'plot for [ in "",Index1,"",Index2,"",Index3,"",Index4,"",Index5,"",Index6,"",Index7,"",Index8,""] "xydata1.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)

Else If (Present(Index7)) Then
    Write(10,'("A",7(I3,A))') 'plot for [ in "",Index1,"",Index2,"",Index3,"",Index4,"",Index5,"",Index6,"",Index7,""] "xydata1.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)

Else If (Present(Index6)) Then
    Write(10,'("A",6(I3,A))') 'plot for [ in "",Index1,"",Index2,"",Index3,"",Index4,"",Index5,"",Index6,""] "xydata1.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)

Else If (Present(Index5)) Then
    Write(10,'("A",5(I3,A))') 'plot for [ in "",Index1,"",Index2,"",Index3,"",Index4,"",Index5,""] "xydata1.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)

Else If (Present(Index4)) Then
    Write(10,'("A",4(I3,A))') 'plot for [ in "",Index1,"",Index2,"",Index3,"",Index4,""] "xydata1.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)

Else If (Present(Index3)) Then
    Write(10,'("A",3(I3,A))') 'plot for [ in "",Index1,"",Index2,"",Index3,""] "xydata1.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)

Else If (Present(Index2)) Then
    Write(10,'("A",2(I3,A))') 'plot for [ in "",Index1,"",Index2,""] "xydata1.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)

Else
    Write(10,'("A",1(I3,A))') 'plot for [ in "",Index1,""] "xydata1.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)
End If

CLOSE(10,STATUS="KEEP")

! Plot curve with gnuplot and cleanup files
!----------------------------------------------------------
Call SYSTEM("gnuplot gp.txt")

! Pause ! To check the gnu file before deletion if the graph is not as expected.
Call SYSTEM("del gp.txt")
Call SYSTEM("del xydata1.dat")

! Check message
!----------------------------------------------------------
Write(*,'(A)') 'File ""/trim(title)/".png" is created.'
End Subroutine

!----------------------------------------------------------
**Subroutine** f2gSieve2(title, xlabel, ylabel, Sample_Code, D, Sample_Code_V, D_V, Index1, style1, Index2, style2, Index3, style3, Index4, style4, Index5, style5, Index6, style6, Index7, style7, Index8, style8, Index9, style9)

Plots different sandcurves, up to 5 in the same graph.

Implicit None

Character(*) : title
Character(*) : xlabel, ylabel
Character(14) : Sample_Code(:), Sample_Code_V(:)
Integer, Optional : Index3, Index4, Index5, Index6, Index7, Index8, Index9
Character(2) : Style1, Style2
Character(2), Optional : Style3, Style4, Style5, Style6, Style7, Style8, Style9
Character(25) : Layout1
Character(25) : Layout2, Layout3, Layout4, Layout5, Layout6, Layout7, Layout8, Layout9
Real : D(:, :) Real : D_V(:, :)
Integer : i, j

---

Write data in files

OPEN(10, Access = 'SEQUENTIAL', File = 'xydata1.dat')
If ((size(D)/99) == 69) Write(10, '(A,69(1x,A))') 'Ydata', Sample_Code(:)
Do j = 1, 99
Write(10, '(1,1x,98(F,1x),F)') j, D(:, j)
End Do
CLOSE(10, Status = 'KEEP')

OPEN(10, Access = 'SEQUENTIAL', File = 'xydata2.dat')
Write(10, '(A,43(1x,A))') 'Ydata', Sample_Code_V(:)
Do j = 1, 99
Write(10, '(1,1x,98(F,1x),F)') j, D_V(:, j)
End Do
CLOSE(10, Status = 'KEEP')

Create gnuplot command file

OPEN(10, ACCESS= 'SEQUENTIAL', FILE= 'gp.txt')
Write(10, '(A)') 'set terminal pngcairo size 2400,2400' ! linewidth 2.0 font "?,40" size 2000,1000'
Write(10, '(A)') 'set output ''""''/D:/hijnekam/Desktop/Program/Graphs/SieveCurves/Plot '''//trim(title)//''.png''''
Write(10, '(A)') 'set title '''''//trim(title)'''' font "?,32''''''
Write(10, '(A)') 'set xrange [0.001 : 10 ]' ! Found manually based on the picture used, since it already contains a scale...
Write(10, '(A)') 'set xtics'
Write(10, '(A)') 'set mxtics'
Write(10, '+') 'set xlabel ''''''''//Trim(xlabel)'''''''' font",28'''''''
Write(10, '(A)') 'set yrange [ 0 : 100 ]' ! Idem dito...
Write(10, '(A)') 'set ytics 0,10,100'
Write(10, '+') 'set ylabel ''''''''//Trim(ylabel)'''''''' font",28'''''''' offset -4.0'''''''
Write(10, '(A)') 'set grid xtics mxtics ytics lc rgb "black" lt 1 lw 1'
Write(10, '(A)') 'set key Left reverse font "?,24''''''''
Write(10,+) 'set tmargin at screen 0.95'
Write(10,+) 'set bmargin at screen 0.05'
Write(10,+) 'set rmargin at screen 0.95'
Write(10,+) 'set lmargin at screen 0.05'

If (Present(Index3)) Call GraphStyle(Style3,Layout3,Index3)
If (Present(Index4)) Call GraphStyle(Style4,Layout4,Index4)
If (Present(Index5)) Call GraphStyle(Style5,Layout5,Index5)
If (Present(Index6)) Call GraphStyle(Style6,Layout6,Index6)
If (Present(Index7)) Call GraphStyle(Style7,Layout7,Index7)
If (Present(Index8)) Call GraphStyle(Style8,Layout8,Index8)
If (Present(Index9)) Call GraphStyle(Style9,Layout9,Index9)

Write(10,+) 'set log x'

If (Present(Index9)) Then
Write(10,'(A,9(I3,A))') 'plot for [n in "',Index1,'",Index2,'",Index3,'",Index4,'",Index5,'",Index6,'",Index7,'",Index8,'",Index9,'"]' "xydatal.dat" using (column(n+1))
   :1 with lines is (n) title columnheader(n+1)
Else If (Present(Index8)) Then
Write(10,'(A,8(I3,A))') 'plot for [n in "',Index1,'",Index2,'",Index3,'",Index4,'",Index5,'",Index6,'",Index7,'",Index8,'"]' "xydatal.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)
Else If (Present(Index7)) Then
Write(10,'(A,7(I3,A))') 'plot for [n in "',Index1,'",Index2,'",Index3,'",Index4,'",Index5,'",Index6,'",Index7,'"]' "xydatal.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)
Else If (Present(Index6)) Then
Write(10,'(A,6(I3,A))') 'plot for [n in "',Index1,'",Index2,'",Index3,'",Index4,'",Index5,'",Index6,'"]' "xydatal.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)
Else If (Present(Index5)) Then
Write(10,'(A,5(I3,A))') 'plot for [n in "',Index1,'",Index2,'",Index3,'",Index4,'",Index5,'"]' "xydatal.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)
Else If (Present(Index4)) Then
Write(10,'(A,4(I3,A))') 'plot for [n in "',Index1,'",Index2,'",Index3,'",Index4,'"]' "xydatal.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)
Else If (Present(Index3)) Then
Write(10,'(A,3(I3,A))') 'plot for [n in "',Index1,'",Index2,'",Index3,'"]' "xydatal.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1)
Else
Write(10,'(A,2(I3,A))') 'plot for [n in "',Index1,'"]' "xydatal.dat" using (column(n+1)):1 with lines is (n) title columnheader(n+1), for [n in "",Index2,'"]' "xydata2.dat" using (column(m(n+1))):1 with lines is (m) title columnheader(m(n+1))
End If

CLOSE(10,STATUS='KEEP')

! Plot curve with gnuplot and cleanup files
!---------------
Call SYSTEM('gnuplot gp.txt')
!
! Pause ! To check the gnu file before deletion if the graph is not as expected.
!
Call SYSTEM('del gp.txt')
Call SYSTEM('del xydatal.dat')
Call SYSTEM('del xydata2.dat')
!
! Check message
!---------------
Write(+) '(A)' 'File '/i'""'i'""''/ Plot '/trim(title)'/.'png' '/""''/ is created.'
End Subroutine
!
Subroutine f2gpScenarios(ID, title, xlabel, ylabel, style0, style1, style2, style3, style4, style5, style6, style7, style8, style9)
Character(*), Parameter : : filefolder_out = "D:\hijnem\Desktop\Program\ProgramOutput\" ! Output files
Integer :: Temp=21
Integet :: Y, n0, i ! Sample ID., used for title of graph and graph filename.
Character(2) : : style1, style2, style3, style4, style5, style6, style7,
FILE_DATA  style8, style9, style0
Character(len=*) : : title
Character(len=*) : : xlabel, ylabel
Character(len=*) : : xlabell, ylabell
Character(len=*) : : title
Character(11) : : g_title1, g_title2, g_title3, g_title4, g_title5, g_title6,
g_title7, g_title8, g_title9, g_title0
Character(25) : : Layout1, Layout2, Layout3, Layout4, Layout5, Layout6, Layout7,
FILE_DATA Layout8, Layout9, Layout0
Character(150) : : Filename1, Filename2, Filename3, Filename4, Filename5, Filename6,
FILE_DATA Filename7, Filename8, Filename9, Filename0
Character(19) :: Allocatable : : Var(:)
Character(19) : : DateTimeSource
Character(19), Allocatable : : Var(:)
Character(19) :: DateTimeMin
Character(19) :: DateTimeMax
Write (Filename0, '(A, A, A, I2, 2, A)') 'D:\hijnem\Desktop\Program\ProgramOutput\Scenarios\',
FILE_DATA trim(ID), 'Away Scenario01.txt' ! This is the same for all scenarios for one samples since it depends on the sampling time which is constant.
Write (Filename1, '(A, A, A, I2, 2, A)') 'D:\hijnem\Desktop\Program\ProgramOutput\Scenarios\',
FILE_DATA trim(ID), 'PumpedVol Scenario01.txt'
Write (Filename2, '(A, A, A, I2, 2, A)') 'D:\hijnem\Desktop\Program\ProgramOutput\Scenarios\',
FILE_DATA trim(ID), 'PumpedVol Scenario02.txt'
Write (Filename3, '(A, A, A, I2, 2, A)') 'D:\hijnem\Desktop\Program\ProgramOutput\Scenarios\',
FILE_DATA trim(ID), 'PumpedVol Scenario03.txt'
Write (Filename4, '(A, A, A, I2, 2, A)') 'D:\hijnem\Desktop\Program\ProgramOutput\Scenarios\',
FILE_DATA trim(ID), 'PumpedVol Scenario04.txt'
Write (Filename5, '(A, A, A, I2, 2, A)') 'D:\hijnem\Desktop\Program\ProgramOutput\Scenarios\',
FILE_DATA trim(ID), 'PumpedVol Scenario05.txt'
Write (Filename6, '(A, A, A, I2, 2, A)') 'D:\hijnem\Desktop\Program\ProgramOutput\Scenarios\',
FILE_DATA trim(ID), 'PumpedVol Scenario06.txt'
Write (Filename7, '(A, A, A, I2, 2, A)') 'D:\hijnem\Desktop\Program\ProgramOutput\Scenarios\',
FILE_DATA trim(ID), 'PumpedVol Scenario07.txt'
Write (Filename8, '(A, A, A, I2, 2, A)') 'D:\hijnem\Desktop\Program\ProgramOutput\Scenarios\',
FILE_DATA trim(ID), 'PumpedVol Scenario08.txt'
Write (Filename9, '(A, A, A, I2, 2, A)') 'D:\hijnem\Desktop\Program\ProgramOutput\Scenarios\',
FILE_DATA trim(ID), 'PumpedVol Scenario09.txt'
Open(Temp, file = Filename0)
n0 = lincount(Temp)
Allocate (Var(n0))
Do i=1,n0
   Read(Temp, '(A)') Var(i)
   If (var(i) .eq. '') var(i) = " "
End Do
DateTimeSource = Var(n0)
Deallocate (Var)
Close(Temp)
Y = 12 ! Amount of hours that will be substracted from the source datetime and added to it, in order to respectively form the minimum and maximum of the zoomed plot
Call ZoomGPHours(DateTimeSource, DateTimeMin, DateTimeMax, Y)
OPEN(10,ACCESS='SEQUENTIAL',FILE='gp.txt')
Write(10,'(A)') 'set terminal pngcairo size 1800_1800' ! Sets it to create .png
Write(10,'(A)') 'set output "D:/hijnekam/Desktop/Program/Graphs/Plot Scenarios/Plot_Scenarios_//trim(ID)///.png"' ! Creates .png file and stores it in a different folder. The "//" used is different from the default or fortran ("/")!!
Write(10,'(A)') 'set title font ",28" //"Bold '//trim(Title)//"' ! Graph Title, title in bold
Write(10,'(A)') 'set xlabel font ",24" //"' TRIM(xlabel) / /' " //trim(Layout0) / /' title "' / /' ! Because Gnuplot reads time and data as two separate columns the y-data becomes column 3.
CLOSE(10,STATUS='KEEP')
Write(10,'(A)') 'set key font ",20" opaque right top box'
Write(10,'(5(A))') 'set xrange ['','DateTimeMin',':','DateTimeMax',']'
Write(10,'(A)') 'set yrange [0:1.5]' ! Offset changes the position of the title, in this case lowers it, w/o it the title and tic labels overlap. First number offsets the horizontal position.
Write(10,'(A)') 'set ylabel font ",24" //"TRIM(ylabel)//"'
Write(10,'(A)') 'set grid'
Write(10,'(A)') 'set lmargin at screen 0.05'
Write(10,'(A)') 'set rmargin at screen 0.85'
Write(10,'(A)') 'set bmargin at screen 0.06'
Write(10,'(A)') 'set tmargin at screen 0.95'
Call GraphStyle(Style1,Layout1)
Call GraphStyle(Style2,Layout2)
Call GraphStyle(Style3,Layout3)
Call GraphStyle(Style4,Layout4)
Call GraphStyle(Style5,Layout5)
Call GraphStyle(Style6,Layout6)
Call GraphStyle(Style7,Layout7)
Call GraphStyle(Style8,Layout8)
Call GraphStyle(Style9,Layout9)
Call GraphStyle(Style0,Layout0)
Write(10,'(A)') 'plot ''//trim(filename0)//'' using 1:3 with ''//trim(Layout0)///' title ''//TRIM(g_title0)//'' \//''//trim(filename1)//'' using 1:3 with ''//trim(Layout1)///' title ''//TRIM(g_title1)//'' \//''//trim(filename2)//'' using 1:3 with ''//trim(Layout2)///' title ''//TRIM(g_title2)//'' \//''//trim(filename3)//'' using 1:3 with ''//trim(Layout3)///' title ''//TRIM(g_title3)//'' \//''//trim(filename4)//'' using 1:3 with ''//trim(Layout4)///' title ''//TRIM(g_title4)//'' \//''//trim(filename5)//'' using 1:3 with ''//trim(Layout5)///' title ''//TRIM(g_title5)//'' \//''//trim(filename6)//'' using 1:3 with ''//trim(Layout6)///' title ''//TRIM(g_title6)//'' \//''//trim(filename7)//'' using 1:3 with ''//trim(Layout7)///' title ''//TRIM(g_title7)//'' \//''//trim(filename8)//'' using 1:3 with ''//trim(Layout8)///' title ''//TRIM(g_title8)//'' \//''//trim(filename9)//'' using 1:3 with ''//trim(Layout9)///' title ''//TRIM(g_title9)//'' ! Because Gnuplot reads time and data as two separate columns
Call SYSTEM('gnuplot gp.txt')
Write(*,'(A)') 'File "Plot Scenarios //trim(ID)///.png" is created.'
OPEN(10, ACCESS='SEQUENTIAL', FILE='gp.txt')
!
! Create gnuplot command file
!-----------------------------------------
Call GraphStyle(Style1, Layout1)
Call GraphStyle(Style2, Layout2)
Call GraphStyle(Style3, Layout3)
Call GraphStyle(Style4, Layout4)
Call GraphStyle(Style5, Layout5)
Call GraphStyle(Style6, Layout6)
Call GraphStyle(Style7, Layout7)
Call GraphStyle(Style8, Layout8)
Write(10, (A)) 'plot "/trim(BackgroundName)" / / "D:\bijnekam/Desktop/Program/Graphs/CrossSections/" / / "trim(Filename)" / / '
Write(10, (A)) 'set key at 2825, 1891
Write(10, (A)) 'set title " / / trim(titl e) / / " font " , 32"
Write(10, (A)) 'set output '" / / filename, BackgroundName
Write(10, (A)) 'set terminal pngcairo size 4800,2800' ! linewidth 2.0 font ",.40"
Write(10, (A)) 'set rmargin at screen 0.99'
Write(10, (A)) 'set lmargin at screen 0.05.0.05'
Write(10, (A)) 'set yrange [-100 : 100]' ! Found manually based on the picture used, since it already contains a scale...
Write(10, (A)) 'set ytics!
Write(10, (A)) 'set x2range [0 : *]!' noreverse nowriteback'
Write(10, (A)) 'set grid front lw 3'
Write(10, (A)) 'set xlabel " / / trim(xlabel)" / / "font",28" offset -4.4'
Write(10, (A)) 'set y2range [0 : *]!' noreverse nowriteback'
Write(10, (A)) 'set ytics
Write(10, (A)) 'set yrange [-153.5 : 32]'
Write(10, (A)) 'set xmargin at screen 0.03'
Write(10, (A)) 'set rmargin at screen 0.99'
Write(10, (A)) 'set tmargin at screen 0.95'
Write(10, (A)) 'set key at 2825, -100 Left reverse box lt -1 opaque font ",.24"
! And plots results of the Sample Source program on top of it.
Implicit None
Charact er(*) :: File1, File2, File3, File4, File5, File6, File7, File8
Charact er(*) :: gtitle1, gtitle2, gtitle3, gtitle4, gtitle5, gtitle6, gtitle7, gtitle8
Charact er(2) :: Style1, Style2, Style3, Style4, Style5, Style6, Style7, Style8
Charact er(25) :: Layout1, Layout2, Layout3, Layout4, Layout5, Layout6, Layout7, Layout8
Charact er(*) :: title
Charact er(len=*) :: xlabel, ylabel
Integ er :: ScenarioNo
Charact er(50) :: Filename, BackgroundName
Charact er(50) :: Command

End Subroutine

!-------------------------------------------------------------
Subroutine f2gpwBFm(ScenarioNo, BackgroundName, title, xlabel, ylabel, File1, gtitle1, Style1, File2
, gtitle2, Style2, File3, gtitle3, Style3, File4, gtitle4, Style4, File5, gtitle5, Style5, File6,
, gtitle6, Style6, File7, gtitle7, Style7, File8, gtitle8, Style8)
! Implement a picture for the background. Based on amongst others: http://www.gnuplotting.org/tag/image/
! And plots results of the Sample Source program on top of it.
Implicit None

Character(*) :: File1, File2, File3, File4, File5, File6, File7, File8
Character(*) :: gtitle1, gtitle2, gtitle3, gtitle4, gtitle5, gtitle6, gtitle7, gtitle8
Character(2) :: Style1, Style2, Style3, Style4, Style5, Style6, Style7, Style8
Character(25) :: Layout1, Layout2, Layout3, Layout4, Layout5, Layout6, Layout7, Layout8
Character(*) :: title
Character(len=*) :: xlabel, ylabel
Integer :: ScenarioNo
Character(50) :: Filename, BackgroundName
Character(50) :: Command

! Create gnuplot command file
!-----------------------------------------
OPEN(10, ACCESS='SEQUENTIAL', FILE='gp.txt')
Write(10, (A)) 'set terminal pngcairo size 4800,2800' ! linewidth 2.0 font ",.40"
Write(10, (A)) 'set rmargin at screen 0.99'
Write(10, (A)) 'set lmargin at screen 0.05.0.05'
Write(10, (A)) 'set yrange [-650 : 6110]' ! Found manually based on the picture used, since it already contains a scale...
Write(10, (A)) 'set ytics
Write(10, (A)) 'set yrange [-153.5 : 32]'
Write(10, (A)) 'set xmargin at screen 0.03'
Write(10, (A)) 'set rmargin at screen 0.99'
Write(10, (A)) 'set tmargin at screen 0.95'
Write(10, (A)) 'set key at 2825, -100 Left reverse box lt -1 opaque font ",.24"
Call GraphStyle(Style1, Layout1)
Call GraphStyle(Style2, Layout2)
Call GraphStyle(Style3, Layout3)
Call GraphStyle(Style4, Layout4)
Call GraphStyle(Style5, Layout5)
Call GraphStyle(Style6, Layout6)
Call GraphStyle(Style7, Layout7)
Call GraphStyle(Style8, Layout8)
Write(10, (A)) 'plot "/trim(BackgroundName)" / / ",.png" binary filetype=png using 1:2:3:(225)

260  F. Code of the MarsdiepVisual Program
1953
Write (10, '(A, I, 10 (F))') xdata(i), i, ydata1(i), ydata2(i), ydata3(i), ydata4(i), ydata5(i),

1952
Do i = 1, n1

1950
OPEN(10, Access = 'SEQUENTIAL', File = 'x ydata1.dat')

1949
!

1948
! W r i t e d a t a i n f i l e s

1947
!

1946
!

1945
n1 = size(xdata)

1943
Character(25) : : Layout1, Layout2, Layout3, Layout4, Layout5, Layout6, Layout7,

1942
Character(len=*) : : xlabel, ylabel, title, name ! p l o t a x i s l a b e l s and

1941
Real, Optional : : ydata2(:), ydata3(:), ydata4(:), ydata5(:), ydata6(:), ydata7(:), ydata8(:), ydata9(:), ydata10(:)

1940
Character(len=*) : : xlabel, ylabel, title, name ! p l o t a x i s l a b e l s and
title(s)

1939
Character(10) : : Location

1938
Integer : : i

1937
Character(*) : : graph_title1

1936
Character(*), Optional : : graph_title2, graph_title3, graph_title4, graph_title5,

1935
graph_title6, graph_title7, graph_title8, graph_title9, graph_title10

1934
Character(2) : : style1

1933
Character(2), Optional : : style2, style3, style4, style5, style6, style7, style8, style9,

1932
style10

1931
Character(25) : : Layout1, Layout2, Layout3, Layout4, Layout5, Layout6, Layout7,

1930
Layout8, Layout9, Layout10

1929
nl = size(xdata)

1928
! Write data in files

1927
!----------------------------------------

1926
OPEN(10, Access = 'SEQUENTIAL', File = 'xydata1.dat')

1925
If (Present(ydata10)) Then

1924
Do i=1, nl

1923
Write(10, '(A, I, 10 (F))') xdata(i), i, ydata1(i), ydata2(i), ydata3(i), ydata4(i), ydata5(i),
ydata6(i), ydata7(i), ydata8(i), ydata9(i), ydata10(i)
End Do
El If (Present(ydata9)) Then
  Do i=1,n1
  Write(10, '(A,1,9(F))') xdata(i), i, ydata(i), ydata1(i), ydata2(i), ydata3(i), ydata4(i), ydata5(i),
  ydata6(i), ydata7(i), ydata8(i), ydata9(i)
End Do
End If (Present(ydata8)) Then
  Do i=1,n1
  Write(10, '(A,1,8(F))') xdata(i), i, ydata1(i), ydata2(i), ydata3(i), ydata4(i), ydata5(i),
  ydata6(i), ydata7(i), ydata8(i)
End Do
End If (Present(ydata7)) Then
  Do i=1,n1
  Write(10, '(A,1,7(F))') xdata(i), i, ydata1(i), ydata2(i), ydata3(i), ydata4(i), ydata5(i),
  ydata6(i), ydata7(i)
End Do
End If (Present(ydata6)) Then
  Do i=1,n1
  Write(10, '(A,1,6(F))') xdata(i), i, ydata1(i), ydata2(i), ydata3(i), ydata4(i), ydata5(i),
  ydata6(i)
End Do
El If (Present(ydata5)) Then
  Do i=1,n1
  Write(10, '(A,1,5(F))') xdata(i), i, ydata1(i), ydata2(i), ydata3(i), ydata4(i), ydata5(i)
End Do
El If (Present(ydata4)) Then
  Do i=1,n1
  Write(10, '(A,1,4(F))') xdata(i), i, ydata1(i), ydata2(i), ydata3(i), ydata4(i)
End Do
El If (Present(ydata3)) Then
  Do i=1,n1
  Write(10, '(A,1,3(F))') xdata(i), i, ydata1(i), ydata2(i), ydata3(i)
End Do
El If (Present(ydata2)) Then
  Do i=1,n1
  Write(10, '(A,1,2(F))') xdata(i), i, ydata1(i)
End Do
El If (Printed(ydata1)) Then
  Do i=1,n1
  Write(10, '(A,1,F)') xdata(i), i, ydata1(i)
End Do
End If
CLOSE(10, Status='KEEP')

! Create gnuplot command file
OPEN(10, ACCESS='SEQUENTIAL', FILE= 'gp.txt')
Write(10,*) 'set terminal png size 2400,800' ! Sets it to create .png
Write(10, '(A)') 'set output' '$D:$/hijmekam/Desktop/Program/Graphs/Plot_'//trim(Short(Location))//'.//trim(Name)//'.png'//''' ! Creates .png file and stores it in a different folder. The "/" used is different from the default or fortran ("\") !!!
Write(10,*) 'set title' '//''/Bold''/Title//'''' ! Graph Title, title in bold, works
Write(10,*) 'set xlabel' '//''/Trim(xlabel)//''''
Write(10,*) 'set ylabel' '//''/Trim(ylabel)//''''
Write(10,*) 'set key Left right outside'
Write(10,*) 'set xtics rotate by -45' ! show gridlines in the xy plane
Write(10,*) 'set grid'
Call GraphStyle(Style1, Layout1)
If (Present(Style2)) Call GraphStyle(Style2, Layout2)
If (Present(Style3)) Call GraphStyle(Style3, Layout3)
If (Present(Style4)) Call GraphStyle(Style4, Layout4)
If (Present(Style5)) Call GraphStyle(Style5, Layout5)
If (Present(Style6)) Call GraphStyle(Style6, Layout6)
If (Present(Style7)) Call GraphStyle(Style7, Layout7)
Else If (Present(ydata4)) Then
Write(10,'(A)') 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout1)" title ""//TRIM(graph_title1)//"", "xydata1.dat" using 2:4:xticlabels(1) with "//trim(Layout2)" title ""//TRIM(graph_title2)//", "xydata1.dat" using 2:5:xticlabels(1) with "//trim(Layout3)" title ""//TRIM(graph_title3)//", "xydata1.dat" using 2:6:xticlabels(1) with "//trim(Layout4)" title ""//TRIM(graph_title4)//", "xydata1.dat" using 2:7:xticlabels(1) with "//trim(Layout5)" title ""//TRIM(graph_title5)//", "xydata1.dat" using 2:8:xticlabels(1) with "//trim(Layout6)" title ""//TRIM(graph_title6)//", "xydata1.dat" using 2:9:xticlabels(1) with "//trim(Layout7)" title ""//TRIM(graph_title7)//", "xydata1.dat" using 2:10:xticlabels(1) with "//trim(Layout8)" title ""//TRIM(graph_title8)//", "xydata1.dat" using 2:11:xticlabels(1) with "//trim(Layout9)" title ""//TRIM(graph_title9)//", "xydata1.dat" using 2:12:xticlabels(1) with "//trim(Layout10)" title ""//TRIM(graph_title10)//"

Else If (Present(ydata5)) Then
Write(10,'(A)') 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout1)" title ""//TRIM(graph_title1)//", "xydata1.dat" using 2:4:xticlabels(1) with "//trim(Layout2)" title ""//TRIM(graph_title2)//", "xydata1.dat" using 2:5:xticlabels(1) with "//trim(Layout3)" title ""//TRIM(graph_title3)//", "xydata1.dat" using 2:6:xticlabels(1) with "//trim(Layout4)" title ""//TRIM(graph_title4)//", "xydata1.dat" using 2:7:xticlabels(1) with "//trim(Layout5)" title ""//TRIM(graph_title5)//", "xydata1.dat" using 2:8:xticlabels(1) with "//trim(Layout6)" title ""//TRIM(graph_title6)//", "xydata1.dat" using 2:9:xticlabels(1) with "//trim(Layout7)" title ""//TRIM(graph_title7)//", "xydata1.dat" using 2:10:xticlabels(1) with "//trim(Layout8)" title ""//TRIM(graph_title8)//", "xydata1.dat" using 2:11:xticlabels(1) with "//trim(Layout9)" title ""//TRIM(graph_title9)//", "xydata1.dat" using 2:12:xticlabels(1) with "//trim(Layout10)" title ""//TRIM(graph_title10)//"

Else If (Present(ydata7)) Then
Write(10,'(A)') 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout1)" title ""//TRIM(graph_title1)//", "xydata1.dat" using 2:4:xticlabels(1) with "//trim(Layout2)" title ""//TRIM(graph_title2)//", "xydata1.dat" using 2:5:xticlabels(1) with "//trim(Layout3)" title ""//TRIM(graph_title3)//", "xydata1.dat" using 2:6:xticlabels(1) with "//trim(Layout4)" title ""//TRIM(graph_title4)//", "xydata1.dat" using 2:7:xticlabels(1) with "//trim(Layout5)" title ""//TRIM(graph_title5)//", "xydata1.dat" using 2:8:xticlabels(1) with "//trim(Layout6)" title ""//TRIM(graph_title6)//", "xydata1.dat" using 2:9:xticlabels(1) with "//trim(Layout7)" title ""//TRIM(graph_title7)//", "xydata1.dat" using 2:10:xticlabels(1) with "//trim(Layout8)" title ""//TRIM(graph_title8)//", "xydata1.dat" using 2:11:xticlabels(1) with "//trim(Layout9)" title ""//TRIM(graph_title9)//", "xydata1.dat" using 2:12:xticlabels(1) with "//trim(Layout10)" title ""//TRIM(graph_title10)//"

Else If (Present(ydata6)) Then
Write(10,'(A)') 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout1)" title ""//TRIM(graph_title1)//", "xydata1.dat" using 2:4:xticlabels(1) with "//trim(Layout2)" title ""//TRIM(graph_title2)//", "xydata1.dat" using 2:5:xticlabels(1) with "//trim(Layout3)" title ""//TRIM(graph_title3)//", "xydata1.dat" using 2:6:xticlabels(1) with "//trim(Layout4)" title ""//TRIM(graph_title4)//", "xydata1.dat" using 2:7:xticlabels(1) with "//trim(Layout5)" title ""//TRIM(graph_title5)//", "xydata1.dat" using 2:8:xticlabels(1) with "//trim(Layout6)" title ""//TRIM(graph_title6)//", "xydata1.dat" using 2:9:xticlabels(1) with "//trim(Layout7)" title ""//TRIM(graph_title7)//", "xydata1.dat" using 2:10:xticlabels(1) with "//trim(Layout8)" title ""//TRIM(graph_title8)//", "xydata1.dat" using 2:11:xticlabels(1) with "//trim(Layout9)" title ""//TRIM(graph_title9)//", "xydata1.dat" using 2:12:xticlabels(1) with "//trim(Layout10)" title ""//TRIM(graph_title10)//"
Else If (Present(ydata3)) Then
  Write(10,'(A)') 'plot "xydata1.dat" using 2:3:xticlabels(1) with '//'trim(Layout1)//'
title ""//TRIM(graph_title1)//"","xydata1.dat" using 2:4:xticlabels(1) with '//'trim(Layout2)//'
title ""//TRIM(graph_title2)//"","xydata1.dat" using 2:5:xticlabels(1) with '//'trim(Layout3)//'
title ""//TRIM(graph_title3)//""
Else If (Present(ydata2)) Then
  Write(10,'(A)') 'plot "xydata1.dat" using 2:3:xticlabels(1) with '//'trim(Layout1)//'
title ""//TRIM(graph_title1)//"","xydata1.dat" using 2:4:xticlabels(1) with '//'trim(Layout2)//'
title ""//TRIM(graph_title2)//""
Else
  Write(10,'(A)') 'plot "xydata1.dat" using 2:3:xticlabels(1) with '//'trim(Layout1)//'
End If
CLOSE(10,STATUS='KEEP')
!
! Plot curve with gnuplot and cleanup files
!---------------------------------------------------------
Call SYSTEM('gnuplot gp.txt')
!
! Pause ! To check the gnu file before deletion if the graph is not as expected
!
Call SYSTEM('del gp.txt')
Call SYSTEM('del xydata1.dat')
!
! Check message
!---------------------------------------------------------
Write(*,'(A)') 'File "Plot //trim(Title)//".png" is created.'
End Subroutine
!
Subroutine f2gpYPData(Location, name, title, xlabel, ylabel, xdata, ydata1, graph_title1, style1,
ydata2, graph_title2, style2, ydata3, graph_title3, style3, ydata4, graph_title4, style4,
ydata5, graph_title5, style5, ydata6, graph_title6, style6, ydata7, graph_title7, style7,
ydata8, graph_title8, style8, ydata9, graph_title9, style9, ydata10, graph_title10, style10)
!
Subroutine f2gpRData plots the results of the viscometer
Implicit None

Integer :: nl ! number of data points
Character(*) :: xdata(-) ! first x data array
Real, Optional :: ydata1(-) !
Real, Optional :: ydata2(-), ydata3(-), ydata4(-), ydata5(-), ydata6(-), ydata7(-), ydata8(-)
Real, Optional :: ydata9(-), ydata10(-)
Character(len=*):= xlabel, ylabel, title, name ! plot axis labels and
title(s)
Character(10) :: Location
Integer :: i
Character(*) :: graph_title1
Character(*), Optional :: graph_title2, graph_title3, graph_title4, graph_title5,
graph_title6, graph_title7, graph_title8, graph_title9, graph_title10
Character(2) :: style1
Character(2), Optional :: style2, style3, style4, style5, style6, style7, style8, style9,
style10
Character(25) :: Layout1, Layout2, Layout3, Layout4, Layout5, Layout6, Layout7,
Layout8, Layout9, Layout10
Character(25) :: Layout99
nl = size(xdata)
!
! Write data in files
!
!-----------------------------------------------
OPEN(10,Access="SEQUENTIAL", File='xydata1.dat')
If (Present(ydata10)) Then
  Do i=1,nl
    Write(10,'(A,1,10(F))') xdata(i),i,ydata1(i),ydata2(i),ydata3(i),ydata4(i),ydata5(i),
  !
Else If (Present(ydata9)) Then
  Do i=1,n1
    Write(10,"(A, I, 9(F))") xdata(i), i, ydata1(i), ydata2(i), ydata3(i), ydata4(i), ydata5(i), ydata6(i), ydata7(i), ydata8(i), ydata9(i)
  End Do
Else If (Present(ydata8)) Then
  Do i=1,n1
    Write(10,"(A, I, 8(F))") xdata(i), i, ydata1(i), ydata2(i), ydata3(i), ydata4(i), ydata5(i), ydata6(i), ydata7(i), ydata8(i)
  End Do
Else If (Present(ydata7)) Then
  Do i=1,n1
    Write(10,"(A, I, 7(F))") xdata(i), i, ydata1(i), ydata2(i), ydata3(i), ydata4(i), ydata5(i), ydata6(i), ydata7(i)
  End Do
Else If (Present(ydata6)) Then
  Do i=1,n1
    Write(10,"(A, I, 6(F))") xdata(i), i, ydata1(i), ydata2(i), ydata3(i), ydata4(i), ydata5(i)
  End Do
Else If (Present(ydata5)) Then
  Do i=1,n1
    If (ydata1(i) < ydata2(i)) Then
      Write(10,"(A, I, 5(F))") xdata(i), i, ydata1(i), ydata2(i), ydata3(i), ydata4(i), ydata3(i)
    Else If (ydata1(i) > ydata2(i)) Then
      Write(10,"(A, I, 5(F))") xdata(i), i, ydata1(i), ydata2(i), ydata3(i), ydata4(i), ydata4(i)
    End If
  End Do
Else If (Present(ydata4)) Then
  Do i=1,n1
    Write(10,"(A, I, 4(F))") xdata(i), i, ydata1(i), ydata2(i), ydata3(i), ydata4(i), ydata3(i), ydata4(i)
  End Do
Else If (Present(ydata3)) Then
  Do i=1,n1
    Write(10,"(A, I, 3(F))") xdata(i), i, ydata1(i), ydata2(i), ydata3(i)
  End Do
Else If (Present(ydata2)) Then
  Do i=1,n1
    Write(10,"(A, I, 2(F))") xdata(i), i, ydata1(i), ydata2(i)
  End Do
Else
  Do i=1,n1
    Write(10,"(A, I, F)") xdata(i), i, ydata1(i)
  End Do
End If

CLOSE(10, Status='KEEP')

! Create gnuplot command file
OPEN(10, ACCESS='SEQUENTIAL', FILE='gp.txt')
Write(10,'set terminal png size 2200,600') ! Sets it to create .png
Write(10,'set output "D:\hijnekam\Desktop\Programs\Graphs\Plot\trim(Short(Location))"') ! Creates .png file and stores it in a different folder. The "\" used is different from the default or fortran ("\")!!!
Write(10,'set title ""') ! Graph Title, title in bold, works
Write(10,'set xlabel ""') !/""/Trim(xlabel)"/" font ",18"
Write(10,'set ylabel ""') !/""/Trim(ylabel)"/" font ",18"
Write(10,'set yrange [0:150]')
Write(10,'set ytics 0,25,150')
Write(10,'set y2label "Yield Stress [Pa]" font ",18"
Write(10,'set y2range [0:30]')
Write(10,'set y2tics 0,5,30')
Write(10,'set key left inside box font ",16"')
Write(10,*), 'set xtics rotate by −90'  ! sample code on the right misses a part if −45
Write(10,*), 'set grid'  ! show gridlines in the xy plane

Call GraphStyle('Style1', Layout1)
If (Present('Style2')) Call GraphStyle('Style2', Layout2)
If (Present('Style3')) Call GraphStyle('Style3', Layout3)
If (Present('Style4')) Call GraphStyle('Style4', Layout4)
If (Present('Style5')) Call GraphStyle('Style5', Layout5)
If (Present('Style6')) Call GraphStyle('Style6', Layout6)
If (Present('Style7')) Call GraphStyle('Style7', Layout7)
If (Present('Style8')) Call GraphStyle('Style8', Layout8)
If (Present('Style9')) Call GraphStyle('Style9', Layout9)
If (Present('Style10')) Call GraphStyle('Style10', Layout10)

Call GraphStyle('lm', Layout99)

If (Present('ydata10')) Then
Write(10,*) 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout2)" title "" //trim(graph_title1)" "", "xydata1.dat" using 2:4:xticlabels(1) with "//trim(Layout3)" title "" //trim(graph_title2)" "", "xydata1.dat" using 2:5:xticlabels(1) with "//trim(Layout4)" title "" //trim(graph_title3)" "", "xydata1.dat" using 2:6:xticlabels(1) with "//trim(Layout5)" title "" //trim(graph_title4)" "", "xydata1.dat" using 2:7:xticlabels(1) with "//trim(Layout6)" title "" //trim(graph_title5)" "", "xydata1.dat" using 2:8:xticlabels(1) with "//trim(Layout7)" title "" //trim(graph_title6)" "", "xydata1.dat" using 2:9:xticlabels(1) with "//trim(Layout8)" title "" //trim(graph_title7)" "", "xydata1.dat" using 2:10:xticlabels(1) with "//trim(Layout9)" title "" //trim(graph_title8)" "", "xydata1.dat" using 2:11:xticlabels(1) with "//trim(Layout10)" title "" //trim(graph_title10)"
Else If (Present('ydata9')) Then
Write(10,*) 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout2)" title "" //trim(graph_title1)" "", "xydata1.dat" using 2:4:xticlabels(1) with "//trim(Layout3)" title "" //trim(graph_title2)" "", "xydata1.dat" using 2:5:xticlabels(1) with "//trim(Layout4)" title "" //trim(graph_title3)" "", "xydata1.dat" using 2:6:xticlabels(1) with "//trim(Layout5)" title "" //trim(graph_title4)" "", "xydata1.dat" using 2:7:xticlabels(1) with "//trim(Layout6)" title "" //trim(graph_title5)" "", "xydata1.dat" using 2:8:xticlabels(1) with "//trim(Layout7)" title "" //trim(graph_title6)" "", "xydata1.dat" using 2:9:xticlabels(1) with "//trim(Layout8)" title "" //trim(graph_title7)" "", "xydata1.dat" using 2:10:xticlabels(1) with "//trim(Layout9)" title "" //trim(graph_title8)"
Else If (Present('ydata8')) Then
Write(10,*) 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout2)" title "" //trim(graph_title1)" "", "xydata1.dat" using 2:4:xticlabels(1) with "//trim(Layout3)" title "" //trim(graph_title2)" "", "xydata1.dat" using 2:5:xticlabels(1) with "//trim(Layout4)" title "" //trim(graph_title3)" "", "xydata1.dat" using 2:6:xticlabels(1) with "//trim(Layout5)" title "" //trim(graph_title4)" "", "xydata1.dat" using 2:7:xticlabels(1) with "//trim(Layout6)" title "" //trim(graph_title5)" "", "xydata1.dat" using 2:8:xticlabels(1) with "//trim(Layout7)" title "" //trim(graph_title6)" "", "xydata1.dat" using 2:9:xticlabels(1) with "//trim(Layout8)" title "" //trim(graph_title7)" "", "xydata1.dat" using 2:10:xticlabels(1) with "//trim(Layout9)" title "" //trim(graph_title8)"
Else If (Present('ydata7')) Then
Write(10,*) 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout2)" title "" //trim(graph_title1)" "", "xydata1.dat" using 2:4:xticlabels(1) with "//trim(Layout3)" title "" //trim(graph_title2)" "", "xydata1.dat" using 2:5:xticlabels(1) with "//trim(Layout4)" title "" //trim(graph_title3)" "", "xydata1.dat" using 2:6:xticlabels(1) with "//trim(Layout5)" title "" //trim(graph_title4)" "", "xydata1.dat" using 2:7:xticlabels(1) with "//trim(Layout6)" title "" //trim(graph_title5)" "", "xydata1.dat" using 2:8:xticlabels(1) with "//trim(Layout7)" title "" //trim(graph_title6)" "", "xydata1.dat" using 2:9:xticlabels(1) with "//trim(Layout8)" title "" //trim(graph_title7)"
Else If (Present('ydata6')) Then
Write(10,*) 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout2)" title "" //trim(graph_title1)" "", "xydata1.dat" using 2:4:xticlabels(1) with "//trim(Layout3)" title "" //trim(graph_title2)" "", "xydata1.dat" using 2:5:xticlabels(1) with "//trim(Layout4)" title "" //trim(graph_title3)" "", "xydata1.dat" using 2:6:xticlabels(1) with "//trim(Layout5)" title "" //trim(graph_title4)" "", "xydata1.dat" using 2:7:xticlabels(1) with "//trim(Layout6)" title "" //trim(graph_title5)" "", "xydata1.dat" using 2:8:xticlabels(1) with "//trim(Layout7)" title "" //trim(graph_title6)"
Else If (Present('ydata5')) Then
Write(10,*) 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout2)" title "" //trim(graph_title1)" "", "xydata1.dat" using 2:4:xticlabels(1) with "//trim(Layout3)" title "" //trim(graph_title2)" "", "xydata1.dat" using 2:5:xticlabels(1) with "//trim(Layout4)" title "" //trim(graph_title3)" "", "xydata1.dat" using 2:6:xticlabels(1) with "//trim(Layout5)" title "" //trim(graph_title4)" "", "xydata1.dat" using 2:7:xticlabels(1) with "//trim(Layout6)" title "" //trim(graph_title5)" "", "xydata1.dat" using 2:8:xticlabels(1) with "//trim(Layout7)" title "" //trim(graph_title6)"
If (Present(ydata1)) Then
Write(10, '(A)') 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout1)" title "" //TRIM(graph_title1) //"", "xydata1.dat" using 2:4:xticlabels(1) with "//trim(Layout2)" title "" //TRIM(graph_title2) //"", "xydata1.dat" using 2:5:xticlabels(1) with "//trim(Layout3)" title "" //TRIM(graph_title3) //"", "xydata1.dat" using 2:6:xticlabels(1) with "//trim(Layout4)" title "" //TRIM(graph_title4) //"", "xydata1.dat" using 2:7:xticlabels(1) with "//trim(Layout5)" title "" //TRIM(graph_title5) //"
Else If (Present(ydata4)) Then
Write(10, '(A)') 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout1)" title "" //TRIM(graph_title1) //"", axes xyl, "xydata1.dat" using 2:4:xticlabels(1) with "//trim(Layout2)" title "" //TRIM(graph_title2) //"", axes xyl, "xydata1.dat" using 2:5:xticlabels(1) with "//trim(Layout3)" title "" //TRIM(graph_title3) //"", axes xly2, "xydata1.dat" using 2:6:xticlabels(1) with "//trim(Layout4)" title "" //TRIM(graph_title4) //"", axes xly2, "xydata1.dat" using 2:7:xticlabels(1) with "//trim(Layout99)" title "" //trim(title) //"", axes xly2
Else If (Present(ydata3)) Then
Write(10, '(A)') 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout1)" title "" //TRIM(graph_title1) //"", "xydata1.dat" using 2:4:xticlabels(1) with "//trim(Layout2)" title "" //TRIM(graph_title2) //"", "xydata1.dat" using 2:5:xticlabels(1) with "//trim(Layout3)" title "" //TRIM(graph_title3) //"
Else If (Present(ydata2)) Then
Write(10, '(A)') 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout1)" title "" //TRIM(graph_title1) //"", "xydata1.dat" using 2:4:xticlabels(1) with "//trim(Layout2)" title "" //TRIM(graph_title2) //"
Else
Write(10, '(A)') 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout1)" title "" //TRIM(graph_title1) //"
End If
!
CLOSE(10, STATUS='KEEP')
!
OPEN(10, ACCESS='SEQUENTIAL', FILE= 'gp2.txt')
Write(10, '*) ' set terminal png size 2200,900 ! Sets it to create .png
Write(10, '*) ' set output '/**'; '/D:/hijnekam/Desktop/Program/Gaphs/Plot_ ' //trim(Short(Location)) //"", '/trim(Name) //", png' //"" ! Creates .png file and stores it in a different folder. The '**' used is different from the default or fortran ('"')!!!
Write(10, '*) ' set title "" //Bold //trim(Location) //"" //Title //"" font '',.20'' ! Graph Title, title in bold, works
Write(10, '*) ' set xlabel "/**'; //Trim(xlabel) //"" font '',.18''
Write(10, '*) ' set yrange [0:30] !
Write(10, '*) ' set ytics 0.25,150 !
Write(10, '*) ' set ytics "Yield Stress [Pa]" font '',.18'' !
Write(10, '*) ' set y2range [0:30] !
Write(10, '*) ' set y2tics 0.5,30 !
Write(10, '*) ' set key left inside box font '',.16'' !
Write(10, '*) ' set xtics rotate by -90 !
Write(10, '*) ' set grid ' ! show gridlines in the xy plane
!
Call GraphStyle(Style1/Layout1)
If (Present(Style2)) Call GraphStyle(Style2/Layout2)
If (Present(Style3)) Call GraphStyle(Style3/Layout3)
If (Present(Style4)) Call GraphStyle(Style4/Layout4)
If (Present(Style5)) Call GraphStyle(Style5/Layout5)
If (Present(Style6)) Call GraphStyle(Style6/Layout5)
If (Present(Style7)) Call GraphStyle(Style7/Layout7)
If (Present(Style8)) Call GraphStyle(Style8/Layout8)
If (Present(Style9)) Call GraphStyle(Style9/Layout9)
If (Present(Style10)) Call GraphStyle(Style10/Layout10)

Call GraphStyle('1m',Layout99)
Call GraphStyle('3k',Layout3)
Call GraphStyle('3l',Layout4)

If (Present(ydata10)) Then
Write(10, '(A)') 'plot "xydata1.dat" using 2:3:xticlabels(1) with "//trim(Layout1)" title "" //TRIM(graph_title1) //"", "xydata1.dat" using 2:4:xticlabels(1) with "//trim(Layout2)" title "" //TRIM(graph_title2) //"", "xydata1.dat" using 2:5:xticlabels(1) with "//trim(Layout3)" title "" //TRIM(graph_title3) //"", "xydata1.dat" using 2:6:xticlabels(1) with "//trim(Layout4)" title "" //TRIM(graph_title4) //"", "xydata1.dat" using 2:7:xticlabels(1) with "//trim(Layout5)" title "" //TRIM(graph_title5) //"", "xydata1.dat" using 2:8:xticlabels(1) with "//trim(Layout6)" title "" //TRIM(graph_title6) //"", "xydata1.dat" using 2:9:xticlabels(1) with "//trim(Layout7)" title "" //TRIM(graph_title7) //"", "xydata1.dat" using 2:10:xticlabels(1) with "//trim(Layout99)" title "" //trim(title) //"", "xydata1.dat" using 2:11:xticlabels(1) with "//trim(Layout100)" title "" //trim(title) //"", axes xly1, axes xly2, axes xly3, axes xly4, axes xly5, axes xly6, axes xly7, axes xly8, axes xly9, axes xly10, axes xly11,
(I) with "/ trim(Layout4) /" title " / TRIM(graph_title4) /" , "xymat.dat" using 2:7: xticklabels(I) with "/ trim(Layout5) /" title " / TRIM(graph_title5) /" , "xymat.dat" using 2:8: xticklabels(I) with "/ trim(Layout6) /" title " / TRIM(graph_title6) /" , "xymat.dat" using 2:9: xticklabels(I) with "/ trim(Layout7) /" title " / TRIM(graph_title7) /" , "xymat.dat" using 2:10: xticklabels(I) with "/ trim(Layout8) /" title " / TRIM(graph_title8) /" , "xymat.dat" using 2:11: xticklabels(I) with "/ trim(Layout9) /" title " / TRIM(graph_title9) /" , "xymat.dat" using 2:12: xticklabels(I) with "/ trim(Layout10) /" title " / TRIM(graph_title10) /" 

Else If (Present(ydata4)) Then
Write(10, (A)) plot "xymat.dat" using 2:3: xticklabels(I) with "/ trim(Layout1) /" title " / TRIM(graph_title1) /" , "xymat.dat" using 2:4: xticklabels(I) with "/ trim(Layout2) /" title " / TRIM(graph_title2) /" , "xymat.dat" using 2:5: xticklabels(I) with "/ trim(Layout3) /" title " / TRIM(graph_title3) /" , "xymat.dat" using 2:6: xticklabels(I) with "/ trim(Layout4) /" title " / TRIM(graph_title4) /" , "xymat.dat" using 2:7: xticklabels(I) with "/ trim(Layout5) /" title " / TRIM(graph_title5) /" , "xymat.dat" using 2:8: xticklabels(I) with "/ trim(Layout6) /" title " / TRIM(graph_title6) /" , "xymat.dat" using 2:9: xticklabels(I) with "/ trim(Layout7) /" title " / TRIM(graph_title7) /" , "xymat.dat" using 2:10: xticklabels(I) with "/ trim(Layout8) /" title " / TRIM(graph_title8) /" 

Else If (Present(ydata8)) Then
Write(10, (A)) plot "xymat.dat" using 2:3: xticklabels(I) with "/ trim(Layout1) /" title " / TRIM(graph_title1) /" , "xymat.dat" using 2:4: xticklabels(I) with "/ trim(Layout2) /" title " / TRIM(graph_title2) /" , "xymat.dat" using 2:5: xticklabels(I) with "/ trim(Layout3) /" title " / TRIM(graph_title3) /" , "xymat.dat" using 2:6: xticklabels(I) with "/ trim(Layout4) /" title " / TRIM(graph_title4) /" , "xymat.dat" using 2:7: xticklabels(I) with "/ trim(Layout5) /" title " / TRIM(graph_title5) /" , "xymat.dat" using 2:8: xticklabels(I) with "/ trim(Layout6) /" title " / TRIM(graph_title6) /" , "xymat.dat" using 2:9: xticklabels(I) with "/ trim(Layout7) /" title " / TRIM(graph_title7) /" 

Else If (Present(ydata6)) Then
Write(10, (A)) plot "xymat.dat" using 2:3: xticklabels(I) with "/ trim(Layout1) /" title " / TRIM(graph_title1) /" , "xymat.dat" using 2:4: xticklabels(I) with "/ trim(Layout2) /" title " / TRIM(graph_title2) /" , "xymat.dat" using 2:5: xticklabels(I) with "/ trim(Layout3) /" title " / TRIM(graph_title3) /" , "xymat.dat" using 2:6: xticklabels(I) with "/ trim(Layout4) /" title " / TRIM(graph_title4) /" , "xymat.dat" using 2:7: xticklabels(I) with "/ trim(Layout5) /" title " / TRIM(graph_title5) /" , "xymat.dat" using 2:8: xticklabels(I) with "/ trim(Layout6) /" title " / TRIM(graph_title6) /" 

Else If (Present(ydata5)) Then
Write(10, (A)) plot "xymat.dat" using 2:3: xticklabels(I) with "/ trim(Layout1) /" title " / TRIM(graph_title1) /" , "xymat.dat" using 2:4: xticklabels(I) with "/ trim(Layout2) /" title " / TRIM(graph_title2) /" , "xymat.dat" using 2:5: xticklabels(I) with "/ trim(Layout3) /" title " / TRIM(graph_title3) /" , "xymat.dat" using 2:6: xticklabels(I) with "/ trim(Layout4) /" title " / TRIM(graph_title4) /" , "xymat.dat" using 2:7: xticklabels(I) with "/ trim(Layout9) /" title " / TRIM(title) /" 

Else If (Present(ydata3)) Then
Write(10, (A)) plot "xymat.dat" using 2:3: xticklabels(I) with "/ trim(Layout1) /" title " / TRIM(graph_title1) /" , "xymat.dat" using 2:4: xticklabels(I) with "/ trim(Layout2) /" title " / TRIM(graph_title2) /" , "xymat.dat" using 2:5: xticklabels(I) with "/ trim(Layout3) /" title " / TRIM(graph_title3) /" 

Else If (Present(ydata2)) Then
Write(10, (A)) plot "xymat.dat" using 2:3: xticklabels(I) with "/ trim(Layout1) /" title " / TRIM(graph_title1) /" , "xymat.dat" using 2:4: xticklabels(I) with "/ trim(Layout2) /" title " / TRIM(graph_title2) /" 

Else
Write(10,'(A)') 'plot"xydata1.dat" using 2:3:xticlabels(1) with '//'trim(Layout1)//' 

title ""//trim(graph_title1)//''''
End If

CLOSE(10, STATUS='KEEP')

' Plot curve with gnuplot and cleanup files
!-----------------------------
Call SYSTEM('gnuplot gp.txt')
Call SYSTEM('gnuplot gp2.txt')

! Pause ! To check the gnu file before deletion if the graph is not as expected.

Call SYSTEM('del gp.txt')
Call SYSTEM('del xydata1.dat')

! Check message
!-----------------------------
Write(*,'(A)') 'File "Plot '//'trim(Title)//'.png" is created.'

End Subroutine

!-----------------------------
Subroutine f2gpDataMulti(Location, name, title, title1, title2, xlabel, ylabel1, ylabel2, ylabel3, 
ylabel4, xdata, ydata01, graph_title01, style01, ydata02, graph_title02, style02, ydata03, 
graph_title03, style03, ydata04, graph_title04, style04)

! Due to the 2 y axis limit multiple plots with the same x axis.
Implicit None

Character(2) :: style01, style02, style03, style04, style05, style06
Integer :: n1 ! number of data points
Character(*) :: xdata(:) ! first x data array
Real :: ydata01(:), ydata02(:), ydata03(:), ydata04(:)
Character(len=*) :: xlabel, ylabel1, ylabel2, ylabel3, ylabel4
Character(len=*) :: title, title1, title2, name ! plot axis labels and title(s)
Character(10) :: Location
Integer :: i
Character(25) :: graph_title01, graph_title02, graph_title03, graph_title04

n1 = size(xdata)

! Write data in files
!-----------------------------
Open(10, ACCESS='Sequential', File='xydata1.dat')

Do i=1,n1
Write(10, '(A,1,4(F))') xdata(i), i, ydata01(i), ydata02(i), ydata03(i), ydata04(i)
End Do

CLOSE(10, Status='KEEP')

! Create gnuplot command file
!-----------------------------
OPEN(10, ACCESS='SEQUENTIAL', FILE='gp.txt')
Write(10,*) 'set terminal pngcairo size 2970,2100' ! Sets it to create .png
Write(10,*) 'set output ''//''//D:/hijnekam/Desktop/Program/Graphs/Plot_'//'trim(Short(Location))//''//trim(Name)//''//''//''//''
Create .png file and stores it in a different folder. The '' used is different from the default or fortran (")"!!!
Write(10,*) 'set multiplot title '//'''//''Bold ''//trim(title)//''''
Write(10,*) 'set size 1.05'
Write(10,*) 'set origin 0.0,0.0'
Write(10,*) 'set title ''''//''Bold ''//Title1''''

Graph Title, title in bold, works
Write(10,*) 'set xlabelf //**//Trim(xlabelf)//**'
Write(10,*) 'set ylabelf //**//Trim(ylabelf2)//**'
Write(10,*) 'set y2labelf //**//Trim(ylabelf1)//**'
Write(10,*) 'set key Left right bottom inside opaque'

Write(10,*) 'set xtics rotate by -90'
Write(10,*) 'set grid'

Write(10,'(A)') 'plot "xydata1.dat" using 2:3 with //trim(Layout1)// title "" //TRIM(graph_title01)//"" axes x2y2, "xydata1.dat" using 2:4 with //trim(Layout2)// title "" //TRIM(graph_title02)//"" axes xyl1'

Write(10,*) 'set size 1,0.5'
Write(10,*) 'set origin 0.0,0.5'

Write(10,*) 'set title //""[/i: Bold //""title2 //"]"
Write(10,*) 'unset xlabel'
Write(10,*) 'set ylabel //**//Trim(ylabelf3)//**'
Write(10,*) 'set y2labelf //**//Trim(ylabelf4)//**'
Write(10,*) 'unset xtics'
Write(10,*) 'set y2tics'

Write(10,*) 'set key Left right bottom inside opaque'

Write(10,*) 'set xtics rotate by -45'
Write(10,*) 'set grid'

Call GraphStyle(Style03,Layout3)
Call GraphStyle(Style04,Layout4)

Write(10,'(A)') 'plot "xydata1.dat" using 2:5 with //trim(Layout3)// title "" //TRIM(graph_title03)//"" axes x2y2, "xydata1.dat" using 2:6 with //trim(Layout4)// title "" //TRIM(graph_title04)//""'

CLOSE(10,STATUS='KEEP')

! Plot curve with gnuplot and cleanup files
!----------------------------------------------------------
Call SYSTEM('gnuplot gp.txt')

! Pause ! To check the gnu file before deletion if the graph is not as expected.

Call SYSTEM('del gp.txt')
Call SYSTEM('del xydata1.dat')

! Check message
!----------------------------------------------------------
Write(*,'(A)') 'File "Plot"//trim(Title)//.png" is created.'

End Subroutine

!----------------------------------------------------------
Subroutine f2gDataMulti2(name, title, title1, title2, xlabelf, ylabelf1, ylabelf2, ylabelf3, ylabelf4, xdata1, xdata2, ydata1, ydata2, graph_title01, style01, ydata21, ydata22, graph_title02, style02, ydata31, ydata32, graph_title03, style03, ydata41, ydata42, graph_title04, style04, ydata51, ydata52, graph_title05, style05, ydata61, ydata62, graph_title06, style06)
! Due to the 2 yaxis limit multiple plots with the same xaxis.
Implicit None
Character(2) :: style01,style02,style03,style04,style05,style06
Integer :: i, n1, n2 ! number of data points
Character(*) :: xdata1(:), xdata2(:) ! first x data array
Real :: ydata11(:), ydata12(:), ydata21(:), ydata22(:), ydata31(:), ydata32(:),
ydata41(:), ydata42(:), ydata51(:), ydata52(:), ydata61(:), ydata62(:)
Character(len=**) :: xlabel1, ylabel1, ylabel2, ylabel3, ylabel4
Character(len=**) :: title, title1, title2, name ! plot axis labels and
title(s)
Character(10) :: Location
Character(*) :: graph_title01, graph_title02, graph_title03, graph_title04,
graph_title05, graph_title06
Character(25) :: Layout1, Layout2, Layout3, Layout4, Layout5, Layout6
Real :: max_d5, max_d6, max_tot

n1 = size(xdata1)
n2 = size(xdata2)

max_d5 = max(maxval(ydata51), maxval(ydata52))
max_d6 = max(maxval(ydata61), maxval(ydata62))
max_tot = max(max_d6, max_d5)

!-----------------------------
! Write data in files
!-----------------------------
Open(10, Access='Sequential', File= 'xydata1.dat')

Do i = 1, (n1+n2)
   If (i <= n1) Then
      If (ydata11(i) == 0 .or. ydata21(i) == 0 .or. ydata31(i) == 0 .or. ydata41(i) == 0 .or.
ydata51(i) == 0 .or. ydata61(i) == 0) Then
         Write(10,*)
      Else
         Write(10,'(A, I, 6(F))') xdata1(i), i, ydata11(i), ydata21(i), ydata31(i), ydata41(i),
         1-(ydata51(i)/max_tot), 1-(ydata61(i)/max_tot)
      End If
   Else If (i > n1) Then
      If (ydata12(i-n1) == 0 .or. ydata22(i-n1) == 0 .or. ydata32(i-n1) == 0 .or. ydata42(i-
n1) == 0 .or. ydata52(i-n1) == 0 .or. ydata62(i-n1) == 0) Then
         Write(10,*)
      Else
         Write(10,'(A, I, 6(F))') xdata2(i-n1), i, ydata12(i-n1), ydata22(i-n1), ydata32(i-n1),
ydata42(i-n1), 1-(ydata52(i-n1)/max_tot), 1-(ydata62(i-n1)/max_tot)
      End If
   End If
End Do

CLOSE(10, Status='KEEP')

! Create gnuplot command file

OPEN(10, ACCESS='SEQUENTIAL', FILE='gp.txt')
Write(10,*) set terminal pngcairo size 2970,2100
Write(10,'(A)') 'set output '/ '' '/''D:/hijnekam/Desktop/Program/Graphs/Plot_''//trim(Name)''/''
   .png''//'' ! Creates .png file and stores it in a different folder. The ''/'' used is different from the default or fortran ("\")!!!
Write(10,*) 'set multiplot title '/ '' '/''Bold ''//trim(title)''/' font ",22"
Write(10,*) 'set size 1.0,49'
Write(10,*) 'set origin 0.0,0.0'
Write(10,*) 'set title '/ '' '/''Bold ''//Title1''/' font ",22"
Write(10,*) 'set xlabel '/ '' '/''//Trim(xlabel)''/' font ",22" offset 0.2
Write(10,*) 'set ylabel '/ '' '/''//Trim(ylabel)''/' font ",22"
Write(10,*) 'set y2label '/ '' '/''//Trim(ylabel2)''/' font ",22"
Write(10,*) 'set key Left left top opaque box font ",20"
Write(10,'(A)') 'set label ''Texel'' at 36.1,35 font ",22"
Write(10,'(A)') 'set label ''Den Helder'' at 39.5,1.35 font ",22"
Write(10,*) 'set xtics rotate by -90'
Write (10, *) 'set grid noxtics ytics lw 2' ! show gridlines in the xy plane
Write (10, ' (A)') 'set xrange [0 : 70 ]'
Write (10, ' (A)') 'set x2range [0 : * ]' ! noreverse nowriteback'
Write (10, ' (A)') 'unset x2tics'
Write (10, ' (A)') 'set yrange [0 : 30 ]'
Write (10, ' (A)') 'set ytics 0, 5, 30 nomirror'
Write (10, ' (A)') 'set y2range [0 : 1.2 ]'
Write (10, ' (A)') 'set y2tics 0, 0.2, 1.2 nomirror'
Write (10, ' (A)') 'set arrow from 38.5, 0 to 38.5, 30 filled nohead ls 0 lw 3'
Write (10, ' (A)') 'plot "xydata1.dat" using 2:3:xticlabels(1) with ''/trim(Layout1)//' title "' //TRIM(graph_title01)/'' axis xyl."xydata1.dat" using 2:4:xticlabels(1) with ''/trim(Layout2)//' title "'//TRIM(graph_title02)/'' axis xly2'
Write (10, *) 'set size 1, 0.49'
Write (10, *) 'set origin 0.0, 0.49'
Write (10, *) 'set title ''/Bold ''//title2'' '' font ',.22''
Write (10, *) 'unset xlabel'
Write (10, *) 'set ylabel ''/// ''//Trim(ylabel3)/'' '' font ',.22''
Write (10, *) 'set ylabel ''/// ''//Trim(ylabel4)/'' '' font ',.22''
Write (10, *) 'unset xtics'
Write (10, *) 'unset ytics'
Write (10, ' (A)') 'set yrange [0:30 ]'
Write (10, ' (A)') 'set ytics 0, 5, 30 nomirror'
Write (10, ' (A)') 'set y2range [0:1.2 ]'
Write (10, ' (A)') 'set y2tics 0, 0.2, 1.2 nomirror'
Write (10, ' (A)') 'set arrow from 38.5, 1.05 to 38.5, 1.4 filled nohead ls 0 lw 3'
Write (10, ' (A)') 'plot "xydata1.dat" using 2:5:xticlabels(1) with ''/trim(Layout3)//' title "' //TRIM(graph_title03)/'' axes xyl."xydata1.dat" using 2:6:xticlabels(1) with ''/trim(Layout4)//' title "' ///TRIM(graph_title04)/'' axes xyl."xydata1.dat" using 2:7: xticlabels(1) with '/trim(Layout5)//' title "' ///TRIM(graph_title05)//'" axes xly2;" xydata1.dat" using 2:8:xticlabels(1) with ''/trim(Layout6)//' title "' ///TRIM( graph_title06)/'' axes xly2'
CLOSE (10, STATUS='KEEP')
Write(*, '(A)') 'File "Plot ' // trim(Title) // '.png" is created.'

End Subroutine

!-------------------------------------------------------------

Subroutine f2gpDataHisto(title, xlabel, ylabel, xdata1, ydata1, ydata2, graph_title1, style1, xdata2, ydata2, ydata2, graph_title2, style2)
! Subroutine f2gpRData plots the results of the viscometer
Implicit None
!
Integer :: i
Integer :: n1, n2
Character(*) :: xdata1(:,), xdata2(:)
Real :: ydata1(:,), ydata2(:,), ydata1(:,), ydata2(:,)
Character(len=*) :: xlabel, ylabel, title ! plot axis labels and title(s)
Character(*) :: graph_title1, graph_title2
Character(2) :: style1, style2
Character(25) :: Layout1, Layout2, Layout3, Layout4

n1 = size(xdata1)
n2 = size(xdata2)
!
! Write data in files
!
OPEN(10, Access='SEQUENTIAL', File='xy_data1.dat')

Write(10, '(A)') 'Sample Code i Bingham Herschel–Bulkley'
!
Do i=1, (n1+n2)
  If (i <= n1) Then
    If (ydata1(i) == 0 .or. ydata2(i) == 0) Then
      Write(10,*)
    Else
      Write(10, '(A, I, 2 (F))') xdata1(i), i, ydata1(i), ydata2(i)
    End If
  End Else If (i > n1) Then
    If (ydata2(i-n1) == 0 .or. ydata2(i-n1) == 0) Then
      Write(10,*)
    Else
      Write(10, '(A, I, 6 (F))') xdata2(i-n1), i, ydata1(i-n1), ydata2(i-n1)
    End If
  End If
End Do
!
CLOSE(10, Status='KEEP')
!
! Create gnuplot command file
!
OPEN(10, ACCESS='SEQUENTIAL', FILE='gp.txt')
Write(10,*) 'set terminal pngcairo size 2400,800' ! Sets it to create .png
Write(10, '(A)') 'set output '/"D:/hjinakam/Desktop/Program/Graphs/Plot"/ // trim(title) // .png"' ! Creates .png file and stores it in a different folder. The "/" used is different from the default or fortran ("/")!!
Write(10,*) 'set title '/"Bold"/ // Title//"')' ! Graph Title, title in bold, works
Write(10,*) 'set xlabel '/"Trim(xlabel)"//"'
Write(10,*) 'set ylabel '/"Trim(ylabel)"//"'
Write(10,*) 'set key Left right inside'
Write(10,*) 'set xtics rotate by -45'
Write(10,*) 'set grid' ! show gridlines in the xy plane
Write(10,*) 'set boxwidth 0.5'
Call GraphStyle(Style1,Layout1)
Call GraphStyle(Style2,Layout2)
CLOSE(10, STATUS='KEEP')

! Plot curve with gnuplot and cleanup files

!-----------------------------------------------
Call SYSTEM('gnuplot gp.txt')

! Pause ! To check the gnu file before deletion if the graph is not as expected
.
Call SYSTEM('del gp.txt')
Call SYSTEM('del xydata1.dat')

! Check message

!-----------------------------------------------
Write(*, '(A)') 'File "Plot //trim(Title)="/\''png'' is created.'

End Subroutine

!-----------------------------------------------
Subroutine GraphStyle( Style, Layout, Index )
! In order to create uniformity in the output, this is separated from the plot subroutines
so it can be used by all of those.
Implicit None

Character(2) :: style
Character(25) :: Layout
Integer, Optional :: Index

Select Case(style)
Case('1a')
  Layout = 'lines ls 1'
  If (Present(Index)) Write(Layout, '(A,12)') 'lines ls ', Index
  If (Present(Index)) Then
    Write(10, '(A,12,A)') 'set style line ', Index, ' lc rgb "red" lt 1 lw 2 pt 7 ps 1'
  Else
    Write(10, '(A)') 'set style line 1 lc rgb "red" lt 1 lw 2 pt 7 ps 1'
  End If

Case('1b')
  Layout = 'lines ls 2'
  If (Present(Index)) Write(Layout, '(A,12)') 'lines ls ', Index
  If (Present(Index)) Then
    Write(10, '(A,12,A)') 'set style line ', Index, ' lc rgb "dark-blue" lt 1 lw 2 pt 7 ps 1'
  Else
    Write(10, '(A)') 'set style line 2 lc rgb "dark-blue" lt 1 lw 2 pt 7 ps 1'
  End If

Case('1c')
  Layout = 'lines ls 3'
  If (Present(Index)) Write(Layout, '(A,12)') 'lines ls ', Index
  If (Present(Index)) Then
    Write(10, '(A,12,A)') 'set style line ', Index, ' lc rgb "orange" lt 1 lw 2 pt 7 ps 1'
  Else
    Write(10, '(A)') 'set style line 3 lc rgb "orange" lt 1 lw 2 pt 7 ps 1'
  End If

Case('1d')
  Layout = 'lines ls 4'
  If (Present(Index)) Write(Layout, '(A,12)') 'lines ls ', Index
  If (Present(Index)) Then
    Write(10, '(A,12,A)') 'set style line ', Index, ' lc rgb "dark-green" lt 1 lw 2 pt 7 ps 1'
  Else
    Write(10, '(A)') 'set style line 4 lc rgb "dark-green" lt 1 lw 2 pt 7 ps 1'
  End If
Case ('1e')
  Layout = 'lines ls 5'
  If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls', Index
  If (Present(Index)) Then
    Write(10, '(A, I2), (A)') 'set style line ', Index, ' lc rgb "magenta" lt 1 lw 2 pt 7 ps 1'
  Else
    Write(10, '(A)') 'set style line 5 lc rgb "magenta" lt 1 lw 2 pt 7 ps 1'
  End If
Case ('1f')
  Layout = 'lines ls 6'
  If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls', Index
  If (Present(Index)) Then
    Write(10, '(A, I2), (A)') 'set style line ', Index, ' lc rgb "sandybrown" lt 1 lw 2 pt 7 ps 1'
  Else
    Write(10, '(A)') 'set style line 6 lc rgb "sandybrown" lt 1 lw 2 pt 7 ps 1'
  End If
Case ('1g')
  Layout = 'lines ls 7'
  If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls', Index
  If (Present(Index)) Then
    Write(10, '(A, I2), (A)') 'set style line ', Index, ' lc rgb "orange-red" lt 1 lw 2 pt 7 ps 1'
  Else
    Write(10, '(A)') 'set style line 7 lc rgb "orange-red" lt 1 lw 2 pt 7 ps 1'
  End If
Case ('1h')
  Layout = 'lines ls 8'
  If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls', Index
  If (Present(Index)) Then
    Write(10, '(A, I2), (A)') 'set style line ', Index, ' lc rgb "spring-green" lt 1 lw 2 pt 7 ps 1'
  Else
    Write(10, '(A)') 'set style line 8 lc rgb "spring-green" lt 1 lw 2 pt 7 ps 1'
  End If
Case ('1i')
  Layout = 'lines ls 9'
  If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls', Index
  If (Present(Index)) Then
    Write(10, '(A, I2), (A)') 'set style line ', Index, ' lc rgb "cyan" lt 1 lw 2 pt 7 ps 1'
  Else
    Write(10, '(A)') 'set style line 9 lc rgb "cyan" lt 1 lw 2 pt 7 ps 1'
  End If
Case ('1j')
  Layout = 'lines ls 10'
  If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls', Index
  If (Present(Index)) Then
    Write(10, '(A, I2), (A)') 'set style line ', Index, ' lc rgb "khaki" lt 1 lw 2 pt 7 ps 1'
  Else
    Write(10, '(A)') 'set style line 10 lc rgb "khaki" lt 1 lw 2 pt 7 ps 1'
  End If
Case ('1k')    ! Reddish
  Layout = 'lines ls 11'
  If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls', Index
  If (Present(Index)) Then
    Write(10, '(A, I2), (A)') 'set style line ', Index, ' lc rgb "#dd181f" lt 1 lw 3 pt 7 ps 1'
  Else
    Write(10, '(A)') 'set style line 11 lc rgb "#dd181f" lt 1 lw 3 pt 7 ps 1'
  End If
Case ('1l')
  Layout = 'lines ls 12'
  If (Present(Index)) Write(Layout, '(A, I2)') 'lines ls', Index
If (Present(Index)) Then
  Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "dark-blue" lt 1 lw 3 pt 7 ps 1'
Else
  Write(10,'(A)') 'set style line 12 lc rgb "dark-blue" lt 1 lw 3 pt 7 ps 1'
End If

Case ('lm')
  Layout = 'lines ls 13'
  If (Present(Index)) Write(Layout,'(A,I2)') 'lines ls ',Index
  If (Present(Index)) Then
    Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "dark-blue" lt 1 lw 3 pt 7 ps 1'
  Else
    Write(10,'(A)') 'set style line 12 lc rgb "dark-blue" lt 1 lw 3 pt 7 ps 1'
  End If

Case ('ln')
  Layout = 'lines ls 14'
  If (Present(Index)) Write(Layout,'(A,I2)') 'lines ls ',Index
  If (Present(Index)) Then
    Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "dark-green" lt 1 lw 3 pt 7 ps 1'
  Else
    Write(10,'(A)') 'set style line 13 lc rgb "orange" lt 1 lw 3 pt 7 ps 1'
  End If

Case ('lo')
  Layout = 'lines ls 15'
  If (Present(Index)) Write(Layout,'(A,I2)') 'lines ls ',Index
  If (Present(Index)) Then
    Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "magenta" lt 1 lw 3 pt 7 ps 1'
  Else
    Write(10,'(A)') 'set style line 14 lc rgb "magenta" lt 1 lw 3 pt 7 ps 1'
  End If

Case ('lp')
  Layout = 'lines ls 16'
  If (Present(Index)) Write(Layout,'(A,I2)') 'lines ls ',Index
  If (Present(Index)) Then
    Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "sandybrown" lt 1 lw 3 pt 7 ps 1'
  Else
    Write(10,'(A)') 'set style line 15 lc rgb "sandybrown" lt 1 lw 3 pt 7 ps 1'
  End If

Case ('lq')
  Layout = 'lines ls 17'
  If (Present(Index)) Write(Layout,'(A,I2)') 'lines ls ',Index
  If (Present(Index)) Then
    Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "orange-red" lt 1 lw 3 pt 7 ps 1'
  Else
    Write(10,'(A)') 'set style line 16 lc rgb "orange-red" lt 1 lw 3 pt 7 ps 1'
  End If

Case ('lr')
  Layout = 'lines ls 18'
  If (Present(Index)) Write(Layout,'(A,I2)') 'lines ls ',Index
  If (Present(Index)) Then
    Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "spring-green" lt 1 lw 3 pt 7 ps 1'
  Else
    Write(10,'(A)') 'set style line 17 lc rgb "spring-green" lt 1 lw 3 pt 7 ps 1'
  End If

Case ('ls')
  Layout = 'lines ls 19'
  If (Present(Index)) Write(Layout,'(A,I2)') 'lines ls ',Index
  If (Present(Index)) Then
    Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "cyan" lt 1 lw 3 pt 7 ps 1'
Else
Write(10,'(A)') 'set style line 19 lc rgb "cyan" lt 1 lw 3 pt 7 ps 1'
End If

Case('1t')
Layout = 'lines ls 20'
If (Present(Index)) Write(Layout,'(A,I2)') 'lines ls ',Index
If (Present(Index)) Then
Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "khaki" lt 1 lw 3 pt 7 ps 1'
Else
Write(10,'(A)') 'set style line 20 lc rgb "khaki" lt 1 lw 3 pt 7 ps 1'
End If

Case('1u')
Layout = 'lines ls 21'
If (Present(Index)) Write(Layout,'(A,I2)') 'lines ls ',Index
If (Present(Index)) Then
Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "bisque" lt 1 lw 3 pt 7 ps 1'
Else
Write(10,'(A)') 'set style line 21 lc rgb "bisque" lt 1 lw 3 pt 7 ps 1'
End If

Case('1v')
Layout = 'lines ls 22'
If (Present(Index)) Write(Layout,'(A,I2)') 'lines ls ',Index
If (Present(Index)) Then
Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "plum" lt 1 lw 3 pt 7 ps 1'
Else
Write(10,'(A)') 'set style line 22 lc rgb "plum" lt 1 lw 3 pt 7 ps 1'
End If

Case('1w')
Layout = 'lines ls 23'
If (Present(Index)) Write(Layout,'(A,I2)') 'lines ls ',Index
If (Present(Index)) Then
Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "medium-blue" lt 1 lw 3 pt 7 ps 1'
Else
Write(10,'(A)') 'set style line 23 lc rgb "medium-blue" lt 1 lw 3 pt 7 ps 1'
End If

Case('1x')
Layout = 'lines ls 24'
If (Present(Index)) Write(Layout,'(A,I2)') 'lines ls ',Index
If (Present(Index)) Then
Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "dark-orange" lt 1 lw 3 pt 7 ps 1'
Else
Write(10,'(A)') 'set style line 24 lc rgb "dark-orange" lt 1 lw 3 pt 7 ps 1'
End If

Case('1y')
Layout = 'lines ls 25'
If (Present(Index)) Write(Layout,'(A,I2)') 'lines ls ',Index
If (Present(Index)) Then
Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "dark-plum" lt 1 lw 3 pt 7 ps 1'
Else
Write(10,'(A)') 'set style line 25 lc rgb "dark-plum" lt 1 lw 3 pt 7 ps 1'
End If

Case('1z')
Layout = 'lines ls 26'
If (Present(Index)) Write(Layout,'(A,I2)') 'lines ls ',Index
If (Present(Index)) Then
Write(10,'(A,I2,A)') 'set style line ',Index,' lc rgb "dark-goldenrod" lt 1 lw 3 pt 7 ps 1'
Else
Write(10,'(A)') 'set style line 26 lc rgb "dark-goldenrod" lt 1 lw 3 pt 7 ps 1'
End If
Case ('2a')
   Layout = 'lp ls 1'
   Write (10, '(A)') 'set style line 1 lc rgb "red" lt 1 lw 2 pt 7 ps 1'

Case ('2b')
   Layout = 'lp ls 2'
   Write (10, '(A)') 'set style line 2 lc rgb "dark-blue" lt 1 lw 2 pt 7 ps 1'

Case ('2c')
   Layout = 'lp ls 3'
   Write (10, '(A)') 'set style line 3 lc rgb "orange" lt 1 lw 2 pt 7 ps 1'

Case ('2d')
   Layout = 'lp ls 4'
   Write (10, '(A)') 'set style line 4 lc rgb "dark-blue" lt 1 lw 3 pt 7 ps 2'

Case ('2e')
   Layout = 'lp ls 5'
   Write (10, '(A)') 'set style line 5 lc rgb "dark-red" lt 1 lw 2 pt 7 ps 1'

Case ('2f')
   Layout = 'lp ls 6'
   Write (10, '(A)') 'set style line 6 lc rgb "orange-red" lt 1 lw 2 pt 7 ps 1'

Case ('2g')
   Layout = 'lp ls 7'
   Write (10, '(A)') 'set style line 7 lc rgb "tan1" lt 1 lw 2 pt 7 ps 1'

Case ('2h')
   Layout = 'lp ls 8'
   Write (10, '(A)') 'set style line 8 lc rgb "dark-green" lt 1 lw 2 pt 7 ps 1'

Case ('2i')
   Layout = 'lp ls 9'
   Write (10, '(A)') 'set style line 9 lc rgb "green" lt 1 lw 2 pt 7 ps 1'

Case ('2j')
   Layout = 'lp ls 10'
   Write (10, '(A)') 'set style line 10 lc rgb "chartreuse" lt 1 lw 2 pt 7 ps 1'

Case ('2k')
   Layout = 'lp ls 11'
   Write (10, '(A)') 'set style line 11 lc rgb "dark-goldenrod" lt 1 lw 2 pt 7 ps 1'

Case ('2l')
   Layout = 'lp ls 12'
   Write (10, '(A)') 'set style line 12 lc rgb "orange" lt 1 lw 2 pt 7 ps 1'

Case ('3a')
   Layout = 'points ls 1'
   Write (10, '(A)') 'set style line 1 lc rgb "red" lt 1 lw 3 pt 6 ps 2'

Case ('3b')
   Layout = 'points ls 2'
   Write (10, '(A)') 'set style line 2 lc rgb "dark-blue" lt 1 lw 3 pt 6 ps 2'

Case ('3c')
Case ("3u")
    Layout = 'points ls 21'
    Write(10,'(A)') 'set style line 21 lc rgb "dark-violet" lt 1 lw 1 pt 7 ps 3'

Case ("3v")
    Layout = 'points ls 22'
    Write(10,'(A)') 'set style line 22 lc rgb "dark-green" lt 1 lw 3 pt 6 ps 2'

Case ("3w")
    Layout = 'points ls 23'
    Write(10,'(A)') 'set style line 23 lc rgb "orange-red" lt 1 lw 1 pt 7 ps 3'

Case ("3x")
    Layout = 'points ls 24'
    Write(10,'(A)') 'set style line 24 lc rgb "chartreuse" lt 1 lw 3 pt 6 ps 2'

Case ("3y")
    Layout = 'points ls 25'
    Write(10,'(A)') 'set style line 25 lc rgb "dark-red" lt 1 lw 3 pt 6 ps 2'

Case ("3z")
    Layout = 'points ls 26'
    Write(10,'(A)') 'set style line 26 lc rgb "dark-green" lt 1 lw 3 pt 6 ps 2'

Case ("4a")
    Layout = 'circles ls 1'
    Write(10,'(A)') 'set style line 1 lc rgb "gray" lw 3'

Case ("4b")
    Layout = 'circles ls 2'
    Write(10,'(A)') 'set style line 2 lc rgb "red" lw 3'

Case ("4c")
    Layout = 'circles ls 3'
    Write(10,'(A)') 'set style line 3 lc rgb "light-green" lw 3'

Case ("4d")
    Layout = 'circles ls 4'
    Write(10,'(A)') 'set style line 4 lc rgb "pink" lw 3'

Case ("4e")
    Layout = 'circles ls 5'
    Write(10,'(A)') 'set style line 5 lc rgb "dark-gray" lw 3'

Case ("4f")
    Layout = 'circles ls 6'
    Write(10,'(A)') 'set style line 6 lc rgb "dark-blue" lw 3'

Case ("4g")
    Layout = 'circles ls 7'
    Write(10,'(A)') 'set style line 7 lc rgb "dark-yellow" lw 3'

Case ("4h")
    Layout = 'circles ls 8'
    Write(10,'(A)') 'set style line 8 lc rgb "orange" lw 3'

Case ("5a")
    Layout = 'boxes ls 1'
    Write(10,'(A)') 'set style lines 1 lc rgb "red" lw 3'

Case ("5b")
    Layout = 'boxes ls 2'
    Write(10,'(A)') 'set style lines 2 lc rgb "dark-blue" lw 3'

Case Default
    Write(*,'(A)') "Graph Style unknown, process aborted."
    Return
End Select
End Subroutine

End Module VisualModule

Code/VisualModule.f90