NPEX Finite Element Analysis Program Release

Unrestricted

SR.16.12288

August 2016

Unrestricted **SR.16.12288**

NPEX Finite Element Analysis Program Release

by

R. Peek (GSNL-PTE/EPFA)

Shell Global Solutions International B.V., Rijswijk

Further electronic copies can be obtained from the Global Information Centre.

Executive summary

NPEX was developed at the University of Michigan as finite element analysis platform (1987-1995) that is developer-friendly in the sense that as open source and license free finite element analysis platform there is a simple, modular and clearly documented protocol for adding new types of elements, together with testing routines to check the consistency of such new elements. These capabilities have been used at Shell (1996-2016) to develop elements to model uniform bending and initiation of wrinkling for pipes, plastic expansion of pipes by a pig and limits to such expansion beyond which the plastic deformations tend to localize, two-dimensional beam type elements to model pipes with internal pressure that may undergo lateral and or upheaval buckling, and some pipe-soil interaction elements to describe the interaction of a buried or surface-laid pipeline with the seabed.

All NPEX developments have been results-driven: the purpose of the past efforts was to generate results, not software for a community of users. Considerable additional effort would be required to make existing capabilities suitable for general use, except that pre- and postprocessors for specific applications of NPEX such as Latbuck, Worm2, and FlawStress are intended to make certain specific applications of NPEX accessible for general use. Since Shell does not expect to develop NPEX in-house, nor to market it, it has been decided to release it for further use and/or development.

This report serves as the release of NPEX. The intent is not so much for retention of critical knowledge (ROCK), but rather, enabling further use of non-critical but still useful knowledge. Source, executable, and available documentation are embedded as an attachment to this report. A brief overview of the background (including references to theory and results published in the open literature), and capabilities is included.

Disclaimer

The use of NPEX and all other programs/elements associated with it are entirely on the user's own risk. Shell provides no guarantee as to the accuracy of the results calculated with NPEX. Although certain opportunities for validation against other codes, methods of analysis, and/or experiments have been exploited, some of which is reported in referenced papers and reports, NPEX can be used in a wide variety of different ways, and there has been no systematic validation of all such possible uses of NPEX. Therefore there can be no guarantee of the accuracy of the results from NPEX.

Table of contents

1. Background

NPEX started as a Programming EXercise (PEX) for graduate students at the University of Michigan (UM). Then capabilities for nonlinear analysis, including stability analysis were added, together with the letter "N" to the name of the program. This program together with the related LSKFE[1] have been used at UM for a number of challenging elastic stability problems [1-7].

Later, at Shell, additional elements were added to NPEX mainly with the intent of producing reported results, rather for the sake of creating software. These include:

- Generalized plane strain elements (nelmt13)[2] to model uniform bending of pipes, and perform a bifurcation for the initiation of wrinkling according to a buckling mode for which the incremental displacements vary sinusoidally along the length of the pipe [8].
- Application of the same finite strain generalized plane strain elements (nelmt13) to model the (downhole) expansion of a pipe by a pig (nelmt19), and via the bifurcation check the limits to such expansion, beyond which the plastic deformations will tend to localize [9].
- Axisymmetric buckling and postbuckling of pipe including an implementation of the incrementally continuous deformation theory of plasticity with unloading ("ICU deformation theory of plasticity) [10,11], also using the generalized plane strain elements (nelmt13).[3]
- Two-dimensional beam type elements (nelmt25 and nelmt29) to model upheaval buckling [12, 13], and lateral buckling [14].
- Special elements (nelmt22) to model the contribution of concrete coating to the flexural stiffness of pipes, including the effect of cracking of the concrete, and limited bond strength according to the formulation of Verley and Ness [16, 17]
- Simple linear-elastic 2 node elements (nelmt24) with 1 degree of freedom per node to model pipeline thermal expansion and walking when this involves only axial displacements of the pipeline and not lateral or upheaval buckling, or transverse displacements due to some other reason.
- Spring-type elements to capture pipe-soil interaction for lateral buckling (nelmt17, 27, 28, 33) including berm formation (nelmt27), upheaval buckling with cohesionless cover (nelmt16), and a general nonlinear elastic spring used to model ice gouging over a buried pipe (nelmt33).

Documentation for the various elements may be found in the npex.txt file[4], and within the source code, e.g. in file "nelmt27.f90" for nelmt27, as well as in the referenced publications.

[1] LSKFE performs nonlinear elastic finite element analysis together with Lyapunov-Smidt-Koiter (LSK) decomposition and asymptotic expansion in the vicinity of (multiple) bifurcation points.

[2] This refers to the element name in NPEX. The last 2 digits are the element number. The source code file for this element is nelmt13.f90, with other elements similarly named. Typically it is necessary to go back to comments in the element source code, for the definition of coordinates and degrees of freedom used by the element, and the interpretation of output parameters generated.

[3] Axisymmetric conditions are a special case of generalized plane strain, in which the lubricated sticky planes remain fixed.

[4] Within this report any reference to a "file" refers to a file contained the zip file included as Appendix 1.

2. NPEX Approach, Concepts and Key Terms

A clear distinction has been maintained between the NPEX code itself, and the element routines it uses.

The code itself has been developed entirely by the author of this report, with emphasis placed in keeping it simple and reasonably compact, yet with powerful and general possibilities. NPEX recognizes nodes, which may have any number of coordinates and any number of degrees of freedom per node, and it recognizes elements which may be attached to any number of nodes. NPEX itself does not distinguish between volume, surface, line, and spring elements. They are all treated in exactly the same way. Only the user-specified number of coordinates per node, and number of degrees of freedom per node may differ.

"Element properties" refers to any input parameter that is the same for all elements in an "element group"; element "parameters" are different for each element in an element group, and include for instance the state variables that capture the plastic deformation history for every integration point in the element. The element "properties" cannot change in time, but it is possible to use "properties" to define how properties change with time.[5] Loads such as internal pressure in a pipe element are also defined by "properties". For instance a "property" may be a factor which is multiplied by a user-specified time function to determine the internal pressure.

What in Abaqus[6] is referred to as an "increment" in NPEX is referred to as a "loadstep". This refers to the change from one converged solution to the next. NPEX uses standard Newton iteration to calculate converged solutions. No other solution algorithms are available in NPEX. Experience with this has been good. In NPEX the answer to convergence difficulties from one "loadstep" to the next, must be smaller increments, or better definition of the problem (e.g. to include the real physics rather than artificial physics that gives rise to the convergence problem), rather than more sophisticated algorithms that may enable larger increments, but risk converging to the wrong solution, or sacrificing accuracy in the time-integration of the history-dependent plastic behaviour of the material.

Since NPEX was originally developed for challenging structural stability problems, a generalized version of Riks solution path arclength increment control [18] is standard for NPEX. This enables unstable as well as stable portions of the solution paths to be calculated. Standard time increment control is also available as a special case of the generalized arclength control algorithm. Dynamic analysis is also available but only with time-increment control, and not (yet) generalized to enable arclength increment control within a dynamic analysis.[7]

It is the element routines that establish the meaning of the nodal coordinates and degrees of freedom. It is up to the user to make sure that these are compatible. NPEX itself treats all nodes and degrees of freedom in the same way. NPEX has been structured in such a way that it is straightforward to add new elements, and this process is described in file nelmt.doc. Graduate students who have added new elements include M. Kheyrkhahan [1, 2, 4] and J. van Hilten [15].

The process of debugging new element routines is aided by a program "tst17" that will test the consistency of the tangent stiffness matrix. It does this by calculating the tangent stiffness matrix

[5] For this purpose "properties" need to include information on how the properties change in time: e.g. if it is a linear variation, the initial value and rate of change must be included in "properties". More generally a series of (time,value) points can be included whereby the element routines can determine the value as a function of time by linear interpolation between the (time,value) points included in the "properties".

[6] Abaqus, a general purpose and commercially available finite element package by Dassault Systèmes.

[7] To the author's knowledge dynamic integration algorithm using arclength control do not yet exist, but are a good idea to develop for case when essentially quasi-static response changes to dynamic response due to loss of stability.

by numerical differentiation[8], and checking whether the tangent stiffness matrix calculated in this way agrees with the one that is calculated by the element routine. Similarly “tst25” can be used to test certain material routines.

Even users that do not develop a new element may need to look at the source code of the element routines, to understand definition of the nodal coordinates, element nodes, and degrees of freedom, as well as the output parameters for the element used. For some elements there is also additional element specific documentation in files carrying the same name as the element.[9] On the other hand the file npex.txt should be consulted to see which element might do the job.

NPEX is quite limited in providing diagnostics of errors in the input files. The author has at times run NPEX in the debugger to find what turned out to be errors in the inputs.

[8] Performed by multiple calls of the element routine with small differences in the input nodal displacements, to determine the corresponding change in the element out-of-balance force vector.

[9] Together with an extension indicative of the type of file.

3. NPEX Procedures

3.1. Quasi-Static Analysis by Generalized Arclength Algorithm

For quasi-static analysis, NPEX uses a generalized version of Riks' arclength control. Here "arclength" refers to the length measured along the "solution path". The "solution path" is a path in N+1 dimensional load-displacement plot, that shows the solutions for varying values of the load. (Here N denotes the total number of unknown nodal displacement or rotation components to be determined).

The load axis on this load-displacement plot is a parameter called "TIME" in NPEX, though it need not correspond to real time, except in dynamic analysis. All loads need to be defined as a function of "TIME". This includes loads due to thermal expansion. A consequence of this interpretation of "TIME" as a load parameter, rather than real time in quasi-static analysis is that "TIME" can decrease, as well as increase. NPEX has no limitations as to what sort of loading can be included when using Riks arclength control method. Any available type of loading is also available with the arclength method.

The arclength control method is generalized by introducing weighting factors WFU on displacement components, and WFL on the load parameter "TIME". These are used to calculate the arclength increment as in

$\Delta\eta = \{\ \mathrm{WFL}\ \ \Delta t^2 + \ \Sigma\ \mathrm{WFU}\ \Delta u^2\ \}^{1/2}$

where $\Delta\eta$ denotes the increment in arclength[10] that is controlled at each "loadstep"; Δt denotes the increment in the load parameter "TIME"; Δu denotes an increment in nodal displacement or rotation, and Σ indicates the summation over all displacement and rotation degrees of freedom. The weighting factors WFU can have a different value for each degree of freedom, and are specified as input by the user in the same way as applied nodal loads or applied nodal displacements are specified.

By picking WFU=0 for all degrees of freedom and WFL=1 one recovers the normal time stepping algorithm, and the specified arclength increment becomes the time increment.

Alternatively, by picking WFL=0, and WFU=0 for all degrees of freedom, except WFL=1 for a selected one, one controls the displacement increments for that degree of freedom, with the arclength increment becoming that displacement increment. Sometimes it can be an advantage to use this approach, for instance when modeling upheaval buckling triggered by a prop imperfection on an otherwise flat seabed. In this case the selected degree of freedom with WFU=1 is the uplift displacement at the prop (Other points may well move down first and then up, which makes them less suitable as points for which one controls the displacement increments).

Further details of arclength algorithm are described in the file npex.txt. See heading "=== PROCEDURE FOR RIKS ARCLENGTH METHOD" therein.

3.2. Dynamic Time-History Analysis

For dynamic time integration, the implicit Hilber, Hughes, Taylor (HHT) [19] algorithm is used, as in Abaqus. This has 2nd order accuracy and includes numerical damping, which means that frequencies that are in any case too high to be accurately resolved by the time increment used are

[10] It is not exactly the increment in arclength, but approximately so. What is controlled as "DETA" from one loadstep to the next is also not exactly the same as this $\Delta\eta$, but approximately so. These approximations become asymptotically exact in the limit as the increments become small, with $O(\Delta\eta^3)$ errors, i.e. the ratio of the error divided by $\Delta\eta^3$ remains bounded as $\Delta\eta \rightarrow 0$.

damped out. Of course 2nd order accuracy means that the error (including any algorithmic damping for lower frequencies) are of order $O(\Delta t^2)$, and thus should become small quite rapidly as the time increments are decreased. Of course this means that algorithmic damping becomes small for frequencies that are properly resolved, as it should. By picking $\alpha=0$ in for the HHT algorithm, numerical damping is eliminated, and the scheme is the same as the Newmark $\beta=1/4$ scheme, which is also sometimes referred to as the trapezoidal rule.

To date mostly only the quasi-static capabilities of NPEX have been used, with the only dynamic analyses reported being those of van Hilten [15], which involved combination of lateral buckling and on bottom stability analysis of a pipeline on the seabed, in which waves can give rise to imperfections that then trigger lateral buckles.

NPEX does not currently have the capability for dynamic modal analysis based on a linearization of the system, as would be required for instance for the modal analysis of pipeline spans. It does however include eigenvalue analysis for stability, as described in the next section.

3.3. Stability Analysis

NPEX can calculate the eigenvalues and eigenvectors of tangent stiffness matrix or stability matrix. For this purpose it can use different boundary conditions for the stability check. This is useful if there is a symmetric prebifurcation solution followed by bifurcation into an asymmetric mode.

Also the element routine may be programmed to return different tangent stiffness matrices for the Newton iterations and the stability check. For instance in certain problems of plastic buckling it has been found than the tangent stiffness matrix based on the deformation theory of plasticity gives a better indication of when buckling occurs than that based on the flow theory of plasticity. This capability also enables the bifurcation check for sinusoidal buckling or wrinkling modes for the generalized plane strain elements, as reported in [8]. Even the number of degrees of freedom per node may be different for bifurcation check.

The eigenvectors (i.e. buckling modes) from the stability analysis can also be used for a "branch-switching" restart[11]. This applies at bifurcation points, where two or more solutions paths intersect, and one would like to the calculated solution to switch from one path (or "branch") to the other. However in practice it may be easier and more representative of the real system, to introduce a symmetry-breaking imperfection that removes the bifurcation and the need for any branch-switching restart.

It is not necessary to solve the eigenvalue problem to assess stability. NPEX also can output the number of negative eigenvalue of the tangent stiffness matrix at every loadstep. Thus the appearance of negative eigenvalues signals loss of stability of the solution. If this happens with increasing load parameter "TIME", it suggests a bifurcation point has been passed with the solution proceeding on the unstable portion of the principal solution branch. However if stability is to be assessed based on a stability matrix different from the tangent stiffness matrix used in the Newton iterations, the only way to currently do this in NPEX is to first obtain the principal solution, storing the state at the loadsteps for which stability is to be checked, and then perform a restart to do the bifurcation check.

[11] Bifurcation points mean a bifurcation of the solution path. Mostly the solution on a bifurcated solution branch is of more interest than the continuation of the principal solution branch (that starts from zero load and displacement), since the bifurcated branch describes the postbuckling behaviour for the system.

4. Limitations

Current limitations of NPEX include:

1) It has no graphical user interface (GUI). This can make it difficult to debug models without making significant efforts to plot erroneous results in order to understand where it has gone wrong.
2) There are minimal diagnostics for input errors. (At times it has been necessary to use a code debugger, in order to identify what turned out to be input errors).
3) It is not possible in general to change models as you go, e.g. to change a boundary condition half way through an analysis, or to change some inputs half way through the analysis. In most cases any changes must be anticipated, e.g. by including special elements that can turn themselves on an off, or change in some other way.
4) The tangent stiffness matrix must be symmetric. One can approximate a non-symmetric tangent stiffness matrix by a symmetric one, but could lead to failure of the Newton iterations to converge, or to slower (non-quadratic) convergence of the Newton iterations at each "loadstep". Frictional phenomena for instance give rise to a non-symmetric tangent stiffness matrix. This is the case for the axial resistance from the bottom of trench in the pipe-soil-interactions elements for upheaval buckling (nelmt16). There the tangent stiffness matrix has been symmetrized by ignoring the interaction between the vertical and axial directions (i.e. setting the corresponding off-diagonal term that arises to one side of the diagonal to zero). Experience shows that this works satisfactorily for values of the friction factor up to 0.3.

   As long as the Newton Iterations are taken to convergence to a tight tolerance, approximating a non-symmetric tangent stiffness matrix by a symmetric one does not influence the accuracy of the results, only the number of Newton needed at each loadstep to reach a given accuracy is affected.[12]
5) Currently NPEX does not have capabilities for 3-dimensional continuum elements to model solids. These could quite readily be added.

[12] In this case convergence is exponential rather than quadratic. This means that the errors decrease exponentially with then number of Newton interactions once they are converging, whereas with quadratic convergence the error at the next iteration is of the order of the current error squared, which means that the number of digits to which the solution is calculated accurately doubles at each iteration until a limit determined by truncation errors in the floating point operations is reached.

5. User Interface

5.1. Input

Input is provided in a text-based question and answer session. Output is also a text file, including limited explanations of the results provided. The user is given some control on how much output to produce. Creating plots of the results is something that needs to be done by the user of NPEX, e.g. by importing the results into any available plotting enabled software such as MS Excel.

All operations are carried out in the same directory which also contains the executable file. NPEX first looks for input in the npex.u9 file, if this is absent it prompts the user for inputs. If the npex.u9 file ends before all input has been provided, it prompts the user for the missing input. To start a new problem the npex.u9 file should be first renamed or removed, so that NPEX will prompt the user for input from the start.

The output includes the npex.u13 file containing history parameters defined in the input file, and standard (console) output. The console output can be routed to a file, with a command line such as

```
npex_32.exe >npex.u6
```

This will work if the file `npex.u9` is present and contains all the required inputs. The file npex.u6 will then be created containing the console output, which includes the prompts generated by NPEX for the inputs.

5.2. Output

The output also includes a binary restart file called `npex.u7`. This can be used to restart NPEX from a chosen state. For a restart the loadstep number from which the restart is to be performed must be specified as the value of "IST0" in the input. Caution is required to ensure that the results are available for the desired restart point. (Depending on selected printout control options (see "N7PRINT" in the input or output files) the restart information may not be saved for every loadstep.)

If no npex.u9 file is present, it is also possible to specify both the input and the output file in the following single command line:

```
npex_32.exe  <fn_input  >npex.u6
```

where "fn_input" denotes the file name of the input file, which may be an npex.u9 file that and been renamed and edited to reflect desired changes in the inputs.

5.3. Executables and Compilation

Appendix 1 includes a 32-bit executable (`npex_32.exe`) and a 64-bit executable (`npex_64.exe`) for MS Windows.[13] These can be used to run on any windows machine, developer or non-developer. Of course it is also possible to use the source files to compile other versions. A list of all source files included in NPEX is included in the file "npex.fls" which can be used in instructing a compiler which files to use to create the NPEX executable. For instance

[13] For 32-bit machines, only the 32-bit version can be used. For 64-bit machines, both the 32-bit and the 64-bit versions can be used. The 64-bit version can access more of the machine's random access memory (RAM) and this can make it run more efficiently, if it eliminates the need for virtual memory (i.e. using disk instead of RAM). These versions have been compiled with Lahey Fortran 95, and a version of gfortran also provided by Lahey, respectively.
Here "developer" machine refers to which user of this machine is able to add/install new non-GI software.

the 32-bit version is been created with Lahey Fortran using the dos command “`lf95 @npex.fls`” to compile the program.

6. Conclusions

This report is not so much intended for retention of critical knowledge (ROCK) as it is intended to enable further use and development of non-critical but still useful information and software. It allows other developer outside Shell Companies to use, modify, further develop the NPEX program at own risk without restriction and/or violating Shell Intellectual Properties (IP) right.

References

Elastic Stability Problems Addressed with NPEX (1990-1995)

1. Kheyrkhahan, M. and Peek, R., (1999), "Postbuckling Analysis and Imperfection Sensitivity of General Shells by the Finite Element Method," International Journal of Solids and Structures, Vol. 36, pp. 2641-2861.
2. Kheyrkhahan, M, (1995) "Lyapunov-Schmidt-Koiter Singular Asymptotic Expansions for Postbuckling Analysis and Imperfection Sensitivity of General Shells with Multiple Coincident or Nearly Coincident Buckling Modes." Ph.D. Thesis, University of Michigan, Ann Arbor, Michigan, 1995.
3. Triantafyllidis, N. and Peek, R., (1995), "Post-Bifurcation and Imperfection Sensitivity of Space Trusses with Many Simultaneously Buckling Bars and a Strongly Nonlinear Prebuckling Solution," European Journal of Mechanics.
4. Peek, R. and Kheyrkhahan, M., (1993), "Postbuckling Behavior and Imperfection Sensitivity of Elastic Structures by the Lyapunov-Schmidt-Koiter Approach," Computer Methods in Applied Mechanics and Engineering, Vol. pp. 261-279.
5. Peek, R., (1993), "Worst Shapes of Imperfections for Space Trusses with Multiple Global and Local Modes," International Journal of Solids and Structures, Vol. 30, No. 16, pp. 2243-2260.
6. Peek, R. and Triantafyllidis, N., (1992), "Worst Shapes of Imperfections for Space Trusses with Many Simultaneously Buckling Members," International Journal for Solids and Structures, Vol. 29, No. 19, pp. 2385-2402.
7. Triantafyllidis, N. and Peek, R., (1992), "On Stability and Worst Imperfection Shape in Solids with Nearly Simultaneous Eigenmodes," Int. J. Solids Structures, Vol. 29, No. 18, pp. 2281-2299.

Published NPEX results from Elements Developed at Shell (1995-2016)

8. Peek, R., (2002), "Wrinkling of Tubes in Bending using Large Strain Continuum Theory," International Journal of Solids and Structures, Vol. 39, pp. 709-723. (SIEP Disclosure/Ref. No. EP 2000-8171.)
9. Peek, R.; Marketz, F.; Filippov, A.G.; Nyhus, B.: Forming Limits for Expansion of Tubes, Proceedings of the Fifth World Congress on Computational Mechanics (WCCM V), July 7-12, 2002, Vienna, Austria, Editors: Mang, H.A.; Rammerstorfer, F.G.; Eberhardsteiner, J., Publisher: Vienna University of Technology, Austria, ISBN 3-9501554-0-6.
10. Peek, R., (2000), "An Incrementally Continuous Deformation Theory of Plasticity with Unloading," International Journal of Solids and Structures, Vol. 37, pp. 5009-5032.
11. Peek, R., (2000), "Axisymmetric Wrinkling of Cylinders with Finite Strain," Journal of Engineering Mechanics, ASCE, Vol. 126, No. 5, pp. 455-461.
12. Peek, R. and Stewart, G., (1998), "Upheaval Buckling of Reeled Flowlines," OMAE'98. (Oral presentation and hardcopy of paper provided to conference attendees only.)
13. "Upheaval buckling of reeled flowlines", Shell GSI, Report No. SR.12.12579, unrestricted. (Contains copy of [12])
14. Peek, R., and Yun, H., (2009), "Scaling of Solutions for the Lateral Buckling of Elastic-Plastic Pipelines," Journal of Offshore Mechanics and Arctic Engineering, ASME, Vol. 131, August 2009, pp. 031401-1 to 031401-9.
15. Van Hilten, J.M., (2009), "On-Bottom Stability of High-Temperature Pipelines," MSc Thesis, Offshore Engineering, Technical University Delft, The Netherlands, 15 June 2009.

Other References

16. Verley, R., and Ness, O.B., (1995), “Strain Concentrations in Pipelines with Concrete Coating: Full Scale Bending Tests and Analytical Calculations,” Proc. OMAE 1995, Vol. 5, Pipeline Technology, ASME.
17. Ness, O.B. and Verley, R., (1996), "Strain Concentrations in Pipes with Concrete Coating," Journal of Offshore Mechanics and Arctic Engineering, Vol. 118, pp. 225-231, August 1996.
18. Riks, E. and Rankin, C.C., (1987), "Bordered Equations in Continuation Methods: An improved Solution Technique," NLR MP 87057U, National Aerospace Laboratory, The Netherlands.
19. Hilber, H.,M., Hughes, T.J.R., and Taylor, R.L., “Collocation, Dissipation and ‘Overshoot’ for Time Integration Schemes in Structural Dynamics,” Earthquake Engineering and Structural Dynamics, Vol. 6, pp. 99-117, 1978.

Appendix 1. Embedded attachments

This report contains attachments that are embedded in the Word file.

appendix 1.docx

Word file containing a zip file containing npex source, executables, and documentation. This has been encrypted to enable it to be sent by email. Password is 2_Evade_Filter.

Bibliographic information

Classification	Unrestricted
Report Number	SR.16.12288
Title	NPEX Finite Element Analysis Program Release
Author(s)	R. Peek (GSNL-PTE/EPFA)
Keywords	Finite Element, Software, Fortran, Pipeline, Stability, Generalized plane strain
Date of Issue	August 2016
US Export Control	US - Non Controlled (EAR99)
Reviewed by	S. Y. Ang (GSNL-PTE/EPFA)
Approved by/Content owner	W. Guijt (GSNL-PTE/EPFA)
Sponsoring Company / Customer	Shell P&T
Issuing Company	Shell Global Solutions International B.V., Rijswijk P.O. Box 60 2280 AB Rijswijk The Netherlands

Report distribution

Electronic distribution (PDF)

Name, Company, Ref. Ind.	*PDF*
PT Information Services, PTT/TIKE, PT-Information-Services@Shell.com	Word + PDF
Ang, Sze Yu Y GSNL-PTE/EPFA	PDF
Guijt, Wim GSNL-PTE/EPFA	PDF
Simons, Servie JM GSNL-PTE/EPFA	PDF
Carr, Malcolm GSNL-PTE/EPFA	PDF
Chang, DongDong GSUSI-PTE/ACSO	PDF
Hill, Mike M SIEP-PTP/D/ADS	PDF
Nobahar, Arash SIEP-PTU/E/Q	PDF
Papadopoulos, Dimitrios GSNL-PTE/EPFA	PDF