SESAM USER MANUAL

PILSS

Pipelines, umbilicals, cables and flexibles dynamic Lateral Stability Software

Valid from program version 2.3





Sesam User Manual

PILSS 2.3

Date: October 2024, Revision A

Valid from version 2.3

Prepared by: Digital Solutions at DNV

E-mail support: software.support@dnv.com

E-mail sales: digital@dnv.com

© DNV AS. All rights reserved

This publication or parts thereof may not be reproduced or transmitted in any form or by any means, including copying or recording, without the prior written consent of DNV AS.

Table of contents

1 INTRODUCTION 1
1.1 PILSS – Dynamic lateral stability analysis 1
1.2 PILSS in the Sesam System 1
1.3 Status list and Release notes 2
1.4 How to read the manual 2
1.5 Acronyms and symbols frequently used in the User Manual 2

2 FEATURES OF PILSS 3
2.1 Summary 3
2.2 Highlighted features 4

3 GETTING STARTED WITH PILSS 5
3.1 Units 5
3.2 User Interface 5
3.3 Performing an analysis 6
3.4 Convergence and design value 6
3.5 Warnings and Errors 6
3.6 Input from Case File 7
3.7 Output to files 7
3.8 Getting help 8
3.9 Design premise 8
3.10 Limitations of PILSS 8

4 PILSS GUI 11
4.1 Main sheet input 11
4.2 Viewing Results Output 18
4.3 BatchRuns 19

5 PILSS CLI 20
5.1 General 20
5.2 Input 20

6 PILSS API 24

APPENDIX A REFERENCES 25

APPENDIX B EXERCISE: PIPELINE ON SANDBED 26

APPENDIX C DLL CALL - DETAILS 33

1 INTRODUCTION

1.1 PILSS – Dynamic lateral stability analysis

Pipelines, umbilicals, cables, and flexibles dynamic Lateral Stability Software (PILSS) is a calculation tool for dynamic time-domain analyses of lateral on-bottom stability for submarine pipelines, umbilicals, cables and flexibles, developed by DNV and compliant with DNV-RP-F109 /1/.

At PILSS' core, you'll find an efficient dynamic solver calculation engine written in Fortran.

The PILSS calculation engine can be operated through a Graphical User Interface (GUI), a command-based user interface, a "Command Line Interface" (CLI), or directly through an Application Programming Interface (API) from a 3rd party software or script.

In short, PILSS calculates how far a submarine line will move laterally on the seabed during a design storm, in accordance with DNV-RP-F109 /1/.

PILSS is developed by DNV and was initially developed as part of the PILS JIP, ref. /1/ and /11/.

Section 2 summarize features provided by PILSS.

1.2 PILSS in the Sesam System

The Sesam software family is comprised of pre-processors, environmental analysis programs, structural analysis programs and postprocessors. An overview of Sesam is shown in Figure 1-1.

PILSS belongs to the Pipeline Tools group (red circle), which consists of several software that provide solutions to pipeline design and operation.

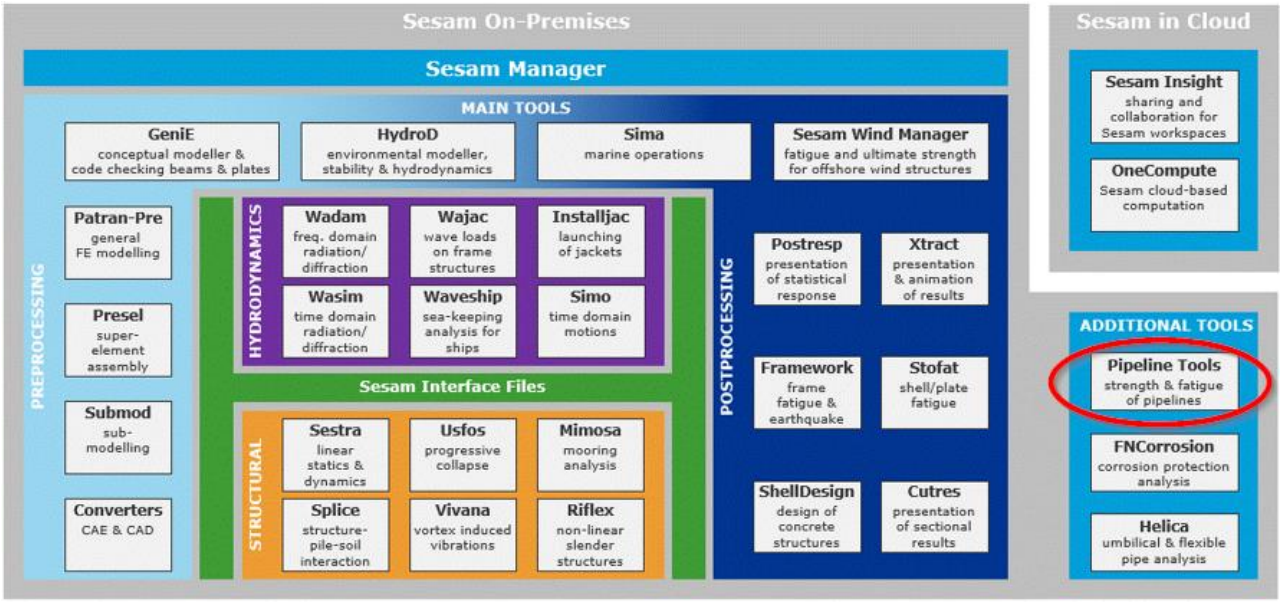


Figure 1-1 Sesam Overview

1.3 Status list and Release notes

In addition to this User Manual, you may find additional information in the PILSS Release Notes, which is updated for each new release. There is also an online Status List. This contains information such as:

- Reasons for update (new version)
- New features
- Errors found and corrected

1.4 How to read the manual

- Read Section 2 to get an overview of PILSS features.
- Read Section 3 to get started using PILSS.
- Read Section 4 to get more detail on how to operate the PILSS GUI.
- Read Section 5 to get more detail on how to operate the PILSS CLI.
- Read Section 6 for information on how to interface directly with the PILSS API.

1.5 Acronyms and symbols frequently used in the User Manual

The table below lists a few acronyms that are frequently used in the User Manual.

API	Application Programming Interface
CLI	Command Line Interface
FJC	Field Joint Coating
GUI	Graphical User Interface
JIP	Joint Industry Project
JONSWAP	JOint North Sea WAve Project
PILS	Pipeline Lateral Stability
PILSS	Pipeline Lateral Stability Software
SI	Système International d'Unités
VBA	Visual Basic for Applications

2 FEATURES OF PILSS

2.1 Summary

PILSS provides the following features:

- Dynamic solver
  - Time-domain dynamic lateral on-bottom stability analyses for submarine pipelines, umbilicals and cables.
  - High speed calculations
- Wave kinematics
  - Short term, stationary and irregular sea states are modelled based on the JONSWAP spectrum.
  - Generators for linear and 2nd order irregular wave time series
  - Wave induced particle velocities at seabed level are automatically calculated based on user defined water depth (d), significant wave height (H_s) and peak period (T_p).
  - Wave heading and spreading are accounted for
  - Generator for regular wave time series
  - Multiple seed run with the click of a button
- Hydrodynamic load models
  - Hydrodynamic loads are calculated based on generated or user-defined wave time series and user defined steady current
- Soil resistance models
  - Three different soil types: Coulomb friction, Sand and Clay
  - Three different models for calculation of initial penetration for Clay
  - Effect of installation on initial penetration for Sand and Clay
  - Effect of water-filled pipe on initial penetration for Sand and Clay
- Pre-processing
  - User-friendly interface with context-based highlighting and validation checks of input data
  - Batch run module
- Post-processing
  - Automated, but customizable plot tool for quick visualization of results
  - Optional full resolution time-series results to file
- User interface
  - Graphical User Interface (GUI) based on Excel
  - Command Line Interface (CLI)
  - Application Programming Interface (API)
  - 32-bit and 64-bit compatibility
  - Professional installer
  - Robust, flexible, and user-friendly licensing functionality

2.2 Highlighted features

- Wave kinematics
    - Irregular wave generator capable of generating first order (linear) and second order timeseries for wave-induced particle velocity and acceleration at pipe level. The generator is based on the well-tested WABIRK software /3/.
    - Short term, stationary and irregular sea states are modelled based on the JONSWAP wave spectrum. JONSWAP ("JOint North Sea WAve Project") is a peak-enhanced Pierson-Moskowitz (single-peaked) spectrum for wind-driven sea states.
- Hydrodynamic load models
    - Hydrodynamic load model based on a Fourier coefficient database derived through extensive experimental campaigns /4/, /5/ and /6/ . The model accounts for wake effects and force amplification after flow reversal, giving significantly improved force predictions compared to conventional constant-coefficient Morison formulations.
    - Empirical method to consider the influence on hydrodynamic loading from the pipe or cable's own velocity and acceleration /6/.
- Soil resistance models
    - Dynamic pipe-soil interaction modelling accounting for the build-up of pipe penetration and passive soil resistance, with different models available for sand and clay soils. A pure Coulomb friction model is also included.
    - A well established and extensively used pipe-sand model developed by Verley and Sotberg /7/.
    - A well established and extensively used pipe-clay model developed by Verley and Lund /8/, with important updates compared to previous implementations.
    - Two new pipe-clay models with initial pipe penetration and break-out resistance calculated according to the geotechnical model in DNV-RP-F114 /2/ and the recent SAFEBUCK JIP recommendations /9/.
    - Option to include a lay factor to account for enhanced initial penetration due to lay effects. The lay factor calculation is based on a recent method proposed by Randolph and White /10/.

3 GETTING STARTED WITH PILSS

3.1 Units

All input and output parameters presented in the main sheet are in SI units.

3.2 User Interface

3.2.1 Graphical User Interface (GUI)

PILSS GUI (Section 4) is a Microsoft Excel Visual Basic (VBA) program. It is based on Microsoft Excel 2016 and the user interface is the same as in a typical Excel spreadsheet[1] with direct input into cells, buttons, and pull-down menus.

The program has been designed so that the input and results are located on the Main sheet. The user can see all the necessary information from this sheet without navigating between windows. There are three kinds of sheets available within PILSS:

- Main – contains all the input and the most important output data.
- Plots – contains results for graphical presentation and additional useful results.
- BatchRuns – used to run multiple cases in series.

The Main sheet has been designed so that the user can modify input cells only. This is to prevent corruption of the input data, formulae and the routines. Cells with input access are white, whereas cells that do not allow user access are coloured light yellow, see Figure 4-1. Some of the cells change access rights according to specific selections. This occurs when some of the User Options have been changed.

3.2.2 Command Line Interface (CLI)

PILSS CLI (Section 5) performs the exact same calculations as PILSS GUI, through the library PILSS_DLL.dll. As the CLI is command based, it is slightly faster than the GUI, but less intuitive. Input to the CLI must be provided through a PILSS Case File (.pls file). A Case File template is shipped with the PILSS Software or can be generated by the PILSS GUI.

PILSS CLI can read wave data and hydrodynamic force data from text files (.txt). Note that the formatting of the data needs to match that of PILSS, which corresponds to Fortran format specification E25.15. Template files are shipped with the PILSS Software. More details are given in Section 5.2.

3.2.3 Application Programming Interface (API)

The PILSS dynamic stability solver library (PILSS_DLL.dll) can also be called directly through a simple API (Application Programming Interface), see Section 6. Using this, it is possible to integrate the PILSS dynamic stability calculations into a 3rd party application, e.g., a script or a different GUI.

[1] Excel Macro-Enabled Workbook file (.xlsm) which requires VBA macros to be enabled to function properly. Excel Macro Settings are typically controlled by the company IT Administrator and can be viewed through Excel -> Options -> Trust Center -> Trust Center Settings... -> Macro Settings. Some may require that macro-enabled workbooks are copied to and opened from a trusted folder to allow for macros to be enabled.
Note that both Excel GUI and backend DLLs in any DNV Sesam Pipeline Tool are signed by DNV (or DNV GL for older versions) with digital certificates originating from DigiCert. These certificates guarantees that no code has been tampered with since it was signed by DNV. For example, if any changes were done to any macros in a GUI Spreadsheet, Excel would automatically invalidate the digital signature and the certificate would be revoked from this particular Excel file.

3.3 Performing an analysis

The procedure to follow when assessing a single on-bottom stability case is briefly described below:

1. Specify input options
2. Specify relevant input parameters:
    a. Define cross-section
    b. Define soil variables
    c. Define variables for hydrodynamic load generation
    d. Define number of seeds and generate random seeds (if applicable)
3. Specify relevant analysis parameters:
    a. Define time series
4. Click the “Run PILSS” button
5. The main results are printed to the Main sheet
6. Plots and more detailed results are given in the “Plots” sheet
7. [Optional] Detailed results and wave time series are written to file

The results are presented on the Main sheet in terms of total design displacement (in metres), which is the sum of the average displacement and one standard deviation (functions of number of seed runs). The design displacement is also presented as function of outer diameter, to allow for direct comparison with the simplified design methodologies presented in DNV-RP-F109 /1/.

Parameter studies may be performed using the BatchRuns function.

An example exercise demonstrating basic usage of PILSS is provided in Appendix B.

3.4 Convergence and design value

DNV-RP-F109 /1/ recommends that the mean value plus one standard deviation should be used as design value when the standard deviation in the displacement resulting from dynamic lateral stability analysis has stabilized. This can be validated visually in the PILSS Main sheet.

In addition to stabilizing the standard deviation, a convergence check should be performed to ensure that a sufficiently small time step has been used in the dynamic analysis.

As a rule of thumb, PILSS typically converges on total design displacement with the following time steps, but this needs to be checked by the user in each design case:

- Coulomb: 0.01 s
- Sand: 0.001 s
- Clay: 0.001 s

3.5 Warnings and Errors

3.5.1 GUI

When the “Run PILSS” button is clicked, PILSS reads input parameters and performs a validation check on these. If a parameter is outside its validity range a Warning or Error message pops up providing information on which parameter that is not valid.

PILSS differs between three types of messages to the user:

- **Guide** message: Interactive message that requires a response from the user (typically ‘Yes’, ‘No’ or ‘Cancel’)
- **Warning** message: The calculation is in some way outside recommended validity ranges; however, the calculation is completed and an indicative result is provided.
- **Error** message: The calculation is in some way outside a specific validity range or there exists a false input value, and the calculation is terminated.

Please note that Warning and Guide messages are muted during BatchRuns calculations, This is to allow lengthy batchruns to complete unsupervised unless fatal errors are encountered.

3.5.2 CLI

The command line-based user interface does currently not provide the same validation checks of input parameters as the GUI does.

Critical error and warning messages related to e.g. reading/writing to file are written to the command line when running the CLI.

3.5.3 API

The API does currently not provide any validation checks of input parameters as the GUI does.

Critical error messages e.g. reading/writing to file are thrown by the Fortran Run-Time Library.

3.6 Input from Case File

3.6.1 GUI

Input parameters for different design cases may be read from PILSS Case Files (.pls). The Case Files may be generated by PILSS GUI using the “Save case” button, or by manually editing existing Case Files using any text editor.

A Case File template, “PILSS_CaseFile.pls” is located in the *example* folder in the PILSS installation folder. Input data is read from the Case File by using the “Open case” button.

Another second case file is also included, representing the example exercise presented in Appendix B.

3.6.2 CLI

Reading input data from the Case File is mandatory when running PILSS CLI. Note that the CLI offers very little input validation, the user is responsible for providing a valid .pls file. Use the PILSS GUI to aid generation of .pls files if in doubt.

3.6.3 API

The API is intended to communicate with its caller through the system memory directly.

3.7 Output to files

PILSS always generates the following output to file:

- PILSS_**log_[processID]**.txt: Contains condensed run-time information from PILSS_DLL.dll. This is written to the %AppData%\DNV\PILSS folder. One log file is written per unique process and is over-written for every run. This makes PILSS “process safe”, meaning that multiple PILSS processes can run simultaneously, but not explicitly thread safe.

PILSS generates the following output files if requested by the user:

- [CaseName]_**ResultsOverview**.txt: Contains main results and an echo of input data.
- [CaseName]_**Results_Seed**_*X*.txt: Contains detailed results in the form of time series. The resolution of the time series is governed by the input parameter $\Delta t_{results}$. One file is written for each seed run.
- [CaseName]_**WaveCurrentData**.txt: Contains full free stream wave velocity and acceleration time series for all seed runs.

If a case name is not given by user, the default case name «PILSS» is used.

The output files are written to a folder specified by the user. Note that all these files are overwritten each time PILSS is run unless the files are renamed or moved to a different folder.

3.8 Getting help

3.8.1 GUI

Comments are provided in many of the cells to give further guidance to the user. The comments are identified by a small red triangle in the top right-hand corner of the cell and are viewed by pointing at the cell.

3.8.2 CLI

The CLI has built-in help functionality. Just type `pilss -h` in the PILSS CLI command line environment.

3.8.3 General

For further assistance, please contact DNV. Contact information is provided on the second page of this User Manual.

3.9 Design premise

Please refer to DNV-RP-F109 /1/ and the PILSS Technical Manual for guidance on how to perform dynamic lateral on-bottom stability analyses.

In short, PILSS calculates how far a submarine line will move laterally on the seabed during a design storm.

In the current version of PILSS, the submarine product is modelled as a point in space, or equivalently, as an inifitely long, straight product on a flat seabed. This also a fundamental model assumption behind the simplified methods in DNV-RP-F109 /1/.

3.10 Limitations of PILSS

3.10.1 Limitation to design parameters

PILSS GUI will give warnings when specific validity ranges are violated or in some cases where the input is clearly unphysical. For example, to use the implemented method for including a lay factor in the calculation of initial penetration, the lay tension must be greater than a lower bound value depending on the bending stiffness and submerged weight during installation. Typical parameter ranges are also indicated in the PILSS Technical Manual for several of the parameters entering the Geotechnical model for clay, and a message is given to the user if an input value is outside these ranges.

Note also that end constraints, i.e., effects of tension and bending moments due to tie-in points, are not handled by PILSS.

3.10.2 On alignment with DNV-RP-F109

PILSS incorporates the design principles for dynamic lateral stability analysis according to DNV-RP-F109 /1/.

3.10.3 On umbilicals, cables and small diameter pipelines

The basic design approaches in DNV-RP-F109 /1/ builds on experiments and analyses performed in the 1980s prior to issuing DNV-RP-E305 in 1988 and on pipe-soil interaction models developed in the early 1990s. To DNV's knowledge, the experiments that were performed to establish load models were performed on cylinders with a diameter range of 300 mm to 600 mm, while the soil models were based on tests with cylinder diameters ranging from 200 mm (clay) or 300 mm (sand) up to 1 m. From a strict scientific point of view, this is thus the applicability range for the method. However, this approach, as described in DNV-RP-E305 and DNV-RP-F109, has been used for pipelines with larger diameters for almost 30 years and has proven to be a sound approach. The experience with smaller diameter umbilicals and cables is much shorter, but DNV has no reason to believe that the approach should not be sufficiently conservative for smaller diameter pipelines, umbilicals and cables.

Finally, it should be noted that the design methods recommended in DNV-RP-F109 leads in general to much higher specific gravity, i.e. weight divided by volume, for small diameters which also indicates that the methods are valid for these cases.

3.10.4 On linear wave theory

Note that the generation of time series for wave-induced water particle velocity and acceleration in PILSS is based on either linear wave theory or second order wave theory. Some combinations of significant wave height H_s, peak period T_p and water depth d may represent sea states that are not adequately described by linear or second order wave theory. Whether linear or second wave theory is applicable for the given input data must be validated by the user. Some guidance on this matter may be found in DNV-RP-C205, ref. /12/.

3.10.5 On diameter and marine growth

The outer umbilical/cable/pipeline diameter D including all relevant coating layers must always be specified as an input parameter to PILSS.

If the checkbox “User defined hydrodynamic diameter” is checked, the user may also specify a hydrodynamic diameter D_{hydro}. The hydrodynamic diameter D_{hydro} may be different from D.

Marine growth thickness, denoted t_{mgrt}, must always be specified as an input parameter to PILSS (and shall be set to zero if marine growth is not relevant). In the presence of marine growth, hydrodynamic loads are calculated using an equivalent diameter D_{eq} and reduction factors $r_{pen,y}$ and $r_{pen,z}$ based on an equivalent penetration z_{eq}, as recommended in DNV-RP-F109 /1/ and the PILSS Technical Manual.

The equivalent diameter D_{eq} is calculated by PILSS as

$$D_{eq} = \begin{cases} D_{hydro} + 2t_{mgrt} & \text{if "User defined hydrodynamic diameter" is checked} \\ D + 2t_{mgrt} & \text{if "User defined hydrodynamic diameter" is not checked} \end{cases}$$

The equivalent penetration z_{eq} is calculated by PILSS as

$$z_{eq} = z_p + t_{mgrt}$$

Hydrodynamic loads and reduction factors due to penetration into the soil are always calculated in PILSS using D_{eq} and z_{eq}. Soil resistance forces are on the other hand always calculated using D and the actual penetration z_p calculated for the relevant timestep.

3.10.6 On mass, submerged weight and buoyancy

Both the mass of the umbilical/cable/pipeline in air, denoted m_{pipe}, and the submerged weight of the umbilical/cable/pipeline, denoted w_s, must be specified as input parameters to PILSS.

The mass m_{pipe} and submerged weight w_s shall reflect the mass/weight of the pipe in the condition considered for the design storm simulation. Hence, content mass/weight shall be included if relevant.

Mass/weight of marine growth should be included in a conservative manner as described in Section 3.10.5.

Hydrodynamic added mass is calculated as part of the hydrodynamic load calculation in PILSS and shall not be included in the input value for m_{pipe}. PILSS calculates hydrodynamic added mass assuming a circular cross-section with diameter D_{eq} and an empirically derived added mass coefficient of 1.5, see the PILSS Technical Manual for further details.

The database force model used to calculate hydrodynamic loads in PILSS is derived for circular cross-sections. If PILSS is used to assess lateral stability of non-circular cross-sections (e.g. piggybacked pipelines) the user should investigate the sensitivity to the value of m_{pipe} and adjust its value to correct for misrepresentation of the hydrodynamic added mass if relevant. Also see DNV-RP-F109 /1/.

As umbilicals and cables are sometimes permeable to seawater, the mass and submerged weights of these may in principle change with time. This needs to be handled by the user of PILSS. PILSS calculates buoyancy based on the total outer diameter of umbilical/cable/pipeline.

A warning message is given to the user if the pipe becomes buoyant during an analysis, i.e., if the lift force at any point exceeds the submerged umbilical/cable/pipeline weight during a design storm simulation.

3.10.7 On initial penetration

If the checkbox “User defined initial penetration” is checked, the initial penetration z_{pi} is given by the user as an input, and PILSS does not calculate initial penetration.

If the checkbox “Include lay factor” is checked, the user may either specify, or instruct PILSS to calculate, a static touchdown lay factor k_{lay} that accounts for increased vertical contact force during installation. The user must specify the submerged weight during installation $w_{s,installation}$ when including a lay factor. The initial penetration is then calculated based on the contact force F_c given by

$$F_c = \max\{w_s \; ; \; k_{lay} \cdot w_{s,installation}\}$$

If the checkbox “Initial penetration water-filled pipe” is checked, the user must specify the waterfilled submerged weight $w_{s,waterfilled}$ as an input parameter and PILSS will calculate the initial penetration based on $w_{s,waterfilled}$ rather than w_s.

If both the checkboxes “Include lay factor” and “Initial penetration water-filled pipe” are checked, the initial penetration is calculated based on the contact force F_c given by

$$F_c = \max\{w_{s,waterfilled} \; ; \; k_{lay} \cdot w_{s,installation}\}$$

Lift forces are not considered for calculation of *initial* penetration.

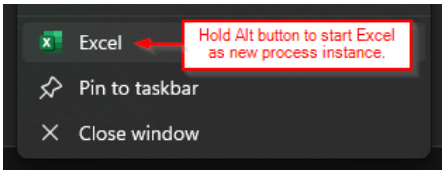
3.10.8 On results dependency of processor architecture

The user should note that negligible differences in results might be observed for identical cases run on different computers. This is due to how the Fortran compiler is configured to optimize floating point operations for speed rather than full double precision consistency. Such code optimization is processor architecture dependent. The largest relative differences found during extensive regression testing of PILSS was in the order of 1E-4, when running on laptops with respectively 8th and 11th generation Intel mobile processors.

3.10.9 Running different PILSS GUI versions in same Excel instance

The user should beware of running different versions of the PILSS Excel GUI workbook in the same Excel instance. Multiple workbooks ran in a single instance of an Excel.exe process share memory so that different versions of PILSS Excel GUI may affect each other. For example, V2.0.0 cannot be run together with V2.2.0 or V2.3.0. The latter two versions can be run together but may affect each other. It has been observed that the randomization of wave time-series realization has been affected so that the second time PILSS is run it effectively realizes the wave time-series with different random seeds than what is given as input.

To run different PILSS GUIs in parallel on the same computer, one should run them in separate Excel instances. To start a new instance of Excel, hold the Alt button while opening Excel the second time:



4 PILSS GUI

4.1 Main sheet input

4.1.1 General

The structure of the Main sheet of PILSS is shown in Figure 4-1. Details are given in the following sub-sections, which are referenced in Figure 4-1.

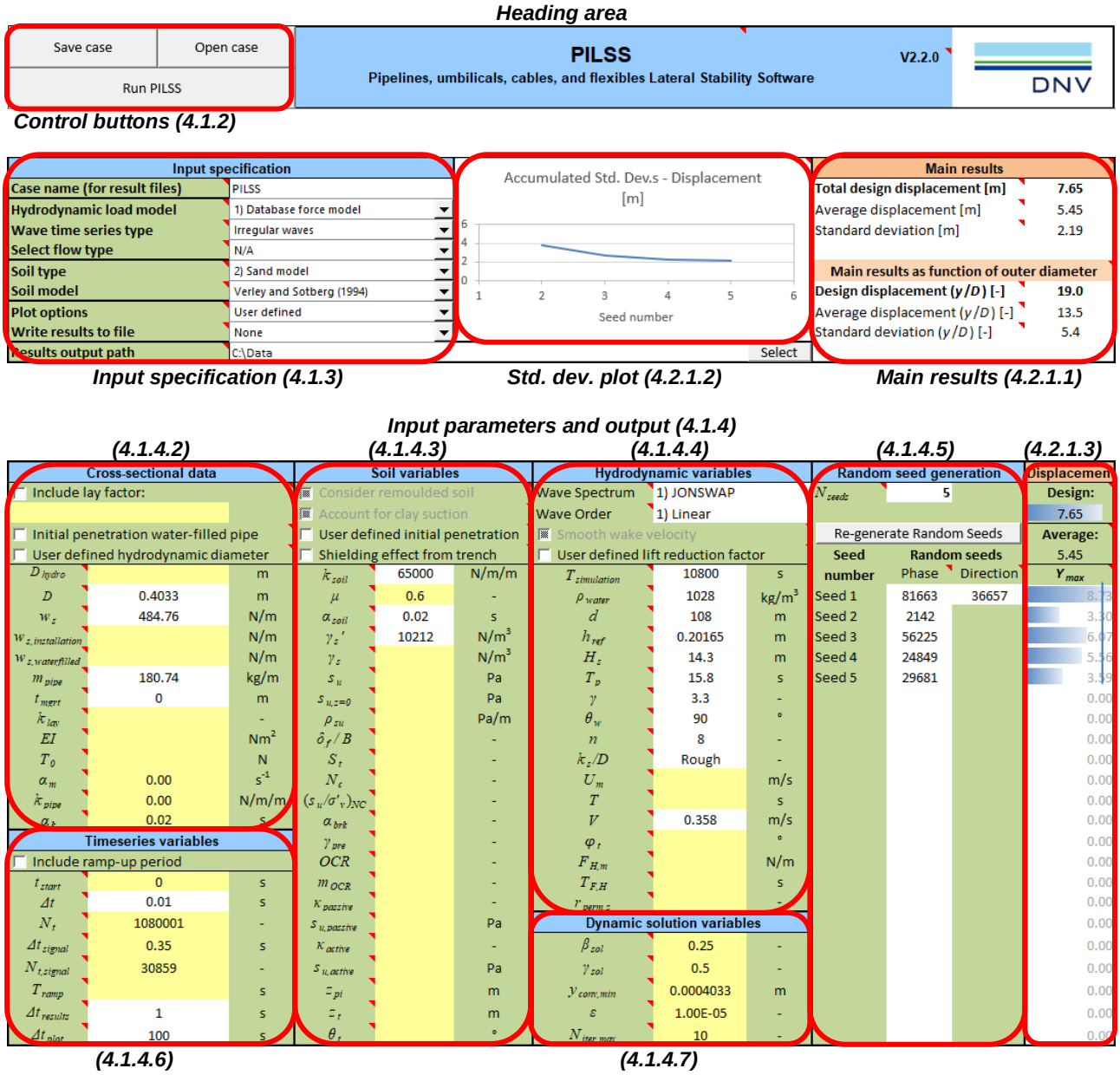


Figure 4-1 Structure of Main sheet

4.1.2 Buttons

Below follows a short description of button functionality in PILSS.

Run PILSS	Performs dynamic lateral stability analyses based on the input specified in the Main sheet.
Save case	Opens a standard Windows Save As dialogue box and allows the user to save the actual case with all input data presented in the main sheet as a PILSS Case File (.pls).
Open case	Opens a standard Windows Open dialogue box and allows the user to populate the Main sheet from a PILSS Case File (.pls) containing all input data required for the specific case.

4.1.3 Input specification

The settings under “Input specification” governs which input cells are open for editing. E.g., if the Sand model is the chosen soil type, input parameters only relevant for clay will be deactivated.

The available input specifications are according to DNV-RP-F109 /1/.

Table 4-1 gives an overview of the different options.

Table 4-1 PILSS Main sheet – Input specification

Specification	Description	Available options*	Default setting
Case name (for result files)	[Optional] Any text in this field will give all result files a prefix. If empty, default prefix will be used, i.e. "PILSS_"	N/A	empty
Hydrodynamic load model	Defines which hydrodynamic load model to be used in the analysis, i.e. how water particle velocities and accelerations are transformed into forces (loads) trying to move the submarine line in subject.	1) Database force model 2) Horizontal sine force	1) Database force model
Wave time series type	Defines what kind of wave induced water particle velocity and acceleration time series to be used in the analysis. Regular waves are sinusoidal waves with constant period and amplitude. Irregular waves resemble the stochastic nature of wave trains during storms in the way that the irregular wave time series are composed of superposed sinusoidal signals with random phase angles and with amplitudes and frequencies selected according to the relevant wave spectrum.	Regular waves Irregular waves	Irregular waves
Select flow type	[Obsolete] Specification of flow type is not applicable for the currently available Hydrodynamic load models.	N/A	N/A
Soil type	Defines which soil type to be modelled, i.e. how the interaction between the submarine line and soil is translated into forces resisting lateral movement of the submarine line. Model 1) only accounts for specified coulomb friction. Model 2) and 3) also accounts for passive resistance forces due to penetration into the seabed in addition to coulomb friction.	1) Coulomb model 2) Sand model 3) Clay model	N/A
Soil model	Defines which model to be used for estimating levels of soil penetration and the magnitude of lateral passive resistance forces. Only applicable for soil type 2) and 3). The three clay models are arranged in order of increasing complexity.	Sand: 1) Verley and Sotberg Clay: 1) Verley and Lund, 2) SAFEBUCK, 3) Geotechnical model	N/A
Plot options	Governs how plots are generated. By default, Δt_{plots} = 1 s. All the other settings are currently defined in the Plots sheet.	Default User defined	Default
Write results to file	If “None”, no results are written to file. If “Overview”, only main results are written to file (“PILSS_ResultsOverview.txt”). If “Detailed”, detailed results, wave velocities and accelerations are also written to file, see Section 3.7.	Detailed Overview None	No

Specification	Description	Available options*	Default setting
Results output path	The user has to specify the output folder. The “Select” button opens a standard Windows folder selection dialogue box and allows the user to specify which folder to write results files to (if this option is activated).	N/A	N/A

*Reference is given to PILSS Technical Manual and DNV-RP-F109 /1/.

4.1.4 Input parameters

4.1.4.1 General

Input parameters that the user needs to specify for a certain analysis are indicated by white cells open for editing. Cells with a light yellow background are either automatically calculated by PILSS or not applicable for the current analysis. This formatting key is illustrated in Figure 4-2.

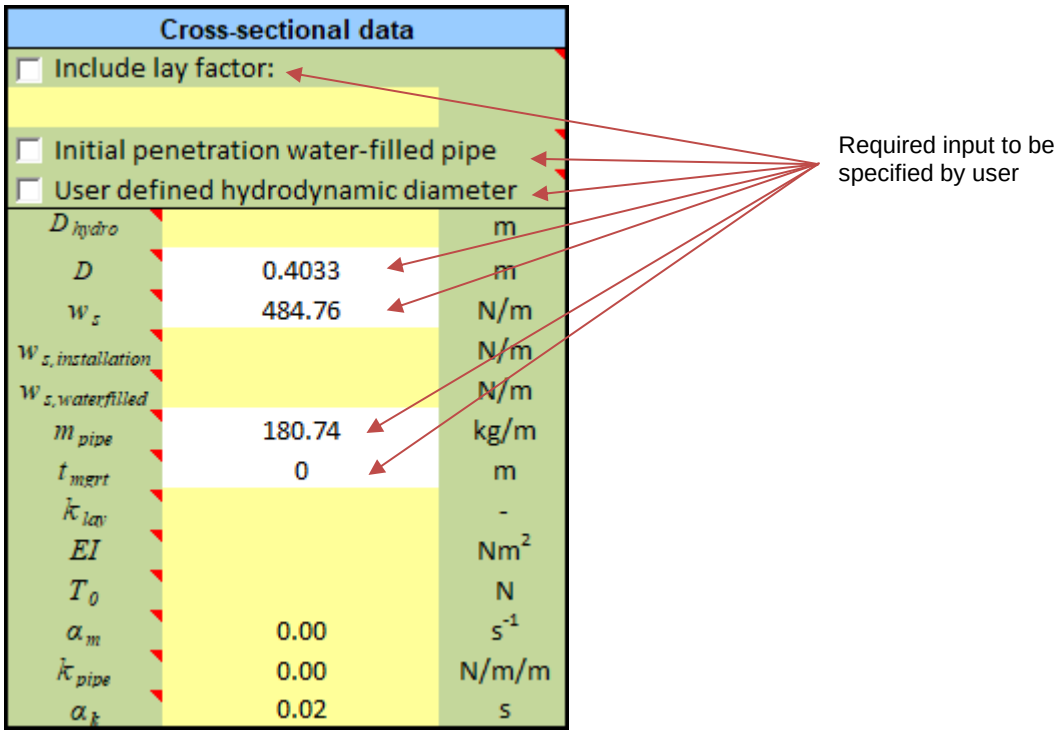


Figure 4-2 White background indicates input parameters required for the current analysis

4.1.4.2 Cross-sectional data

An overview of cross-sectional input data is given in Table 4-2.

Table 4-2 Overview input parameters – Cross-sectional data

Include lay factor	Option for including a lay factor to increase static contact pressure for initial penetration calculation Available options: *1) Lay factor calculated by PILSS* *2) User defined lay factor*
Initial penetration water-filled pipe	Option to account for weight of water-filled pipe when calculating initial penetration
User defined hydrodynamic diameter	Option to specify a user defined hydrodynamic diameter D_{hydro} that is different from the outer pipe/cable diameter D. This may be relevant for instance for piggybacked pipelines.
D_{hydro}	Hydrodynamic diameter of pipe (excluding marine growth)
D	Outer pipe/cable diameter including all coating layers (but excluding marine growth)
w_s	Submerged weight of pipe/cable (including content) in the condition considered for the design storm
$w_{s,inst}$	Submerged weight of pipe/cable during installation. Used if lay factor is included in calculation of initial penetration
$w_{s,waterfilled}$	Submerged weight of pipe/cable in waterfilled condition. Can be used for calculation of initial penetration
m_{pipe}	Mass of pipe/cable including any content (but not including hydrodynamic added mass) in the condition considered for the design storm. Will normally be given by $m_{pipe} = (w_s + b) / g$, where b is the buoyancy and g the acceleration of gravity
t_{mgrt}	Thickness of marine growth
k_{lay}	User defined lay factor
EI	Bending stiffness of pipe/cable. Relevant if lay factor is included in calculation of initial penetration
T_0	Horizontal component of effective lay tension at touch-down point. Relevant if lay factor is included in calculation of initial penetration.
* α_m	Damping coefficient for mass proportional Rayleigh damping ($c_{pipe} = \alpha_m \cdot m_{pipe} + \alpha_k \cdot k_{pipe}$). Currently set to zero
* k_{pipe}	Pipe/cable stiffness against lateral displacement. Currently set to zero
* α_k	Damping coefficient for stiffness proportional Rayleigh damping ($c_{pipe} = \alpha_m \cdot m_{pipe} + \alpha_k \cdot k_{pipe}$). Currently not used since the pipe/cable stiffness is set to zero

*These variables are currently hardcoded in PILSS

4.1.4.3 Soil variables

An overview of soil input data is given in Table 4-3.

Table 4-3 Overview input parameters – Soil variables

Consider remoulded soil	Option for reducing shear strength due to remoulded soil. Only used for Geotechnical clay model
Account for clay suction	Option to account for clay suction. Only used for Geotechnical clay model
User defined initial penetration	Option to specify initial penetration as a user defined input
Shielding effect from trench	Option to reduce hydrodynamic loads for pipelines/cables lying in a trench by including reduction factors according to DNV-RP-F109 /1/
k_{soil}	Elastic stiffness of the soil. Recommended value is 65000 N/m/m
μ	Friction coefficient. Specified by user for Coulomb friction. Set to 0.6 for sand and 0.2 for clay
α_{soil}	Damping coefficient for soil damping in elastic regime ($c_{soil} = \alpha_{soil} \cdot k_{soil}$). As a general rule, setting α_{soil} to 0.02 results in a stable analysis without dominant vibrations.
γ_s'	Submerged (effective) unit weight of soil
γ_s	Total unit soil weight in air. Only used for clay (in Verley and Lund model and after break-out in the other models). Verley and Lund recommend a value of 18000 N/m^3 regardless of the value of submerged unit weight.
s_u	Undrained shear strength of soil. Only used for Verley and Lund clay model (and after break-out in the other clay models)
$s_{u,z=0}$	Undrained shear strength of soil at seabed (z = 0). Only used for SAFEBUCK and Geotechnical clay models
ρ_{su}	Depth gradient of undrained shear strength. Only used for SAFEBUCK and Geotechnical clay models
δ_f / B	Mobilization distance required to reach failure at the relevant penetration depth. Expressed as fraction of the pipe-soil contact width B, which is calculated by the program. Only used for initial penetration calculations according to the SAFEBUCK and Geotechnical clay models. May typically be taken as 0.1, ref. DNV-RP-F114.
S_t	Sensitivity (ratio of intact to remoulded shear strength). Only used for SAFEBUCK and Geotechnical clay models when remoulded soil is considered
N_c	Bearing capacity factor for clay. Only used for initial penetration according to the Geotechnical clay model
$(s_u/\sigma'_v)_{NC}$	Ratio of the normally consolidated shear resistance to the consolidation normal stress. Typical range is between 0.25 and 0.5. Only used for Geotechnical clay model
α_{brk}	Pipe-soil adhesion factor. Only used for Geotechnical clay model
γ_{pre}	Consolidation preloading effect. Taken as the ratio of preloading (e.g. waterfilled submerged weight) to the submerged weight in the condition considered. Only used for Geotechnical clay model
* *OCR*	Overconsolidation ratio. Currently always set equal to γ_{pre}. Only used for Geotechnical clay model
m_{OCR}	Exponent accounting for long-term effect of overconsolidation. Only used for Geotechnical clay model
$\kappa_{passive}$	Passive earth pressure coefficient. A value of 2 is recommended to account for inclined loading (horizontal + lift). Only used for Geotechnical clay model
$s_{u,passive}$	Average undrained shear strength within the passive failure zone. Only used for Geotechnical clay model
κ_{active}	Active earth pressure coefficient. A value of 2 is recommended to account for inclined loading (horizontal + lift). Only used for Geotechnical clay model when suction is included
$s_{u,active}$	Average undrained shear strength within the active failure zone. Only used for Geotechnical clay model when suction is included
z_{pi}	User defined initial penetration
z_t	Trench depth
θ_t	Trench-wall angle in degrees (relative to vertical)

*These variables are currently hardcoded in PILSS

4.1.4.4 Hydrodynamic variables

An overview of hydrodynamical input data is given in Table 4-4.

Table 4-4 Overview input parameters – Hydrodynamic variables

* *Wave Spectrum*	Applicable Wave spectrum. Currently only JONSWAP spectrum is available
Wave Order	Wave order option: 1 - Linear, 2 - Second order
* *Smooth wake velocity*	Option to smooth wake velocity signal prior to internal differentiation. Currently not available in PILSS.
User defined lift reduction factor	Option to include a constant reduction factor on lift forces, for instance to account for a permeable seabed
$T_{simulation}$	Total simulation time for design sea state, taken as the sum of the ramp-up period T_{ramp} and storm duration T_{storm}. T_{storm} equal to 3 hours is recommended if no other information is available. With T_{ramp} = 2000s, this corresponds to $T_{simulation}$ = 12800s.
ρ_{water}	Water density
d	Water depth
h_{ref}	Reference height for generation of wave-induced water particle velocity. Should be taken as $D_{hydro}/2$ (or $D/2$ when user defined D_{hydro} is not included)
H_s	Significant wave height for design sea state
T_p	Peak period for design spectrum
γ	Peak-enhancement factor for design spectrum. If set equal to zero, PILSS will calculate a value according to the recommendation in DNV-RP-F109 /1/
θ_w	Angle between wave heading and pipeline. Generated regular waves are assumed to act perpendicular to the pipeline.
n	Spectral spreading exponent. Long-crested waves are generated if n >= 100.
k_s/D	Relative pipe/cable surface roughness. Three options available: Smooth (0.001), Rough (0.01) and Very rough (0.05).
U_m	Flow velocity amplitude. Only used for regular waves. Flow velocity for regular waves is calculated as: $U(t) = V + U_m \cdot \sin(2\pi t/T + \varphi_t \cdot \pi/180)$
T	Regular-wave period. Flow velocity for regular waves is calculated as: $U(t) = V + U_m \cdot \sin(2\pi t/T + \varphi_t \cdot \pi/180)$
V	Steady current velocity perpendicular to pipe/cable. Reductions due to boundary-layer effect and directionality must currently be included by user. Used for both regular and irregular wave time series.
φ_t	Phase displacement of regular wave. Flow velocity for regular waves is calculated as: $U(t) = V + U_m \cdot \sin(2\pi t/T + \varphi_t \cdot \pi/180)$
$F_{H,m}$	Horizontal force amplitude. Only used if a horizontal sine force is applied
$T_{F,H}$	Horizontal force period. Only used if a horizontal sine force is applied
$r_{perm,z}$	Reduction factor on hydrodynamic lift force, e.g. due to permeable seabed

*These parameters are currently hardcoded in PILSS

4.1.4.5 Random seed generation

An overview of input data for random seed generation is given in Table 4-5.

Table 4-5 Overview input parameters – Random seed generation

N_{seeds}	Number of random seeds (sea state realisations). The standard deviation of lateral displacement should stabilise, requiring at least 10 different realisations. The allowable total number of seeds is not limited, but a high number will affect PILSS performance negatively.
Re-generate Random Seeds	When pushed, the user will get the option to re-generate random seeds. The same option will be provided if N_{seeds} is changed. Existing random seeds will only be over-written if the user actively decides so.
Random seeds – Phase	Random seed used for the generation of random phase angles when generating a random seastate. A unique seed (integer between 0 and 100,000) should be specified for each seed run (Seed 1, Seed 2 etc.) A unique seed represents a unique sea state realisation for a given set of hydrodynamic input parameters.
Random seeds - Direction	Random seed used for the generation of random wave directions when generating a random sea state including wave spreading. A single direction seed is used for each analysis regardless of the number of sea state realisations/phase seeds.

4.1.4.6 Timeseries variables

An overview of input data for the generation of time series is given in Table 4-6.

Table 4-6 Overview input parameters – Timeseries variables

Include ramp-up period	Option for specifying a ramp-up period in the beginning of the design storm. If checked, a linear reduction to hydrodynamic loads for the ramp-up (T_{ramp}) period is introduced.
* t_{start}	Start time for dynamic analysis. Set to: 0.0
Δt	Timestep for dynamic analysis. Typical timestep size is 0.001s.
* N_t	Number of timesteps in storm simulation. Calculated by the program based on input values for timestep size, ramp-up duration and design storm duration
* Δt_{signal}	Timestep for wave-induced particle velocity signal. Recommended value is 0.35 s. Cannot be less than Δt.
* $N_{t,signal}$	Number of timesteps in wave-induced particle velocity signal. Calculated by the program based on input values for timestep size of velocity signal, ramp-up duration and design storm duration
T_{ramp}	Duration of ramp-up period. Hydrodynamic loads increase linearly from zero to full strength during ramp-up period. Recommended duration is approximately 20% of the storm duration, or 2000 s for a three-hour storm
$\Delta t_{results}$	Resolution of results vectors used for plots and writing results to file. It is required that $\Delta t_{results} \geq \Delta t$.
Δt_{plot}	Resolution of results vectors used for plots. It is required that $\Delta t_{plot} \geq \Delta t_{results}$.

*These variables are currently hardcoded in PILSS

4.1.4.7 Dynamic solution variables

An overview of input data for controlling the dynamic solution algorithm is given in Table 4-7.

Table 4-7 Overview input parameters – Dynamic solution variables

* β_{sol}	Newmark method parameter. Currently set to 0.25 (corresponds to average-acceleration method)
* γ_{sol}	Newmark method parameter. Currently set to 0.5
* $y_{conv,min}$	Minimum value of reference displacement used for convergence criterion. Always set to 0.001 times the outer pipe/cable diameter
* ε	Required accuracy in convergence criterion. Currently set to 0.00001 (10^{-5}).
* $N_{iter,max}$	Maximum number of equilibrium iterations in each timestep

* These variables are currently hardcoded in PILSS

4.2 Viewing Results Output

4.2.1 Results in Main sheet

4.2.1.1 Main results

Total design displacement is viewed in the “Main results” box, together with average displacement and accumulated standard deviation as function of maximum absolute displacement and number of seeds. The total design displacement is defined as the sum of the average displacement plus one standard deviation when the standard deviation has stabilized.

Displacements are reported both as absolute displacement in metres, and as function outer diameter (non-dimensional displacement).

4.2.1.2 Accumulated standard deviation plot

The standard deviation of maximum absolute displacement is plotted as function of number of seeds. This plot gives a good, quick qualitative evaluation of how the standard deviation stabilizes.

4.2.1.3 Displacement histogram

Resulting displacements for each seed run are shown on line with each Seed Number. The displacements are given both numerically and as a histogram. The displacement histogram illustrates graphically how much the displacements vary from seed to seed. The heights of the columns are relative to the height of the column with the largest displacement.

Total design displacement is illustrated with a blue line that intersects the histogram.

4.2.2 Results in Plots sheet

4.2.2.1 Numerical results

Maximum displacements per seed (sea state realisation), in addition to Total design displacement, are reported in the box “Detailed Results Overview” in Plots sheet.

The number of timesteps where the dynamic solution algorithm did not satisfy the convergence criterion after $N_{iter,max}$ = 10 iterations are listed for each seed. This number should generally be low. If the algorithm did not converge in more than 10 timesteps during one sea state realisation or in more than 3 consecutive timesteps, PILSS will give a warning message to the user and a smaller timestep is recommended.

4.2.2.2 Graphical results

Resulting time series for each seed are plotted in the Plots sheet. The time series that will be plotted are the ones that are checked as “Active”. If more plots than the active ones are desired, relevant checkboxes must be checked and PILSS re-run.

The time axes of the different plots may be adjusted for all the plots simultaneously with the “Zoom time axis” button. To do this, uncheck the “Auto limit” checkbox, specify desired upper and lower limits and click the button.

4.3 BatchRuns

Multiple design cases may be run in batch mode by using the BatchRuns functionality, e.g. when doing sensitivity studies.

Currently, only a few validation checks are performed for the input parameters specified in the BatchRuns sheet (contrary to the Main sheet). This puts more responsibility on the user to have control of which input parameters are relevant for the specific design case, in addition to the validity ranges for each parameter. BatchRuns does however contain conditional formatting and cell validation rules which help point the user in the right direction.

Input parameters may be copied manually from the Main sheet.

The “Duplicate runs” button will copy the input in the first column to the adjacent right columns a number of times as specified in “Number of cases”. Each column represents one design case.

To perform parameter studies, update the desired parameter(s) across the case columns and click “Run Batch”.

No cells in BatchRuns are protected from editing. If the formatting rules are broken due to e.g. copy pasting, the button “Repair formatting” may be clicked to restore formatting rules.

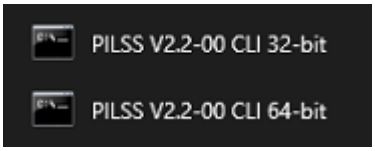
The results from the Batch run are reported on top of each case column.

Note that warning messages given for Main sheet calculations, e.g. if the number of steps failing to satisfy the convergence criterion is too high or if the lift force exceeds the submerged pipe weight during the analysis, are *not* given for BatchRuns calculations. The information is however printed to the detailed results files (if “Write results to file” is set to “Overview” or “Detailed”).

5 PILSS CLI

5.1 General

PILSS CLI is a Windows executable file (PILSS.exe) and can be run from Windows Command Processor ("command window"). The CLI can be run directly from a generic command window by specifying the absolute path to the PILSS installation folder, but is recommended to run PILSS CLI from one of the installed PILSS environments (32-bit or 64-bit), which can be found on the Windows Start Menu (see below screenshot):



PILSS CLI utilizes the same calculation routines as PILSS GUI but requires a valid .pls case file to run an analysis.

PILSS CLI accepts a few different arguments. A full overview of the CLI's functionality is given by the CLI itself, by typing `pilss -h` or, equivalently, `pilss --help` on the command line.

To verify that PILSS is correctly installed, type `pilss test`. PILSS will then run with an example case file, and should return the following result:

```
================================== DESIGN RESULTS ==================================

DESIGN DISPLACEMENT (1 avg. + 1 std.dev.) [m]:

       0.764579666378E+01
```

Results will always be written to screen. To write results to file, run PILSS CLI with the `--output` argument with the path to where to write the results files as the `value`, and optionally a user-defined prefix used to name the results files. The default prefix is "PILSS".

PILSS CLI can run PILSS with wave and/or load signals imported from file, see Section 5.2.2.

5.2 Input

5.2.1 PILSS Case File

Reading input data from the Case File (.pls file) is mandatory when running PILSS CLI. Most of the content in the Case File is explained in Sections 4.1.3 and 4.1.4. Some of the contents in the PILSS Case File is further explained in Table 5-1.

Table 5-1 PILSS Case File – Explanation of selected input parameters

PILSS version number	This is the version number of PILSS with which the current CaseFile was generated, and can be ignored by the user.
PILSS Case File version number	This is the version number of the Case File itself, which is used for compatibility checks by PILSS and can be ignored by the user.
GUI Checkbox settings	Some parameters are only read by PILSS GUI (mainly to initialize checkboxes) and can be ignored when running PILSS CLI, i.e. the following parameters: • PILSS Set-up (GUI only) • Include lay factor - checkbox (GUI only) • Initial penetration water-filled pipe - checkbox (GUI only) • User defined hydrodynamic diameter - checkbox (GUI only) • Consider remoulded soil - checkbox (GUI only) • Account for clay suction - checkbox (GUI only) • User-defined initial penetration - checkbox (GUI only) • Consider shielding from trench - checkbox (GUI only) • Smooth wake velocity - checkbox (GUI only) – Currently not activated • Include lift reduction factor - checkbox (GUI only) • Include ramp-up period - checkbox (GUI only)
Switches	Some parameters work as "switches" mimicking the function of a Boolean variable, where value "1" means the switch is turned on ("True") and value "2" means the switch is turned off ("False"). The switches are in general represented by checkboxes in PILSS GUI. The following switches are open for user manipulation in the Case File: • Include calculated lay factor - value (CalculatedLayFactor_int) • Include user-defined lay factor - value (UserDefinedLayFactor_int) • Initial penetration water-filled pipe - value (waterFilled_int) • User defined hydrodynamic diameter - value (HydrodynamicDiameter_int) • Consider remoulded soil - value (remoulded_int) • Account for clay suction - value (suction_int) • User-defined initial penetration - value (UserDefinedInitialPenetration_int) • Consider shielding from trench - value (IncludeTrench_int) • Smooth wake velocity - value (SmoothWakeVelocity) – Currently not activated • Include lift reduction factor - value (IncludeLiftReduction_int) • Include ramp-up period - value (RampUp_int)
Hardcoded values	Some parameters are listed as "hardcoded in PILSS", these data lines are not read by PILSS and can be ignored.
Spaceholder	Some parameter lines are labeled "Spaceholder". These were introduced during development to align formatting and can be ignored.
IterPILSS variables	Currently not in use, and can be ignored.
Wave time series type (GenerateRegular Wave)	Three options: 1 – Regular waves, 2 – Irregular waves, 3 – Read wave velocity from file, ref. Table 4-1 and Section 5.2.2.3.
Select flow type (flowtype)	Currently not used, but requires integer input. Set to: 1, ref. Table 4-1
Soil type (SoilModel)	Three options: 1 – Coulomb model, 2 – Sand model, 3 – Clay model, ref. Table 4-1
Soil model (InitialPenetrationModel)	Only relevant for Soil type 2 and 3. One option for Sand: 1 - Verley and Sotberg. Three options for clay: 1 - Verley and Lund, 2 - SAFEBUCK, 3 - Geotechnical model, ref. Table 4-1
Plot options (plotOption)	Not relevant for PILSS CLI, but is only relevant for Plots sheet in PILSS GUI. Two options: 1-Default (Δt_{plots} = 1 s), 2 – User defined (Δt_{plots} is user defined), ref. Table 4-1
Write results to file (sResultsToFile)	Three options: 1 – Detailed (detailed results and wave velocities and accelerations are written to file), 2 – None (no results are written to file), 3 – Overview (only main results are written to file) ("PILSS_ResultsOverview.txt") , ref. Table 4-1
Dtotal	Same as "*D*" in PILSS GUI, ref. Table 4-2.
mtotal	Same as "m_{pipe}" in PILSS GUI, ref. Table 4-2.
gamma_total	Same as "γ_s" in PILSS GUI, ref. Table 4-3.
delta_f	Same as "δ_f / B" in PILSS GUI, ref. Table 4-3.
Wave Spectrum	Same as "*Wave Spectrum*" in PILSS GUI. Only option "1" is available, ref. Table 4-4.

Wave Order	Same as "*Wave Order*" in PILSS GUI. Options "1" and "2" are available, ref. Table 4-4.
Tstorm	Same as "$T_{simulation}$" in PILSS GUI, ref. Table 4-4.
r_y	Hardcoded to 1.0 in PILSS.

5.2.2 Input signals from file

5.2.2.1 General

The force time series or wave time series input files consist of two parts: a header containing governing parameters, and a body containing the bulk time series.

Note that the format of the bulk time series must correspond to Fortran format specification E25.15, where each column is 25 characters wide and each number is on exponential form with 15 decimals. By default, 4 of the 25 characters are used to denote the exponent: the letter "E", the exponent sign, and a two-digit number for the exponent itself.

Given a coefficient with 15 decimals, this leaves three empty spaced characters and one space for the number sign, e.g. "`  0.418611954237113E-02`" or "`  -0.708710885893836E+02`". Working example files are provided in the *example* folder, see Sections 5.2.2.2 and 5.2.2.3.

For positive numbers, the sign is omitted, i.e. replaced by a blank space, as shown by the example.

5.2.2.2 Force time series

To activate reading of force time series signals from file, manually set the value of the "Hydrodynamic load model" to "0" in the PILSS Case File. Run PILSS CLI with the `--loads-from-file` argument with the name of the load file to be read as the `value`. The CLI will import the hydrodynamic load time series from file even if the setup specified in the .pls case file is not compatible with this option. In this scenario, a warning will be presented to the user.

The specified force time series will be applied directly to the pipeline in the dynamic analyses.

The required format of the force time series file is analogous to the wave time series file format, the only difference being that header. Governing parameters that need to be specified in the header are described in Table 5-2.

An example file, "*PILSS_Template_Forceseries_200s.txt*", is located in the *example* folder in the PILSS installation directory.

Table 5-2 PILSS Force time series file – Explanation of governing parameters

Number of series	Equivalent to *Number of seeds* in the Wave series file. This value overrides "Nseeds" given in the Case File and governs how many of the horizontal/vertical force time series pairs that is read from file.
Number of timesteps for force signal	The number of total records per variable in the time series, i.e. number of lines constituting the time series.
Constant time step for force signal [s]	This value takes precedence over "*Δt*" given in Table 4-6. Only constant time steps are currently supported.
Vertical load specified:	Can be "Yes" or "No": • "Yes": Vertical load time series must be specified • "No": Only horizontal load time series are specified, the vertical load time series will be set to zero in PILSS and must be removed from the time series file, i.e. the entire vertical load column (25 characters) must be removed.

5.2.2.3 Wave time series

To activate reading of wave time series signals from file, manually set the value of the "Wave time series type" to "3" in the PILSS Case File. Run PILSS CLI with the `--waves-from-file` argument with the name of the load file to be read

as the `value`. The CLI will import wave kinematics time series from file even if the .pls case file is not correctly set up but will throw warnings.

The specified wave time series will be used as input to the hydrodynamic load generation module in PILSS (no wave time series will be generated by PILSS).

The required format of the wave time series is identical to the format of the "*PILSS_WaveCurrentData.txt*" file, which may be used directly as input file to PILSS CLI or as a template for wave generated water particle velocity time series that have been generated or acquired outside of PILSS, and intended to be used as input to PILSS.

An example file, "*PILSS_Template_Waveseries_2000s.txt*", is located in the *example* folder in the PILSS installation directory. Note that PILSS CLI cannot read from and write to the same file at the same time, i.e. "*PILSS_WaveCurrentData.txt*" should not be used as file name for the input file unless a user-defined results file prefix is specified.

Governing parameters that need to be specified in the header are described in Table 5-3.

Table 5-3 PILSS Wave time series file – Explanation of governing parameters

Number of seeds	This value overrides "Nseeds" given in the CaseFile and governs how many of the velocity/acceleration time series pairs that is read from file.
Number of time steps for wave signal	The number of total records per variable in the time series, i.e. number of lines constituting the time series.
Constant time step for wave signal [s]	This value takes precedence over "Δt_{signal}" given in Table 4-6. Only constant time steps are currently supported.
Steady current velocity [m/s]	Steady current velocity component that will be superposed with the pure wave velocity time series to form the total velocity time series that enters the PILSS calculations.
Acceleration specified	Can be "Yes" or "No": • "Yes": Acceleration time series must be specified • "No": Only velocity time series are specified, the acceleration time series will be automatically calculated by PILSS and must be removed from the time series file, i.e. remove entire acceleration column (25 characters).

6 PILSS API

The PILSS dynamic stability solver library (PILSS_DLL.dll) can also be called directly through a simple API (Application Programming Interface). With this, it is possible to integrate the PILSS dynamic stability calculations into a 3rd party application, e.g., a script or a different GUI.

How to connect to the PILSS API is described in Appendix C. Examples are given for API clients written in VBA, C# and Python.

APPENDIX A REFERENCES

/1/ DNV-RP-F109, “On-bottom stability design of submarine pipelines, cables and umbilicals”, May 2021

/2/ DNV-RP-F114, “Pipe-soil interaction for submarine pipelines”, May 2021

/3/ DNV, “Second-order response – WABIRK user manual”, 2014-0054, Rev. 0

/4/ Sorenson, T., Bryndum, M.B., Jacobsen, V. “Hydrodynamic forces on pipelines – model tests”, DHI report no. L51522e, 1986

/5/ Bryndum, M.B., Jacobsen, V., Tsahalis, D.T., “Hydrodynamic forces on pipelines: Model tests”, J. Offshore Mechanics and Arctic Engineering, 114, 1992

/6/ Verley, R.L.P, Reed, K. “Use of laboratory force data in pipeline response simulations”, OMAE, 1989

/7/ Verley, R.L.P, Sotberg, T., “A Soil Resistance Model for Pipelines Placed on Sandy Soils”, J. Offshore Mechanics and Arctic Engineering, 116, pp. 145-153, 1994

/8/ Verley, R.L.P, Lund, K.M., “A Soil Resistance Model for Pipelines Placed on Clay Soils”, OMAE, 1995

/9/ White, D.J., Cheuk, C.Y., “SAFEBUCK Joint Industry Project Pipe-Soil Interaction Models for Lateral Buckling Design - Phase IIA Data Review”, Report no. GEO 09497v3, 2010

/10/ Randolph, M.F., White, D.J., “Pipeline Embedment in Deep Water: Processes and Quantitative Assessment”, OTC, 2008

/11/ DNV, “Design Guideline for Advanced Design for On-Bottom Stability of Submarine Lines: PILS JIP”, 2017-0497, Rev. 0.

/12/ DNV-RP-C205, “Environmental conditions and environmental loads”, September 2021

APPENDIX B EXERCISE: PIPELINE ON SANDBED

Case description

A 12 inch pipeline in the North Sea is found to be unstable according to the 10D Generalized Method in DNV-RP-F109. Evaluations have been performed which show that displacements up to 20 diameters are acceptable for the pipeline in question. PILSS is used to calculate the maximum expected lateral displacement for the pipeline over its design life.

Design Basis

Introduction and Route Description

A multiphase flowline of length approximately 10 km is to be installed between two neighbouring platforms in the North Sea. The flowline will connect Platform A with Platform B. The flowline route has a maximum depth of 112.5 m and a minimum depth of 108.0 m.

The route can be considered as one segment. The pipeline is directed approximately 75 degrees relative to geographic North.

Design lifetime is 20 years.

Pipeline data

The design data for the planned pipeline is given in Table B-1 below.

Table B-1 Pipeline design data

Outer steel diameter		323.9	mm
Steel wall thickness		15.9	mm
Steel density		7850	kg/m^3
Coating thickness		39.7	mm
Coating density		879	kg/m^3
Product density		300	kg/m^3
FJC Density		879	kg/m^3

Environmental data

The directional extreme currents are given in Table B-2.

Directional extreme values for the significant wave height and corresponding values for the spectral peak period are given in Table B-3.

The wave data are intended to be used with a JONSWAP spectral formulation. The duration of sea state extreme events is estimated to 3 hours. The peak enhancement factor can be taken as 3.3.

Density of seawater ranges from 1025 to 1028 kg/m^3.

Marine growth may be disregarded.

Table B-2 Weibull parameters and corresponding extreme values for the sector and omni-directional distributions of current speed 3 m above sea bottom at the Field. Duration of extreme event is 10 minutes.

Direction sector	Sector prob.	Weibull parameters			Annual probability of exceedance		
		Shape	Scale	Location	0.63	10^{-1}	10^{-2}
	%	-	cm/s	cm/s	m/s	m/s	m/s
0°	20.20	2.45	18.52	0.00	0.46	0.50	0.54
30°	20.28	2.54	18.91	0.00	0.45	0.50	0.53
60°	8.73	1.80	13.82	1.16	0.46	0.53	0.59
90°	4.84	1.26	7.63	3.53	0.43	0.52	0.60
120°	3.90	1.35	8.45	2.22	0.40	0.49	0.56
150°	4.88	2.12	12.03	0.19	0.32	0.36	0.40
180°	7.72	2.30	13.39	0.00	0.34	0.37	0.41
210°	7.00	2.40	12.98	0.00	0.31	0.35	0.38
240°	4.67	2.22	11.87	0.00	0.30	0.34	0.37
270°	4.09	2.20	11.79	0.00	0.30	0.34	0.37
300°	4.88	1.97	12.62	0.00	0.36	0.41	0.46
330°	8.80	2.25	15.32	0.00	0.40	0.44	0.48
0°-360°	**100.00**	**2.02**	**15.49**	**0.00**	**0.50**	**0.55**	**0.60**

Table B-3 Directional extreme significant wave height (Hs) and spectral peak period (Tp) at the Field.

Direction sector	Annual probability (q) of exceedance					
	q = 0.63		q = 10^{-1}		q = 10^{-2}	
	Hs (m)	Tp (s)	Hs (m)	Tp (s)	Hs (m)	Tp (s)
0°	9.5	13.3	12.0	14.6	14.3	15.8
30°	5.7	11.2	7.5	12.2	9.1	13.1
60°	4.2	10.4	5.6	11.2	6.8	11.8
90°	7.7	12.3	9.7	13.4	11.5	14.4
120°	10.1	13.6	12.2	14.7	14.1	15.7
150°	8.5	12.7	10.7	13.9	12.7	15.0
180°	8.0	12.5	9.9	13.5	11.5	14.4
210°	7.6	12.3	9.1	13.1	10.4	13.8
240°	8.1	12.5	10.3	13.7	12.2	14.7
270°	8.1	12.5	10.3	13.7	12.3	14.8
300°	7.6	12.3	9.9	13.5	12.0	14.6
330°	8.2	12.6	10.3	13.7	12.2	14.7
0°-360°	**10.5**	**13.8**	**12.5**	**14.9**	**14.3**	**15.8**

Soil Description

The seabed mainly displays loose to medium dense sand. Patches and areas of medium to dense coarser sand with gravel and possible thin embedded layers of clay are expected in some parts along the pipeline routes. The seabed roughness parameter z_0 can be set to $7.5 \cdot 10^{-5}$ m. Table B-4 lists some soil parameters for the pipeline. Soil permeability is unknown, assume no permeability.

Table B-4 Soil parameters

Soil Description	Lateral Resistance Coefficient [-]			Saturated Unit Weight of Soil [kg/m^3]		
	LB	BE	UB	LB	BE	UB
Sand (Unit I)	-	-	-	2039	2069	2100
Pre-lay Rock	0.47	0.56	0.64	-	-	-

Dynamic lateral on-bottom stability analysis in PILSS

Preliminary calculations

In this example, StableLines was used to do preliminary on-bottom stability calculations according to DNV-RP-F109. It is found that the most critical wave and current direction is from the north (zero degrees) with the combination of 100 and 10 year Return Period Value for waves and current, respectively. This preliminary calculation may also be done in PILSS, with some more effort.

PILSS Input

The following parameters are populated in PILSS Main sheet, see also Figure B-1:

Parameter	Input value	Comment
Input specification		
Case name	ExampleCase_Sand_20D	Suitable project name.
Hydrodynamic load model	1) Database force model (recommended)	Recommended value
Wave time series type	Irregular wave (recommended)	Recommended for design
Flow type	N/A	
Soil type	2) Sand model	According to design basis
Soil model	Verley and Sotberg	Only option
Plot options	Default	1 second resolution is sufficient for a visual presentation of the time series
Write results to file	None	Writing detailed results to file takes more time and is not needed for a typical design cases

Parameter	Input value	Comment
Cross-sectional data		
D	0.4033	Steel + coating
w_s	484.76	Calculated independently, e.g. in StableLines, PET or user-made spreadsheet
m_{pipe}	180.74	Mass of pipe in design condition, which in this case is in operation. calculated as $w_{s,dry}/g$. In this case, $w_{s,dry} = w_s + buoyancy$
t_{mgrt}	0	Per Design Basis
Soil variables		
k_{soil}	65000	Recommended value
α_{soil}	0.02	Suitable value
γ_s'	10212	Used Best estimate, subtracted unit weight of sea water to get submerged unit weight of soil.

Parameter	Input value	Comment
Hydrodynamic variables		
Wave Spectrum	JONSWAP	Default
Wave Order	1) Linear	Sufficiently accurate for the current design case.
$T_{simulation}$	10800 and 12800	Storm duration is 10800 s per Design Basis. A 2000 s ramp-up may be included if it is found to be beneficial for the design displacement.
ρ_{water}	1028	Per Design Basis, conservative value.
d	108	Per Design Basis, conservative value.
h_{ref}	0.20165	Taken as center of pipeline
H_s	14.3	Worst case from Preliminary calculation (100 year RPV)
T_p	15.8	Worst case from Preliminary calculation (100 year RPV)
γ	3.3	Per Design Basis
θ_w	90	North directional data comprise the interval (345,15) degrees relative to N. Pipe heading is 75 degress relative to N, implying a 90 degree relative angle for the worst direction within the sector
n	8	If no other information is available DNV-RP-F109 recommends selecting the most conservative value in the range 2 to 8
k_s/D	Rough	Conservatively assumed to be rough as roughness is assumed to worsen over the course of the design life
V	0.358	Reduced to 0.358 m/s according to DNV-RP-F109 (10 year RPV).
Random seed generation		
N_{seeds}	18	Plot indicates stabilized standard deviation.
Random seeds – Phase	99985, 3979, 12320, 56825, 48847, 8098, 60130, 44962, 4137, 54427, 96780, 1915, 59138, 84810, 8788, 93254, 35332, 98198	Randomly generated.

Parameter	Input value	Comment
Random seed - Direction	26011	
Timeseries variables		
Include ramp-up period	No and Yes	Potential benefit to be assessed.
Δt	0.01 initially	Sensitivity performed in BatchRuns. Converging on Δt = 0.001s.
T_{ramp}	0 or 2000	It is assessed whether dynamic penetration build-up has a beneficial effect on design by including a 2000 s (recommended value) ramp-up period.
$\Delta t_{results}$	0.1	Relevant if writing results to file.

Results

Total design displacement is calculated to be 6.3 metres, which corresponds to 15.6 diameters displacement, which is well within the requirement of maximum 20 diameters lateral displacement.

The potential beneficial effect from dynamic build-up of penetration was assessed by including a 2000 s ramp-up period (giving $T_{simulation}$ = 12800 s). The total design displacement was then 6.5 metres, which is slightly higher than the 6.3 metres obtained without ramp-up ($T_{simulation}$ = T_{storm} = 10800 s). Hence, there was no benefit of including a ramp-up period for this case.

A small sensitivity study in Δt was done using the BatchRuns functionality. Input parameters were copied from the Main sheet into the first calculation column in BatchRuns. Three cases were generated with three different values of Δt, i.e. 0.01 s, 0.001s and 0.0005 s, which resulted in total design diplacements of 6.77 m, 6.30 m and 6.29 m, respectively.

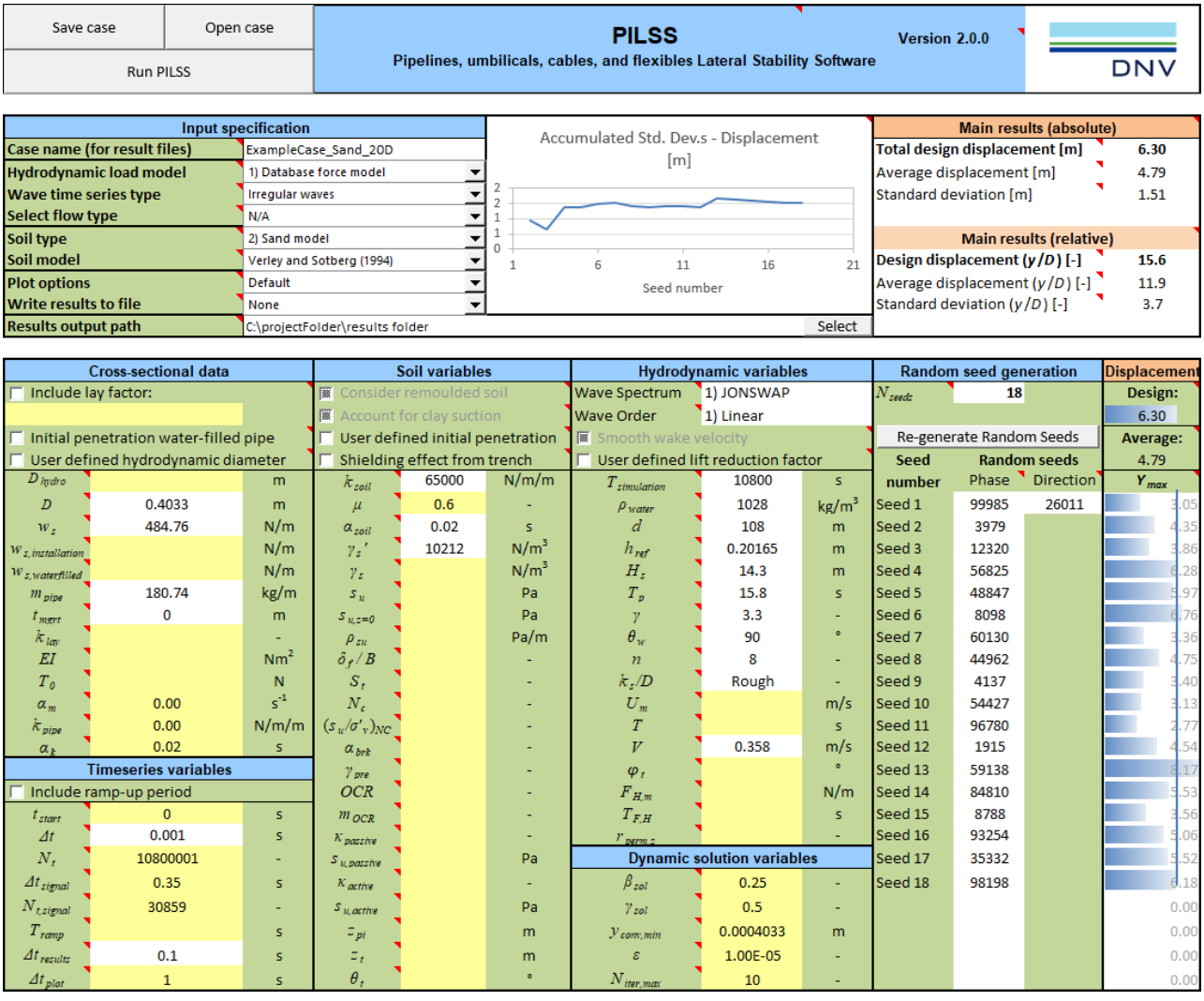


Figure B-1 Example Case Sand - PILSS Main sheet screen shot

APPENDIX C DLL CALL - DETAILS

To directly call the PILSS DLL, use the syntax described in this section.

PILSS_DLL.dll is written in the Fortran language and uses the *STDCALL* calling convention, which must also be used by the calling program.

The main difference between *STDCALL* and *CDECL* (C and C++ default calling convention) is that with *STDCALL* the callee cleans the stack, while with *CDECL*, the caller cleans the stack. See for example https://docs.microsoft.com/en-us/cpp/cpp/stdcall?view=msvc-170 for details.

PILSS_DLL.dll entry point

The PILSS_DLL.dll entry point looks like the following:

```
subroutine PILSSInterface_v2(inCross_real  ,  &
    inHydro_real, inHydro_int, seedArray_phase, seedArray_crest, &
    inSoil_real, inSoil_int, &
    inTime_real, inTime_int, NrOfTimesteps, NrOfTimesteps_signal, NrOfTimesteps_res, &
    inDyn_real, inDyn_int, &
    t_vector_U, U_t_matrix, A_t_matrix, F_H_matrix, F_L_matrix, &
    NrOfPlotsteps, addedPlotRows, delta_t_plot, stepLength, printToFile,
readWaveVelFromFile, generatePlots, ExplicitAcceleration, dummyBoolean1, dummyBoolean2, &
    result_matrix, maxMatrix, minMatrix, maxAbsMatrix, &
    t_matrix_plot, y_matrix_plot, z_matrix_plot, &
    v_matrix_plot, a_matrix_plot, &
    Fsoil_matrix_plot, Fstiff_matrix_plot, FHrel_matrix_plot, FLrel_matrix_plot,
yref_matrix_plot, iElPl_matrix_plot, &
    outVar_int_matrix, resultsPath, caseName)

!DEC$ ATTRIBUTES DLLEXPORT, STDCALL, REFERENCE, ALIAS:"PILSSInterface_v2" ::
PILSSInterface_v2
```

PILSS_DLL.dll legacy entry point

The legacy entry point “PILSSInterface()” is kept for backwards compatibility. It has the same subroutine signature as in PILSS v1.2, and looks like the following:

```
subroutine PILSSInterface(inCross_real  ,  &
    inHydro_real, inHydro_int, seedArray_phase, seedArray_crest, &
    inSoil_real, inSoil_int, &
    inTime_real, inTime_int, NrOfTimesteps, NrOfTimesteps_signal, NrOfTimesteps_res, &
    inDyn_real, inDyn_int, &
    t_vector_U, U_t_matrix, A_t_matrix, F_H_matrix, F_L_matrix, &
    NrOfPlotsteps, addedPlotRows, delta_t_plot, dummyInteger, printToFile,
readWaveVelFromFile, generatePlots, ExplicitAcceleration, dummyBoolean, dummyBoolean2,
result_matrix, maxMatrix, minMatrix, maxAbsMatrix, t_matrix_plot, y_matrix_plot,
z_matrix_plot, v_matrix_plot, a_matrix_plot, Fsoil_matrix_plot, Fstiff_matrix_plot,
FHrel_matrix_plot, FLrel_matrix_plot, yref_matrix_plot, iElPl_matrix_plot,
outvar_int_matrix)

!DEC$ ATTRIBUTES DLLEXPORT, STDCALL, REFERENCE, ALIAS:"PILSSInterface" :: PILSSInterface
```

Fortran Type definitions

The following fortran type definitions are used in PILSS:

- Floating-point numbers (real numbers): `real(selected_real_kind(15,307))` (double precision)
- Integers: `integer(kind = 4)`
- Character strings: `character(len = 4096)`

Output parameters need to be allocated in memory prior to the DLL call, but their values should be set to zero.

Note that Fortran arrays are 1-based, all parameters are sent as references (even the most basic types, e.g., integers), and a "string" in Fortran is a character array ("string of characters"). As an example, when calling a Fortran routine that takes in an integer array, the caller would not send in the array with actual values, but rather a reference to where in memory this array is located.

PILSS_DLL.dll input/output parameters

Input and output parameters required to successfully call PILSS_DLL.dll are listed in Table C-1, Table C-2 and Table C-3. "I/O" referes to the `intent` attribute in Fortran.

Table C-1 PILSS_DLL.dll input/output parameters

Parameter name	Type	Dimension(s)	I/O	Description
`inCross_real(:)`	`real`	12	I	Contains cross sectional data, see Table C-2
`inHydro_real(:)`	`real`	17	I	Contains hydrodynamic data, see Table C-2
`inHydro_int(:)`	`integer`	10	I	Contains hydrodynamic data, see Table C-2
`seedArray_phase(:)`	`integer`	N_{seeds}	I	Contains random seeds for random wave phase generation – integers between 0 and 100,000
`seedArray_crest(:)`	`integer`	N_{seeds}	I	Contains random seeds for generation of random wave directions for short crested waves (wave spreading) – integers between 0 and 100,000. Only the first entry is used, and all other analyses in the same run will be based on the same direction set.
`inSoil_real(:)`	`real`	22	I	Contains soil data, see Table C-2
`inSoil_int(:)`	`integer`	9	I	Contains soil data, see Table C-2
`inTime_real(:)`	`real`	4	I	Contains time control data, see Table C-2
`inTime_int(:)`	`integer`	1	I	Contains time control data, see Table C-2
`NrOfTimesteps`	`integer`	1	I	Number of timesteps in analysis = ceiling(Tstorm / delta_t) + 1
`NrOfTimesteps_signal`	`integer`	1	I	Number of timesteps for generation of hydrodynamic forces = ceiling(Tstorm / delta_t_signal) + 1 NB: NrOfTimesteps_signal <= NrOfTimesteps

Parameter name	Type	Dimension(s)	I/O	Description
NrOfTimesteps_res	integer	1	I	Number of timesteps for results vectors. = NrOfTimesteps if delta_t_res = delta_t NB: NrOfTimesteps_res <= NrOfTimesteps
inDyn_real(:)	real	1	I	Contains dynamic analysis data, see Table C-2
inDyn_int(:)	integer	2	I	Contains dynamic analysis data, see Table C-2
t_vector_U(:)	real	$N_{t,signal}$	I, O	Time vector for hydrodynamic load signal. Generated based on delta_t_signal, or read from file.
U_t_matrix(:,:)	real	$N_{t,signal}$, N_{seeds}	I, O	Free stream flow velocity vector, generated by PILSS or read from file.
A_t_matrix(:,:)	real	$N_{t,signal}$, N_{seeds}	I, O	Free stream flow acceleration vector, generated by PILSS or read from file.
F_H_matrix(:,:)	real	$N_{t,signal}$, N_{seeds}	I, O	Total horizontal load input signal, generated by PILSS or read from file.
F_L_matrix(:,:)	real	$N_{t,signal}$, N_{seeds}	I, O	Total vertical load input signal, generated by PILSS or read from file.
NrOfPlotsteps	integer	1	I	Number of timesteps for plot vectors. = NrOfTimesteps_res if delta_t_plot = delta_t_res NB: NrOfPlotsteps <= NrOfTimesteps_res
addedPlotRows	integer	1	I	= 0 if Tstorm/delta_t_plot is an integer, = 1 otherwise.
delta_t_plot	real	1	I	Resolution of plot results in seconds.
stepLength	integer	1	I	Specifies length of timestep for the written output in terms of delta_t, i.e. if stepLength = 1000 the output variables are written to file at time intervals of (1000 * delta_t)
printToFile	integer	1	I	Option for writing results to file: 1 - Write all results to .txt file, 2 - Do not write results to file, 3 - Only write ResultsOverview to file
readWaveVelFromFile	integer	1	I	Option for reading Wave velocities from file: 1 - read from file, 2 - do not read from file
generatePlots	logical	1	I	Set to TRUE to populate the plot matrices.
ExplicitAcceleration	logical	1	I	Default value: TRUE. Set to FALSE if reading wave velocities, but not accelerations from file.
dummyBoolean1	logical	1	I	Used for testing. Set to: FALSE
dummyBoolean2	logical	1	I	Used for testing. Set to: TRUE
result_matrix(:,:)	real	2 , 6	I, O	Matrix storing condensed results, see **Table C-3**
maxMatrix(:,:)	real	10 , N_{seeds}	I, O	Matrix storing maximum (positive) values for the different result vectors (rows) per seed (coloumns), see **Table C-3**
minMatrix(:,:)	real	10 , N_{seeds}	I, O	Matrix storing minimum (negative) values for the different result vectors (rows) per seed (coloumns), see **Table C-3**
maxAbsMatrix(:,:)	real	10 , N_{seeds}	I, O	Matrix storing maximum of absolute values for the different result vectors (rows) per seed (coloumns), see **Table C-3**
t_matrix_plot(:,:)	real	$N_{t,plots}$, 1	I, O	Matrix storing time array for plot (one coloumn (applies to all seeds)) – resolution governed by delta_t_plot
y_matrix_plot(:,:)	real	$N_{t,plots}$, N_{seeds}	I, O	Matrix storing displacement arrays (one coloumn per seed) – resolution governed by delta_t_plot
z_matrix_plot(:,:)	real	$N_{t,plots}$, N_{seeds}	I, O	Matrix storing penetration arrays (one coloumn per seed) – resolution governed by delta_t_plot
v_matrix_plot(:,:)	real	$N_{t,plots}$, N_{seeds}	I, O	Matrix storing cylinder velocity arrays (one coloumn per seed) – resolution governed by delta_t_plot
a_matrix_plot(:,:)	real	$N_{t,plots}$, N_{seeds}	I, O	Matrix storing cylinder acceleration arrays (one coloumn per seed) – resolution governed by delta_t_plot
Fsoil_matrix_plot(:,:)	real	$N_{t,plots}$, N_{seeds}	I, O	Matrix storing total soil force arrays (one coloumn per seed) – resolution governed by delta_t_plot

Parameter name	Type	Dimension(s)	I/O	Description
`Fstiff_matrix_plot(:,:)`	real	$N_{t,plots}$, N_{seeds}	I, O	Matrix storing soil force without damping contribution arrays (one coloumn per seed) – resolution governed by delta_t_plot
`FHrel_matrix_plot(:,:)`	real	$N_{t,plots}$, N_{seeds}	I, O	Matrix storing total horizontal force (sum of drag and inertia) including correction for relative velocity arrays (one coloumn per seed) – resolution governed by delta_t_plot
`FLrel_matrix_plot(:,:)`	real	$N_{t,plots}$, N_{seeds}	I, O	Matrix storing lift force including correction for relative velocity arrays (one coloumn per seed) – resolution governed by delta_t_plot
`yref_matrix_plot(:,:)`	real	$N_{t,plots}$, N_{seeds}	I, O	Matrix storing origo for force-displacement curve arrays (one coloumn per seed) – resolution governed by delta_t_plot
`iElPl_matrix_plot(:,:)`	integer	$N_{t,plots}$, N_{seeds}	I, O	Matrix storing soil code arrays (one coloumn per seed) – resolution governed by delta_t_plot
`outvar_int_matrix(:,:)`	Integer	5 , N_{seeds}	I, O	Matrix storing information on number of non-converged steps and flags for more than 3 consecutive non-convergent steps and for buoyant pipe encountered in the analysis, see **Table C-3**
`resultsPath(4096)`	`Character, optional`	1	I	Optional assumed length character array specifying path to results folder. Used if writing results to file.
`caseName(4096)`	`Character, optional`	1	I	Optional assumed length character array specifying user-defined prefix to results files. Used if writing results to file.

Note that mixed language interoperability with Fortran typically is challenging when it comes to passing strings. Since v2.2, PILSS provides two exposed interfaces (entry points): `PILSSInterface()`, which does not take in any character strings and is backwards compatible with PILSS v1.2 API, and `PILSSInterface_v2()` , which *does* take in *optional* character strings with pre-defined length of 4096 characters.

A string type is often differently represented in Fortran than in other languages. To pass `resultsPath` and `caseName` from C#, the strings should be padded with white spaces to reach string length of 4096 characters, and then be converted to a character array. Fortran will interpret the array of length 4096 of characters (or strings of length 1) as (non-array) character strings of length 4096. Example in C#:

```
var resultsPath = "ResultsFolder".PadRight(4096).ToCharArray();
var caseName = "CaseName".PadRight(4096).ToCharArray();
```

In the C# example below, the `[Optional]` attribute (https://learn.microsoft.com/en-us/dotnet/api/system.runtime.interopservices.optionalattribute) is used to define `resultsPath` and `caseName`. If these two strings are not declared and sent in with the call to the `PILSSInterface_v2()` subroutine, no results will be written to file regardless of the value of `printToFile.`

If the `[Optional]` attribute is omitted in the subroutine signature definition, the two strings must be declared and sent in with calls to the `PILSSInterface_v2()` subroutine. Whether results will be written to file is fully governed by the value of `printToFile`.

Table C-2 PILSS_DLL.dll input parameter containers

Container name	Index	Input parameter	Description
inCross_real(:)	1	D_hydro	Hydrodynamic pipe diameter
	2	D	Outer pipe diameter including any coating layer
	3	w_s	Submerged pipe weight per unit length
	4	w_s_installation	Submerged pipe weight per unit length during installation
	5	w_s_waterfilled	Submerged pipe weight per unit length in waterfilled condition
	6	m	Dry pipe mass per unit length
	7	k	Dummy parameter. Set to: 0.0
	8	t_mgrt	Thickness of marine growth
	9	k_lay_user	User defined lay factor
	10	EI	Bending stiffness of pipe
	11	T0	Horizontal component of effective lay tension
	12	alpha_m	Damping coefficient for mass-proportional Rayleigh damping
inHydro_real(:)	1	rho_w	Water density
	2	U_m	Flow velocity amplitude. Only used for regular waves.
	3	T	Regular-wave period
	4	U_c	Steady current velocity perpendicular to pipe/cable. Reductions due to boundary-layer effect and directionality must currently be included by user.
	5	Phi_t	Phase displacement of regular wave.
	6	FH_m	Horizontal force amplitude. Only used if a horizontal sine force is applied
	7	T_FH	Horizontal force period. Only used if a horizontal sine force is applied
	8	depth	Water depth
	9	Tsimulation	Simulation length
	10	WaveVelRefHeight	Reference height for generation of wave-induced water particle velocity.
	11	Hs	Significant wave height for design sea state
	12	Tp	Peak period for design spectrum
	13	SpectrumPeakedness	Peak-enhancement factor for design spectrum
	14	theta_w	Relative angle between pipe and wave heading
	15	n_spreading	Spectral spreading exponent
	16	r_H	Dummy parameter. Set to: 1.0
	17	r_L	Reduction factor on hydrodynamic lift load
inHydro_int(:)	1	HydroModel	Force model option: 4 - Database force model, 5 - Horizontal sine force
	2	flowtype	Not in use. Set to: 0
	3	roughness	Pipe surface roguhness: 1 - Smooth, 2 - Rough, 3 - Very rough (roughness = 3 only applicable for HydroModel = 4)
	4	GenerateRegularWave	A regular wave signal should be generated: 1 - True, 2 – False
	5	SmoothWakeVelocity	Wake velocity should be smoothed prior to differentiation: 1 - True, 2 – False. Default value: 2
	6	WaveOrder	Wave order option: 1 - Linear, 2 - Second order
	7	SpectrumSpecification	Applicable Wave spectrum: 1 - JONSWAP
	8	numseeds	Number of random seastates to be analysed
	9	HydrodynamicDiameter_int	Option for specifying user defined hydrodynamic diameter: 1 – True, 2 – False
	10	IncludeLiftReduction_int	Option for including reduction factor on hydrodynamic lift: 1 – True, 2 – False

Container name	Index	Input parameter	Description
inSoil_real(:)	1	k_soil	Elastic stiffness of the soil. Recommended value is 65000 N/m/m
	2	mu	Friction coefficient. Specified by user for Coulomb friction. Set to 0.6 for sand and 0.2 for clay
	3	gamma_mark	Submerged (effective) unit weight of soil
	4	rho_su	Depth gradient of undrained shear strength. Only used for SAFEBUCK and Geotechnical clay models
	5	Nc	Bearing capacity factor for clay. Only used for initial penetration according to the Geotechnical clay model
	6	su_z0	Undrained shear strength of soil at seabed (z = 0). Only used for SAFEBUCK and Geotechnical clay models
	7	delta_mobilization	Mobilization distance required to reach failure at the relevant penetration depth. Expressed as fraction of the pipe-soil contact width B, which is calculated by the program. Only used for initial penetration calculations according to the SAFEBUCK and Geotechnical clay models
	8	St	Sensitivity (ratio of intact to remoulded shear strength). Only used for SAFEBUCK and Geotechnical clay models when remoulded soil is considered
	9	alpha_soil	Damping coefficient for soil damping in elastic regime (c_soil = alpha_soil * k_soil)
	10	su_sigmav_NC	Ratio of the normally consolidated shear resistance to the consolidation normal stress. Typical range is between 0.25 and 0.5. Only used for Geotechnical clay model
	11	alpha_brk	Pipe-soil adhesion factor. Only used for Geotechnical clay model
	12	gamma_pre	Consolidation preloading effect. Taken as the ratio of preloading (e.g. waterfilled submerged weight) to the submerged weight in the condition considered. Only used for Geotechnical clay model
	13	m_OCR	Exponent accounting for long-term effect of overconsolidation. Only used for Geotechnical clay model
	14	kappa_active	Active earth pressure coefficient. A value of 2 is recommended to account for inclined loading (horizontal + lift). Only used for Geotechnical clay model when suction is included
	15	kappa_passive	Passive earth pressure coefficient. A value of 2 is recommended to account for inclined loading (horizontal + lift). Only used for Geotechnical clay model
	16	su_active	Average undrained shear strength within the active failure zone. Only used for Geotechnical clay model when suction is included
	17	su_passive	Average undrained shear strength within the passive failure zone. Only used for Geotechnical clay model
	18	gamma_total	Total unit soil weight in air. Only used for clay (in Verley and Lund model and after break-out in the other models). Verley and Lund recommend a value of 18000 N/m^3 regardless of the value of submerged unit weight.
	19	su	Undrained shear strength of soil. Only used for Verley and Lund clay model
	20	z_ini_user	User defined initial penetration
	21	z_t	Trench depth
	22	theta_t	Trench-wall angle

Container name	Index	Input parameter	Description
inSoil_int(:)	1	SoilModel	Soil model option: 1 - Coulomb friction, 2 - Pipe-sand, 3 - Pipe-clay
	2	CalculatedLayFactor_int	Option for including a lay factor calculated by PILSS to increase static contact pressure for initial penetration calculation. 1 - True, 2 – False
	3	UserDefinedLayFactor_int	Option for including a user defined lay factor to increase static contact pressure for initial penetration calculation. 1 – True, 2 – False
	4	InitialPenetrationModel	Penetration calculation model option for pipe-clay: 1 - Verley and Lund (1995), 2 - SAFEBUCK, 3 - Geotechnical model
	5	remoulded_int	Option for reducing shear strength due to remoulded soil (used for the combination SoilModel = 3 and InitialPenetrationModel = 2 or 3). 1 - True, 2 – False
	6	suction_int	Option to account for clay suction: 1 - True, 2 – False
	7	waterFilled_int	Option to account for weight of water-filled pipe when calculating initial penetration: 1 – True, 2 - False
	8	UserDefinedInitialPenetration_int	Option to include user defined initial penetration: 1 – True, 2 - False
	9	IncludeTrench_int	Option to include reduction factors on hydrodynamic loads due to trench. 1 – True, 2 - False
inTime_real(:)	1	t_start	Start time for dynamic analysis. Set to: 0.0
	2	delta_t	Timestep for dynamic analysis. Typical timestep size is 0.001s.
	3	delta_t_signal	Timestep for wave-induced particle velocity signal. Recommended value is 0.35 s. Cannot be less than Δt.
	4	delta_t_res	Resolution of results vectors used for plots and writing results to file.
inTime_int(:)	1	-	Not in use. Set to: 0
inDyn_real(:)	1	RampUpPeriod	Duration of ramp-up period. Hydrodynamic loads increase linearly from zero to full strength during ramp-up period. Recommended duration is approximately 20% of the storm duration, or 2000 s for a three-hour storm
inDyn_int(:)	1	MaxNrOfIterations	Maximum number of equilibrium iterations in each timestep
	2	RampUp_int	Set to 1 if storm ramp up introduces a linear reduction to hydrodynamic loads for the ramp-up period: 1 – True, 2 – False

Table C-3 PILSS_DLL.dll output parameter containers

Container name	Index	Output parameter	Description
result_matrix(:,:)	1,1	Displacement [m]	Average maximum
	1,2		Average minimum
	1,3		Average maximum absolute
	1,4		Standard deviation of maxima
	1,5		Standard deviation of minima
	1,6		Standard deviation of maximum absolutes
	2,1	Penetration [m]	Average maximum
	2,2		Average minimum
	2,3		Average maximum absolute
	2,4		Standard deviation of maxima
	2,5		Standard deviation of minima
	2,6		Standard deviation of maximum absolutes
maxMatrix(:,:)	1,*	Displacement [m]	Maximum per seed (for all timesteps)
	2,*	Penetration [m]	Maximum per seed (for all timesteps)
	3,*	Pipe velocity [m/s]	Maximum per seed (for output vector only)
	4,*	Pipe acceleration [m/s^2]	Maximum per seed (for output vector only)
	5,*	Soil force (total) [N/m]	Maximum per seed (for output vector only)
	6,*	Soil force (no damping) [N/m]	Maximum per seed (for output vector only)
	7,*	Horizontal load [N/m]	Maximum per seed (for output vector only)
	8,*	Vertical load [N/m]	Maximum per seed (for output vector only)
	9,*	Total plastic displacement [m]	Maximum per seed (for output vector only)
	10,*	Soil code [-]	Maximum per seed (for output vector only)
minMatrix(:,:)	1,*	Displacement [m]	Minimum per seed (for all timesteps)
	2,*	Penetration [m]	Minimum per seed (for all timesteps)
	3,*	Pipe velocity [m/s]	Minimum per seed (for output vector only)
	4,*	Pipe acceleration [m/s^2]	Minimum per seed (for output vector only)
	5,*	Soil force (total) [N/m]	Minimum per seed (for output vector only)
	6,*	Soil force (no damping) [N/m]	Minimum per seed (for output vector only)
	7,*	Horizontal load [N/m]	Minimum per seed (for output vector only)
	8,*	Vertical load [N/m]	Minimum per seed (for output vector only)
	9,*	Total plastic displacement [m]	Minimum per seed (for output vector only)
	10,*	Soil code [-]	Minimum per seed (for output vector only)
maxAbsMatrix(:,:)	1,*	Displacement [m]	Maximum absolute per seed (for all timesteps)
	2,*	Penetration [m]	Maximum absolute per seed (for all timesteps)
	3,*	Pipe velocity [m/s]	Maximum absolute per seed (for output vector only)
	4,*	Pipe acceleration [m/s^2]	Maximum absolute per seed (for output vector only)
	5,*	Soil force (total) [N/m]	Maximum absolute per seed (for output vector only)
	6,*	Soil force (no damping) [N/m]	Maximum absolute per seed (for output vector only)
	7,*	Horizontal load [N/m]	Maximum absolute per seed (for output vector only)
	8,*	Vertical load [N/m]	Maximum absolute per seed (for output vector only)
	9,*	Total plastic displacement [m]	Maximum absolute per seed (for output vector only)
	10,*	Soil code [-]	Maximum absolute per seed (for output vector only)
outVar_int_matrix(:,:)	1,*	Number of timesteps where displacements are reduced (i.e. where the change in displacement exceeds 100D in a single timestep) (-)	
	2,*	Number of timesteps where convergence criterion is not satisfied (-)	
	3,*	First timestep that did not satisfy convergence criterion (-)	
	4,*	Flag indicating whether more than 3 consecutive timesteps failed to satisfy convergence criterion. 1 – True, 2 - False	
	5,*	Flag indicating whether lift force exceeded submerged weight during analysis (i.e. whether pipe became buoyant). 1 – True, 2 - False	

Example of calling PILSS_DLL.dll API from C#:

There are various ways of calling external, unmanaged code from C#. Below is an example using the `DllImport` attribute (https://learn.microsoft.com/en-us/dotnet/api/system.runtime.interopservices.dllimportattribute) in combination with the `extern` modifier.

Alternatively, one could use the `UnmanagedFunctionPointer` attribute (https://learn.microsoft.com/en-us/dotnet/api/system.runtime.interopservices.unmanagedfunctionpointerattribute) in combination with the `delegate` type.

In the *[PILSS installation folder]\example\APIclient_Csharp* folder, example code for calling the PILSS_DLL.dll from C# is included. This uses the former alternative of importing Fortran DLL, and is illustrated below.

Defining the PILSSInterface_v2 Fortran subroutine signature in C#:

```
[DllImport(pilssDllPathFullOrRelative, EntryPoint = "PILSSInterface_v2",
CallingConvention = CallingConvention.StdCall)]
public static extern void PILSSInterface_v2(
    [In] double[] inCross_real,
    [In] double[] inHydro_real,
    [In] int[] inHydro_int,
    [In] int[] seedArray_phase,
    [In] int[] seedArray_crest,
    [In] double[] inSoil_real,
    [In] int[] inSoil_int,
    [In] double[] inTime_real,
    [In] int[] inTime_int,
    [In] ref int NrOfTimesteps,
    [In] ref int NrOfTimesteps_signal,
    [In] ref int NrOfTimesteps_res,
    [In] double[] inDyn_real,
    [In] int[] inDyn_int,
    [In][Out] double[] t_vector_U,
    [In][Out] double[,] U_t_matrix,
    [In][Out] double[,] A_t_matrix,
    [In][Out] double[,] F_H_matrix,
    [In][Out] double[,] F_L_matrix,
    [In] ref int NrOfPlotsteps,
    [In] ref int addedPlotRows,
    [In] ref double delta_t_plot,
    [In] ref int dummyInteger,
    [In] ref int printToFile,
    [In] ref int readWaveVelFromFile,
    [In] ref bool generatePlots,
    [In] ref bool ExplicitAcceleration,
    [In] ref bool dummyBoolean,
    [In] ref bool dummyBoolean2,
    [In][Out] double[,] result_matrix,
    [In][Out] double[,] maxMatrix,
    [In][Out] double[,] minMatrix,
    [In][Out] double[,] maxAbsMatrix,
    [In][Out] double[,] t_matrix_plot,
    [In][Out] double[,] y_matrix_plot,
    [In][Out] double[,] z_matrix_plot,
    [In][Out] double[,] v_matrix_plot,
    [In][Out] double[,] a_matrix_plot,
    [In][Out] double[,] Fsoil_matrix_plot,
    [In][Out] double[,] Fstiff_matrix_plot,
    [In][Out] double[,] FHrel_matrix_plot,
    [In][Out] double[,] FLrel_matrix_plot,
    [In][Out] double[,] yref_matrix_plot,
    [In][Out] int[,] iElPl_matrix_plot,
    [In][Out] int[,] outvar_int_matrix,
    [In][Optional] char[] resultsPath,
    [In][Optional] char[] caseName);
```

Calling the PILSSInterface_v2 subroutine in C#:

```
PILSSInterface_v2(inCross_real, inHydro_real, inHydro_int, seedArray_phase,
seedArray_crest, inSoil_real,
    inSoil_int,
    inTime_real, inTime_int, ref NrOfTimesteps, ref NrOfTimesteps_signal, ref
NrOfTimesteps_res, inDyn_real,
    inDyn_int, t_vector_U, U_t_matrix, A_t_matrix, F_H_matrix, F_L_matrix,
ref NrOfPlotsteps,
    ref addedPlotRows,
    ref delta_t_plot, ref dummyInteger, ref printToFile, ref
readWaveVelFromFile, ref generatePlots,
    ref ExplicitAcceleration, ref dummyBoolean,
    ref dummyBoolean2, result_matrix, maxMatrix, minMatrix, maxAbsMatrix,
t_matrix_plot, y_matrix_plot,
    z_matrix_plot,
    v_matrix_plot, a_matrix_plot, Fsoil_matrix_plot, Fstiff_matrix_plot,
FHrel_matrix_plot,
    FLrel_matrix_plot,
    yref_matrix_plot, iElPl_matrix_plot, outvar_int_matrix,
resultsPath:resultsPath, caseName:caseName);
```

Example of calling PILSS_DLL.dll legacy API from VBA:

Defining the legacy PILSSInterface Fortran subroutine signature in VBA:

```
Public Declare Sub PILSSInterface Lib "PILSS_DLL.dll" ( _

    ByRef inpCrossSection As Double, _

    ByRef inpHydro_real As Double, ByRef inpHydro_int As Long, ByRef seedArray_phase As
Long, ByRef seedArray_crest As Long, _

    ByRef inpSoil_real As Double, ByRef inpSoil_int As Long, _

    ByRef inpTime_real As Double, ByRef inpTime_int As Long, ByRef NrOfTimesteps As Long,
ByRef NrOfTimesteps_signal As Long, ByRef NrOfTimesteps_res As Long, _

    ByRef inpDyn_real As Double, ByRef inpDyn_int As Long, _

    ByRef t_vector_U As Double, ByRef U_t_matrix As Double, ByRef A_t_matrix As Double,
ByRef F_H_matrix As Double, ByRef F_L_matrix As Double, _

    ByRef NrOfPlotsteps As Long, ByRef addedPlotRows As Long, ByRef delta_t_plot As
Double, ByRef stepLength As Long, ByRef resultsToFile As Long, ByRef readWaveVelFromFile
As Long, ByRef bGeneratePlots As Boolean, ByRef ExplicitAcceleration As Boolean, ByRef
dummyBoolean1 As Boolean, ByRef dummyBoolean2 As Boolean, ByRef result_matrix As Double,
ByRef maxMatrix As Double, ByRef minMatrix As Double, ByRef maxAbsMatrix As Double, ByRef
t_matrix_plot As Double, ByRef y_matrix_plot As Double, ByRef z_matrix_plot As Double,
ByRef v_matrix_plot As Double, ByRef a_matrix_plot As Double, ByRef Fsoil_matrix_plot As
Double, ByRef Fstiff_matrix_plot As Double, ByRef FHrel_matrix_plot As Double, ByRef
FLrel_matrix_plot As Double, ByRef yref_matrix_plot As Double, ByRef iElPl_matrix_plot As
Long, ByRef outVar_int_matrix() As Long)
```

Calling the legacy PILSSInterface subroutine in VBA:

```
Call PILSSInterface(_

       inpCrossSection(1), _

       inpHydro_real(1), inpHydro_int(1), seedArray_phase(1), seedArray_crest(1), _

       inpSoil_real(1), inpSoil_int(1), _

       inpTime_real(1), inpTime_int(1), NrOfTimesteps, NrOfTimesteps_signal,
NrOfTimesteps_res, _

       inpDyn_real(1), inpDyn_int(1), _

       t_vector_U(1), U_t_matrix(1, 1), A_t_matrix(1, 1), F_H_matrix(1, 1), F_L_matrix(1,
1), _

       NrOfPlotsteps, addedPlotRows, delta_t_plot, stepLength, resultsToFile,
readWaveVelFromFile, bGeneratePlots, ExplicitAcceleration, dummyBoolean1, dummyBoolean2,
result_matrix(1, 1), maxMatrix(1, 1), minMatrix(1, 1), maxAbsMatrix(1, 1),
t_matrix_plot(1, 1), y_matrix_plot(1, 1), z_matrix_plot(1, 1), v_matrix_plot(1, 1),
a_matrix_plot(1, 1), Fsoil_matrix_plot(1, 1), Fstiff_matrix_plot(1, 1),
FHrel_matrix_plot(1, 1), FLrel_matrix_plot(1, 1), yref_matrix_plot(1, 1),
iElPl_matrix_plot(1, 1), outVar_int_matrix(1, 1))
```

Example of calling PILSS_DLL.dll API from Python:

There exist various ways of calling dynamic link libraries from Python. In the *[PILSS installation folder]**example**python*\ folder, an example script – `PILSS_source.py` – for calling the PILSS_DLL.dll from Python is included. This code utilizes `ctypes` in order to read and use the dynamic link library with `CDLL`, and it uses a combination of `ctypes` and `NumPy` in order to format the input data.

To use the script, make sure to fulfil the requirements specified in *Requirements.txt*.

Example of script use:
As the script is carefully organized into sections, it should be easy to apply, while it retains the function of supporting both the legacy `PILSSInterface` signature subroutine, and the `PILSSInterface_v2` subroutine. Below, a step-by-step procedure is presented for the use of the script.

1.1: Output file location
Optional specification of output results file location. This variable is intended for use with `PILSSInterface_v2`.
Optional specification of output case name.
Required specification of the path to installation folder containing the DLL's.

1.2: Required input data
All input data to the simulations are provided in section 1.2 by the means of python variables. These variables are also described in Table C-2.

1.3: Specification of subroutine
Here, one may specify whether one wants to run the legacy `PILSSInterface` signature subroutine, or the `PILSSInterface_v2` signature subroutine. Simply change the variable in this section to 1 for version the legacy, or 2 for the v2 subroutine.

1.4: Python bitness
The script is compatible with 32-bit and 64-bit python, but this has to be specified as there are slight differences in the routines. This may be done in 1.4.

1.5: Postprocessing option
As a lot of local computer memory may be occupied due to additional storage of python variables, this option is included in case one doesn't want to continue postprocessing in python. If applied, further postprocessing may be performed by calling variables presented in Table C-3.



About DNV

We are the independent expert in risk management and quality assurance. Driven by our purpose, to safeguard life, property and the environment, we empower our customers and their stakeholders with facts and reliable insights so that critical decisions can be made with confidence. As a trusted voice for many of the world's most successful organizations, we use our knowledge to advance safety and performance, set industry benchmarks, and inspire and invent solutions to tackle global transformations.

Digital Solutions

DNV | digital@dnv.com | dnv.com/digital



DNV is a world-leading provider of digital solutions and software applications with focus on the energy, maritime and healthcare markets. Our solutions are used worldwide to manage risk and performance for wind turbines, electric grids, pipelines, processing plants, offshore structures, ships, and more. Supported by our domain knowledge and Veracity assurance platform, we enable companies to digitize and manage business critical activities in a sustainable, cost-efficient, safe and secure way.