A computational strategy to establish algebraic parameters for the Reference Resistance Design of metal shell structures

Adam J. Sadowski*, O. Kunle Fajuyitan, Jie Wang
Department of Civil and Environmental Engineering, Imperial College London, United Kingdom

ARTICLE INFO
Article history:
Received 14 December 2016
Revised 25 February 2017
Accepted 27 February 2017
Available online 18 March 2017

Keywords:
Metal shell structures
Reference Resistance Design
Buckling
Plasticity
Finite element analysis
Automation
Object-oriented programming

ABSTRACT
The new Reference Resistance Design (RRD) method, recently developed by Rotter [1], for the manual dimensioning of metal shell structures effectively permits an analyst working with only a calculator or spreadsheet to take full advantage of the realism and accuracy of an advanced nonlinear finite element (FE) calculation. The method achieves this by reformulating the outcomes of a vast programme of parametric FE calculations in terms of six algebraic parameters and two resistances, each representing a physical aspect of the shell’s behaviour.

The formidable challenge now is to establish these parameters and resistances for the most important shell geometries and load cases. The systems that have received by far the most research attention for RRD are that of a cylindrical shell under uniform axial compression and uniform bending. Their partial algebraic characterisations required thousands of finite element calculations to be performed across a four-dimensional parameter hyperspace (i.e. length, radius to thickness ratio, imperfection amplitude, linear strain hardening modulus).

Handling so many nonlinear finite element models is time-consuming and the quantities of data generated can be overwhelming. This paper illustrates a computational strategy to deal with both issues that may help researchers establish sets of RRD parameters for other important shell systems with greater confidence and accuracy. The methodology involves full automation of model generation, submission, termination and processing with object-oriented scripting, illustrated using code and pseudocode fragments.

© 2017 The Authors. Published by Elsevier Ltd.
This is an open access article under the CC BY license. (http://creativecommons.org/licenses/by/4.0/)

1. Introduction

Shells are widely recognised to exhibit the greatest complexity of any structural form, a consequence of a behaviour governed by local buckling, global nonlinear collapse, plasticity and sensitivity to various imperfection forms, in addition to a myriad of possible geometries and load conditions [2]. The power of modern computing, coupled with the recent development of a powerful framework to characterise the results in a uniform manner, have led to the development of Reference Resistance Design (RRD) [1]. Without this framework, every problem must be treated in an ad-hoc manner. However, RRD requires a huge volume of nonlinear calculations required to support it. This paper presents a new and highly efficient technique to achieve that goal.

Although early advances in the analysis of shell buckling problems were made through algebraic studies [3–5], it did not take long for the set of shell structural systems that could be processed in this manner to become exhausted and recourse had to be made to numerical methods, first in the solution of the equations and ultimately by finite element analysis. The history of progress in shell buckling thus runs in parallel with the history of the development of modern scientific computing [6–8]. While the earliest computational endeavours saw researchers develop their own finite element solvers (e.g. BOSOR [9] and NEPAS [10]), the era of writing software from scratch has now largely passed as the range and complexity of the problems under investigation have burgeoned. A look at the modern engineering research literature reveals a widespread use of extensive ‘general’ FE software suites, both proprietary (e.g. ABAQUS [11]) and open-source (e.g. OpenSees [12]), that have benefited from years of continuous development. Many more design-focused FE packages are used by industry.

2. An overview of Reference Resistance Design

The state-of-the-art European Standard on the Strength and Stability of Metal Shells EN 1993–1–6 [13] was the first in the world to prescribe two formal methods governing the computer-aided
design of metal shells under the generic title ‘design by global numerical analysis’ [14]. The first and simplest method is ‘LBA-MNA’ [15], where the designer must use a computer calculation to determine the reference elastic critical buckling and plastic collapse resistances of the perfect shell from a linear bifurcation (eigenvalue) analysis (LBA) and an ideal elastic-plastic materially nonlinear but small displacement analysis (MNA) respectively. The second method is termed ‘GMNIA’ [16,17] and involves full geometrically and materially nonlinear analyses with explicitly modelled imperfections. Analyses of intermediate complexity are also defined in this standard, termed GNA, GMNA and GNIA, where ‘G’ and ‘M’ indicate the inclusion of geometric and material nonlinearity respectively, while ‘I’ indicates the presence of imperfections. Neither of these computer-aided methodologies come easily to many practitioners, who may lack the training, the time or the software necessary to undertake nonlinear FE analyses on anything but the most important of designs. Indeed, the ECCS European Design Recommendations on the Buckling of Metal Shells [14] offers a commentary on EN 1993-1-6 and devotes much discussion to how such analyses should be conducted, illustrating many of the potential pitfalls that can make a calculation go wrong or be badly misinterpreted.

\[ \chi(\lambda) = \begin{cases} 
\text{Fully plastic: } \chi_h - \left( \frac{\lambda}{\lambda_0} \right) (\chi_h - 1) & \text{when } \lambda \leq \lambda_0 \\
\text{Elastic-plastic: } 1 - \beta \left( \frac{\chi_h - \chi_0}{\lambda_h - \lambda_0} \right) & \text{when } \lambda_0 < \lambda < \lambda_p \\
\text{Fully elastic: } \frac{\alpha_C \alpha_I}{1 - \beta} & \text{when } \lambda_p \leq \lambda 
\end{cases} \]

where \( \lambda_p = \sqrt{\frac{\alpha_C \alpha_I}{1 - \beta}} \) (1)

The new method termed Reference Resistance Design (RRD) was recently devised by Rotter [12,18] to permit analysts to take advantage of advanced computational predictions without having to perform these themselves. RRD is built around a capacity curve which describes the base relationship between a shell’s nonlinear characteristic resistance normalised by its reference plastic collapse resistance \( \lambda(\chi) = R_b/R_p \) and its dimensionless slenderness \( \chi = \sqrt{R_p/R_c} \), where \( R_c \) is the reference elastic critical buckling resistance. The entire relationship is characterised by six independent algebraic parameters \( \alpha_C, \alpha_I, \beta, \eta, \lambda_0 \) and \( \chi_0 \), each one containing information about a real physical effect as described shortly. The most recent formulation of the capacity curve [17] is given in Eq. 1 and illustrated schematically first as a classic buckling curve in Fig. 1a and again in a modified form [19] in Fig. 1b. A detailed account of background to the capacity curve and proposed further enhancements may be found in Rotter [15] and De-erich and Rotter [20].

In Eq. 1, \( \alpha_C \) and \( \alpha_I \) are separate adjustments on \( R_c \), respectively account for the influences of geometric nonlinearity and imperfection sensitivity in slender shells that fail elastically. As most shell systems under simple load conditions suffer from prebuckling geometric softening and detrimental imperfection sensitivity, \( \alpha_C \) and \( \alpha_I \) are usually less than unity. However, more complex systems such as cylindrical silos under unsymmetrical eccentric discharge pressures may exhibit beneficial changes of geometry leading to geometric stiffening and/or strengthening ‘imperfections’ [21,22], such that \( \alpha_C \) and \( \alpha_I \) may become greater than unity. In theory, this very general formulation supports both possibilities. Although \( \alpha_C \geq 1 \) cannot be avoided if the beneficial geometric nonlinearity relates to the fundamental mechanics of the system, a situation where \( \alpha_I > 1 \) usually signifies that an inappropriate ‘imperfection’ has been chosen and it is likely that a more carefully-chosen deviation from the perfect geometry will lead to \( \alpha_I < 1 \) [22].

The ‘plastic range factor’ \( \beta \) identifies the point \( (\lambda, \chi) = (\lambda_p, 1 - \beta) \) on the capacity curve at which plasticity first begins to have a noticeable effect on the behaviour and marks the onset of the ‘elastic-plastic’ region. The ‘interaction exponent’ \( \eta \) controls the curvature of the relationship in this region of Fig. 1a, with \( \eta < 1, > 1 \) and \( \eta = 1 \) allowing convex (positive curvature, indicating a strong elastic-plastic interaction), concave (negative curvature, indicating a weak interaction) and idealised linear (zero curvature) relationships respectively in the \( \chi \) vs \( \lambda \) plane. The ‘squat limit’ \( \lambda_0 \) defines the slenderness at which the reference full plastic conditions \( R_p \) is reached, while \( \chi_h \geq 1 \) defines the projected intercept of the vertical axis for a fictitious shell with zero slenderness, allowing for resistances higher than \( R_p \) for very stocky shells due to geometric or strain hardening.

In effect, RRD permits an analyst working with only a calculator or a spreadsheet to take full advantage of the accuracy and realism of a GMNIA, on the understanding that the six algebraic parameters that account for the complex nonlinear phenomena have already been established \textit{a priori} for all practically relevant parameter ranges. While it is possible in theory to establish these on the basis of laboratory testing, the expense involved in doing so for all systems and parameter ranges would be absolutely prohibitive and difficult to justify in the modern age of limited budgets, powerful software and cheap computing power. A computational programme using a validated and robust FE model remains the only defensible option. The significant reduction in uncertainty in the structural resistance model obtained through RRD should reduce the need for excessive compensation through high partial safety factors, permitting more economic designs.
The RRD method has now been accepted for use in design through an Amendment to EN 1993-1-6 [13] coupled with the creation of a new normative Annex E that will hold RRD expressions for algebraic parameters and other information [18,23]. Two systems have currently been approved based on available research, cylindrical shells under uniform global bending and spherical shells under uniform external pressure, but the call is out to the research community to gradually extend this set to cover new geometries and load cases. In particular, Rotter [2] lists a number of important systems that are ripe for implementation in the RRD format, including tanks and silos with stepped walls, lap-jointed construction, wind loading, stiffened and corrugated shells, localised loading, conical shells under external pressure, cylinders with openings and cutouts, shells with ringbeams and others.

3. Scope of the study

Although RRD and the capacity curve together form a robust framework for the manual dimensioning of shell structures, the challenge of establishing a database of algebraic parameters for the most important geometries and load cases is formidable. A conceptual strategy for doing so is presented in Section 6 of [2], which readers are warmly invited to consult. The aim of this paper is to complement the above with a practical computational strategy on how to design parametric studies for RRD specifically and perform the necessary calculations. In particular, the authors take the view that in order to establish such a database, the international shell buckling research community should embrace the full power of automation now widely available in modern computing to aid in the generation, submission, termination and processing of the many thousands of necessary parametric LBA, MNA, GNA, GNIA and GMNIA studies. The powerful commercial ABAQUS 6.14-2 [11] finite element package is employed here which links with the object-oriented Python and compiled FORTRAN programming languages for extended functionality, although the logic contained in the code and pseudocode examples given may readily be adapted to any other software with a scripting facility. The use of ABAQUS for illustration is not inappropriate, as it has arguably been the most widely used finite element package in the past two decades for shell buckling research (e.g. [7,24–30], among many others).

An outline Python code is presented here, based on a novel conceptual analysis hierarchy, for scripts that automate the mass generation of FE model input files in an ordered manner. It is then shown how to arrange these model files within ordered subdirectories based on the varied system variable, permitting automated submission to the FE solver in simultaneous batches of approximately equal numbers of models. Fully-functional FORTRAN code for ABAQUS user subroutines that permit the automated termination of nonlinear equilibrium path-tracing ‘Riks’ analyses once specific criteria are met, a vital capability, is presented in its entirety in Appendix A. Finally, an object-oriented post-processing methodology for the automated extraction of the RRD algebraic parameters is presented. The intention at this stage is to show how these can be obtained as accurately and as completely as possible, even if ultimately only simplified relationships are processed into a RRD design rule. For illustration purposes, reference is made throughout to the structural system that has, to date, possibly received the most research attention in this regard, cylindrical shells under uniform bending. This system finds ubiquitous application in generic load-bearing circular hollow sections, pipes, pipelines, chimneys, wind turbine support towers and tall slender silos that may be subject to transverse loading, wind, seismic or other actions that induce a global bending moment.

4. The analysis hierarchy

It is proposed here that any metal shell system being characterised can be treated abstractly as a data tree according to an ordered hierarchy of analysis Levels. A physical variable, varied within a particular Level, is held constant when ascending to a higher Level along the same branch. A flowchart illustrating this is shown in Fig. 2, customised for the paper’s recurring example system of cylinders under uniform bending. The capacity curve upon which RRD is based (Fig. 1) is formulated in terms of the resistances \( R_\text{cr} \) and \( R_\text{pl} \), which are linear in the elastic modulus \( E \) and the yield stress \( \sigma_y \), respectively. It may be argued that these can be kept constant at reasonable values for the entirety of the characterisation and constitute the base state, or Level 0. Changes to the structural form or the loading distribution constitute changes to the system and require a start from a different Level 0.

Recent research has begun to acknowledge and incorporate the beneficial effect of strain hardening on steel design through simple piecewise-linear characterisations of the stress-strain relationship [31–34]. Without strain hardening, a system can never attain the full plastic condition at a finite slenderness, unrealistically forcing \( \lambda_0 \) to be zero and \( \chi_y \) to be unity in any RRD characterisation (Fig. 1). The innately slender nature of shells means that their behaviour is governed mainly by stability considerations, and plastic strains are unlikely to go far beyond the ‘yield plateau’ portion of the stress-strain curve. The simplest and most conservative material law is thus a bilinear relationship with an initial elastic region of modulus \( E \) followed by linear strain hardening region of modulus \( E_y \) (expressed as a percentage of \( E \)) which should form the Level 1 variable. It may be set to zero for an ideal elastic-plastic characterisation, though a recent study of 225 stress-strain curves by Sadowski et al. [34] found that the ‘yield plateau’ actually exhibits a statistically significant positive modulus \( E_y \) of the order of 0.1% of \( E \), a more accurate but still conservative lower bound on \( E_y \) that supports realistic values of \( \lambda_0 \) and \( \chi_y \). The Level 2 variable should relate to the relative amplitude \( \delta \) of the imperfection form, carefully chosen to ensure that it is relevant and appropriately deleterious to the shell system within the constraints of practical constructions [13,14,22].

The suggested analysis hierarchy permits Levels 1 and 2 to be universal for all metal shell systems, since every shell is affected by plasticity and imperfections. By contrast, Levels 3 and above govern a system’s dependency on geometric nonlinearity which is controlled by variables unique to its geometry such as the length, width, radius and thickness (or their ratio \( r/t \) for compactness) etc., whose Levels may be arranged in any logical order. While the hierarchy may be followed methodically to perform a vast number of GMNIAs and obtain all RRD parameters by brute force, it is perhaps better used to design sub-studies to obtain individual parameters in isolation, simplifying the overall task. For example, characterising \( R_\text{cr} \) and \( R_\text{pl} \) requires only LBAs and MNAs to be performed at Level 3 and automatically ‘assume’ that the branch corresponding to \( E_y = 0 \) and \( \delta = 0 \) has been chosen at Levels 1 and 2 respectively. Obtaining \( \alpha_\text{cr} \) requires Level 3 GNAs under the same variable choices but using large displacement theory, while \( \alpha_\text{cr} \) requires Level 2 GNIs. Different imperfection forms may be investigated, each resulting in a different \( \alpha_\text{cr} \). Since LBAs, GNAs and GNIs are by definition elastic with plasticity ‘deactivated’, and since MNAs should not include strain hardening [16,17], Level 1 is redundant for these sub-studies. However, the remaining plasticity-related RRD parameters \( \beta, \eta, \lambda_0 \) and \( \chi_y \) must be established using GMNIAs performed at Level 1, though potentially on the basis of fewer analyses across smaller parameter ranges if an understanding of other aspects of the system has been developed. The hierarchy formally establishes that any RRD parameter at any Level is a function only of the physical variables at the same Level or above. For ex-
Fig. 2. Analysis hierarchy to establish RRD parameters from system variables, specialised for cylinders under global bending.
analytical expression for the classical elastic critical buckling moment is calculated using classical beam theory as:

$$M_{cl} = W_{cl} \cdot \sigma_{cl}$$  \hspace{1cm} (2)

where $W_{cl} = \pi r^2 t$ is the thin-walled elastic section modulus, $\sigma_{cl} = 0.605Et/r$ is the classical elastic critical buckling stress, $E$ is the Young's modulus, and $r$ and $t$ are the radius and thickness respectively. However, this simple expression neglects the edge boundary conditions in short cylinders which restrain local buckling in a linear bifurcation analysis, an effect that can only be captured if the full pre-buckling stress state is included in the analysis and a three-dimensional buckle is permitted to develop. Rotter et al. [30] performed hundreds of Level 3 LBAs to show computationally that this boundary effect may be accommodated by a simple adjustment to $M_{cl}$, giving the following more complete expression for $R_{cl}$:

$$R_{cl} \equiv M_{cl} = \left( 1 + \frac{4}{\omega^2} \right) M_{cl}$$  \hspace{1cm} (3)

where $\omega = l_i/\sqrt{(rt)}$ is a carefully-devised dimensionless length that enables $R_{cl}$ to be characterised independently of the $r/t$ ratio. The significance of the $\omega$ parameter is explored in greater detail in Section 7.

## 5. Automated model generation and input file logistics

Advanced modern finite element packages often interface with a scripting language to enhance functionality and aid in the automation of parametric studies (e.g. Python in ABAQUS, APDL in ANSYS, Matlab in COMSOL), an invaluable tool for establishing databases of RR parameters. The analysis hierarchy shown in Fig. 2 lends itself directly to a computational implementation as a system of nested loops, with Level 1 forming the outer loop and each higher Level forming an inner loop. However, this naïve approach may quickly lead to a truly unmanageable number of calculations, as each variation of a low-Level variable spawns an entire branch of potentially thousands of additional GMNIA. Prudent low-Level choices may allow substantial economies in computational effort with minimal loss of information on RR parameter dependencies.

It is advantageous to take Level 3 as the outer loop within a master model generating script, with all higher Levels of the system forming the inner loops. In this way, the same master script may be used with only minor adjustments to automate the generation of LBAs, MNAs and GNAs (to individually establish $K_n$, $K_{pl}$ and $\alpha_C$) and, with specialisations for (a potentially minimised number of) low-Level variable choices, also GNAs (for $\alpha_l$) and GNIAs (for $\beta$, $\eta$, $\lambda_0$ and $\chi_0$). An outline of a generating script writ-
ten in Python illustrating how the generation of .inp input files (which contain the entirety of the model information that is submitted to the ABAQUS/Standard FE solver for analysis) may be automated is shown in Algorithm 1. When 'executed' using the example LVL3 vector for cylinders under uniform bending, the code will create 199 directories named L3_10 to L3_1000 (i.e. for \( r/t = 10 \) to 1000 in steps of 5), each of which will contain as many .inp files for that Level 3 variable (i.e. \( r/t \) ratio) as requested in the LVL4 vector (i.e. length 1). These will all be of the same analysis type (e.g. LBA, MNA, GNA, GNIAX, GMNIA) as specified in Analysis.

The script additionally includes commands to aid in the automated submission of the large number of generated .inp files. For a processor with 8 logical cores (4 physical cores at 2 threads per core), for example, setting Batches = 4 requests the creation of four new Python scripts within each Level 3 folder called L4_runme_X.py where X is 0 up to 3, each one with content similar to that shown in Fig 3. These may be submitted for analysis in parallel on four separate command consoles (via python runme_X.py), and each Level 4 .inp file inside its Level 3 folder will be analysed using 2 processor cores (due to CPUs = 2). The commands additionally specify that a FORTRAN user subroutine named MN_Riks_killer.f or GN_Riks_killer.f be invoked. The vital purpose of these is explained in detail in Section 6.

For cylinders under uniform bending, the Level 3 \( r/t \) ratio directly controls the resolution of points along the capacity curve (see Chen et al. [27]) and thus the base data set upon which RRD parameters can be accurately established. In contrast with Levels 1 and 2, the Level 3 variable is thus likely to cover a substantial range for any shell system leading to a long LVL3 vector and many Level 3 folders, so it may be convenient to gather all sets of individual Level 4 submissions scripts into a single set at Level 3. The complete Python code given in Algorithm 2 shows how a L3_runme_X.py file can conveniently be built out of any number of L4_runme_X.py files, an example of which is shown in Fig. 4. This code additionally suggests useful housekeeping operations, such as deleting unnecessary files once analysis has been completed to preserve disk space. In particular, the capacious ABAQUS output database .odb files (containing the entirety of the computed nodal and element variables) are no longer required once an analysis has terminated, as the buckling loads may be obtained from the much smaller .dat and .sta text files that are created simultaneously.

Algorithm 2 Complete Python code for the creation of Level 3 submission scripts.

```python
import os
command = "abaqus job=(jobname for LVL4(0)) (user=XXX) cpus=2 int"
os.system(command)
command = "abaqus job=(jobname for LVL4(1)) (user=XXX) cpus=2 int"
os.system(command)
command = "abaqus job=(jobname for LVL4(2)) (user=XXX) cpus=2 int"
os.system(command)
command = "abaqus job=(jobname for LVL4(3)) (user=XXX) cpus=2 int"

# etc. until all lengths in LVL4 have been included. This file is L4_runme_0.py. Commands for LVL4[2-4*n] will be in 'L4_runme_2.py', those for LVL4[4+4*n] will be in 'L4_runme_4.py'. To be invoked from command prompt as 'python runme_0.py'. The 'user' command should only be present for 'files' analyses, not LBA.
```
6. Automated termination of analyses

6.1. Introduction

Automation would be of limited use if it were not possible to stop an analysis once a 'criterion of failure' has been met and proceed to the next one. While modern FE solvers are now able to follow nonlinear equilibrium paths deep into the post-buckling domain [35,36], it is invariably only the first critical point that is of interest for design by global numerical analysis [13] and RRD. Unfortunately, FE software often fail to provide a satisfactory facility for automatically terminating a nonlinear analysis at the first critical point. For shell systems, this is exacerbated by their complexity and the many conditions under which 'failure' may be deemed to occur [17]. This section presents a strategy for the automated detection of all relevant 'criteria of failure' for shells and illustrates its implementation in ABAQUS. It is hoped that eventually these will become standard facilities in all structural FE software with a nonlinear solver.

6.2. Automated termination of linear bifurcation analyses (LBAs) and materially nonlinear analyses (MNAs)

The capacity curve (Fig. 1) is defined relative to the reference resistances $R_\eta$ and $R_\phi$ obtained from LBA and MNA computations respectively. As all RRD parameters are defined relative to these two reference results, it is important that they be established as accurately as possible. An LBA is a matrix eigenvalue calculation (*BUCKLE step in ABAQUS) which automatically terminates once the desired number of eigenvalues have been extracted. Assuming that a robust meshing scheme is applied, LBAs require no further monitoring. An MNA, by contrast, is a small displacement analysis to determine the plastic limit state corresponding to a plastic collapse mechanism, manifest as a 'plateau' on the equilibrium path (Fig. 5). This requires an incremental path-tracing analysis, most robustly performed in ABAQUS as a *STATIC step using the 'Riks' [37] modified arc-length algorithm with geometric nonlinearity deactivated and an ideal elastic rigid-plastic material law with no strain hardening.

Doerich and Rotter [38] give a detailed account of the challenges involved in performing MNAs and accurately determining $R_\phi$. For shell systems dominated by membrane yielding, the equilibrium path attains an obvious horizontal plateau relatively quickly, with only a modest number of analysis increments. However, for systems with localised plastic collapse mechanisms involving extensive local bending and surface yielding, the plateau may not be reachable within a tolerable number of increments (Fig. 5). Doerich and Rotter [38] propose a method to accurately predict the load proportionality factor (LPF) at which the 'plateau' may eventually form (and thus obtain $R_\phi$) based on a partially-computed portion of a still-rising equilibrium curve of LPF vs {$\phi$}, the tracked degree of freedom (Fig. 6a). The equilibrium plot is first transformed into a Modified Southwell (MS) plot of LPF vs [LPF/DOF] (Fig. 6b), which consists of an initial vertical portion corresponding to the linear elastic response and a plateauing portion corresponding to the plastic response. The projection of the tip of the non-vertical portion of the curve on the vertical axis (i.e. a linear fit through the three most recent adjacent data points) gives a first estimate of $R_\phi$, called $P_{MS}$. This is then used in a second transformation to construct a Convergence Indicator Plot (CIP) of $P_{MS}$ vs $m=\left(\frac{P_{MS}}{LPF}\right)$ (Fig. 6c). The vertical axis intercept of a linear least-squares fit through the CIP data gives an enhanced estimate of $R_\phi$ based on only a modest number of increments, as illustrated in Doerich and Rotter [38] on several shells systems. The equilibrium path, MS plot and CIP are not unique for any system and the choice of tracked DOF requires a careful assessment prior to automation.

The CIP projection method was arguably conceived for application during post-processing to make the most of an MNA that was allowed to run for as long as was deemed tolerable and then terminated manually. Although an experienced user may become a good judge of how many increments are optimal for a good $R_\phi$ estimate together with the CIP, this is hardly a satisfactory strategy for automation. This paper offers a FORTRAN implementation of an 'on the fly' CIP that can be constructed during an ongoing MNA and automatically terminate it once a Kill Condition (KC) is satisfied. It exploits the extended functionality of user subroutines supported by ABAQUS, compiled during pre-processing, a powerful tool although it requires the proprietary Intel® FORTRAN compiler [39]. This code is contained in a user subroutine referred to in Algorithm 1 as MN-Riks.rcell1.f and presented in full as Subroutine 1 in Appendix A, based on example code given in Section 11.49 of the ABAQUS User Subroutines Reference Guide for version 6.14-2 [11]. The code reads the current LPF and tracked DOF values at each increment (for which a ‘Riks’ analysis specifically is required) and proceeds through each of the stages shown in Fig. 6, though it only builds the CIP once a significant deviation from verticality is detected on the MS plot (Coefficient of Variation...
of the horizontal axis data greater than, say, 10%. A stop command is sent to the ABAQUS/Standard solver once the KC is satisfied that the absolute relative difference between any two adjacent CIP projections of $R_{pl}$ is less than a specified tolerance (Fig. 6d), say, 0.01%.

6.3. Automated termination of geometrically nonlinear analyses (GNAs, GNIA s, GMNIAs etc.)

Shells are widely known to exhibit unstable buckling behaviour characterised by an abruptly descending and highly nonlinear post-buckling equilibrium path. These may be computed by powerful arc-length ‘path tracing’ methods, of which the most robust implementation in ABAQUS is again the modified ‘Riks’ algorithm [37]. The ‘Riks’ procedure has been shown to be reliable in following geometrically nonlinear equilibrium paths deep into the post-buckling range for various applications [21,24,35,36,40,41].

In its detailed description of rules governing ‘design by global numerical analysis using GMNIAs’ which attempt to cover all possible metal shell systems, the EN 1993-1-6 [13] Standard defines four ‘criteria’ that represent a state of failure (Fig. 7). Criteria C1 and C2 represent the most common manifestations of unstable post-buckling, a reversal in the direction of the equilibrium path either due to a limit point (Criterion C1; global snap-through buckling) or a bifurcation (Criterion C2a; local buckling). Criteria C3 and C4 represent more ambiguous cases where there is no drop in the LPF and the equilibrium path continues to rise indefinitely, although there may be an intermediate ‘kink’ where the curve abruptly changes slope. For want of a reliable way of detecting the ‘kink’, the conservative rule was written to terminate the GMNIA either once a ‘largest tolerable deformation’ is reached (Criterion C3) or the highest equivalent stress in the system reaches first yield (Criterion C4). Criterion C3 is already possible in FE software which allow a DOF to be tracked and terminate the analysis once the DOF reaches a pre-defined value (which EN 1993-1-6 would supply), while the very conservative Criterion C4 is a poor basis upon which to establish accurate RRD parameters. The remainder of this section thus focuses on the automation of Criteria C1 and C2, as well as on the automated detection of ‘kinks’ on the equilibrium curve.

The occurrence of unstable post-buckling is manifest by a decrease in the load proportionality factor (LPF), which occurs for both limit points and bifurcations on the equilibrium path (Fig. 7). Unfortunately, there is currently no way through either the ABAQUS CAE interface or the .inp file to automatically terminate a ‘Riks’ analysis when the LPF first begins to decrease, and to the authors’ understanding this facility is also missing by default from other FE software. For shells that exhibit limit point ‘snap-through’ buckling, a maximum number of increments before termination may be specified based on prior experience. However, this approach is not reliable for bifurcations since the LPF incre-
ment may be greatly cut back by the solver in the vicinity of a bifurcation causing the analysis to proceed very slowly, exhausting the permitted number of increments without capturing the bifurcation. While this too is potentially countered by setting a high maximum number of increments, it is a very inefficient solution with no guarantee of success.

The automatic termination of an on-going geometrically nonlinear ‘Riks’ analysis is a crucial capability that is likely to find application far beyond the present context of RDD. To this end, this paper offers a second FORTRAN user subroutine for ABAQUS that implements Criteria C1 and C2 above in addition to a ‘kink detector’. It is referred to in Algorithm 1 as GN_Riks_killer.f and presented in full as Subroutine 2 in Appendix A. Usable with any geometrically nonlinear ‘Riks’ analysis (GNA, GNIA or GMNIA), the code sends a stop command to the solver when the ongoing analysis is deemed to satisfy any one of the following three KCs:

- **KC1**: current increment’s LPF is less than the previous increment’s LPF. This captures both limit point and bifurcation instabilities (Criteria C1 and C2), though for the latter only if the solver correctly switches path (Fig. 7, ‘a’ curve for Criterion C2).
- **KC2**: the change in LPF from the first increment to the next is smaller than a specified tolerance. The authors have found this to be a useful mechanism to prevent the analysis from becoming stuck in an unending loop of cutbacks and increases in the arc-length increment size due to repeated attempts at convergence in the vicinity of a numerically ‘problematic’ bifurcation.
- **KC3**: the absolute relative difference between the current and previous incremental radii of curvature of the equilibrium curve exceeds a specified tolerance.

KC3, the ‘kink detector’, is perhaps the most contentious. It is known that some imperfect shell systems do not necessarily exhibit a monotonically decreasing relationship between the buckling resistance and imperfection amplitude, and a bifurcation point with a descending equilibrium path may, for deeper imperfections, turn into a smooth inflection point followed by an ascending path [42]. For example, Fig. 8a shows equilibrium paths in terms of the LPF against the end rotation and peak inward radial deformation DOFs for a very short cylinder under uniform bending with a ‘Type A’ Rotter and Teng [43] weld depression imperfection with a deep amplitude at midspan (from Fajuyitan et al. [44]). Fig. 8b shows equilibrium paths of the LPF against the end shortening and peak inward radial deformation DOFs for a stepwise varying wall thickness cylindrical silo under eccentric discharge with regularly-spaced weld depression imperfections (the ‘80%’ curve from Figure 13 in Sadowski and Rotter [41]). Both of these systems exhibit unstable bifurcation buckling at lower imperfection amplitudes where the criterion of failure is obvious and easily detectable by KC1, but at higher amplitudes the bifurcations vanish (with no reported negative eigenvalues in the global tangent stiffness matrix) and there is instead a continuous transition from a pre- to post-buckled state but with a clear ‘kink’ on the equilibrium path. The deformed shapes both before and after the ‘kink’ are illustrated in Fig. 9, and clearly show the development of localized buckling deformations. Detecting the ‘kink’ has hitherto required careful manual and visual study of the equilibrium path, and it is easily missed if it is not anticipated.

Any implementation of KC3 is complicated by the fact that the equilibrium path is not unique and depends on the tracked DOF, whereas KC1 and KC2 are assessed on the basis of the LPF alone. Tracking a different DOF may cause the smooth transition from pre- to post-buckling appear as a stronger or weaker ‘kink’ on the equilibrium path (compare left and middle columns in Fig. 8). A very simple ‘kink detector’ would require only the calculation of the second derivative (curvature) of the equilibrium path using a finite difference scheme, coupled with a criterion that tests for a change in the sign of the curvature at any given analysis increment. The change in sign then corresponds to an inflection point, i.e. a change from a softening to a hardening equilibrium path or vice-versa, and is consistent with the criterion used to determine the buckling load of imperfect shells by Yamaki [42]. However, the authors believe that in the most general case, a ‘kink’ may in fact occur without an inflection point where, for example, a softening path abruptly transitions to another softening path with the tangent stiffness remaining positive and no drop in the LPF. In such a case, a simple ‘change in sign’ criterion for the curvature would not be sufficient to detect the ‘kink’.

The proposed simple implementation of a ‘kink detector’ computes the local radius of curvature at any increment by fitting a circle to the three most-recent points on the LPF vs DOF curve. In a geometrically nonlinear analysis, even the initial ‘linear’ portion of the equilibrium curve is never perfectly linear thus the radius of curvature is always finite, although use of double precision arithmetic is advised nonetheless. The proposed detector sends a stop command to the solver if the absolute relative percentage change of two adjacent radii of curvature is found to exceed a tolerance, say 80%. The choice of tolerance should be tailored to the system and the tracked DOF, and should be low enough to be sensitive to the ‘kink’ but high enough to avoid false positives. This detector makes no assumption about the sign of the curvature, it simply tests for disproportionate changes in it from one increment to another. However, a more sensitive detector could track incremental changes in higher derivatives computed using backward finite difference operators, as in some cases the ‘kink’ may be too smooth for detection using the curvature alone. In particular, automated termination poses a particular challenge for systems with a naturally stable post-buckling response, such as a rectangular plate under uniform axial load as shown in Fig. 8c (Benchmark #3 in Sadowski and Rotter [45]) where the transition from pre- to post buckling is often too gradual to qualify as a ‘kink’.

A final issue deserves mention. During a geometrically nonlinear analysis of a perfect shell system, the FE solver may at times fail to switch to the secondary path at a bifurcation and instead continue erroneously on the fundamental path with an increasing LPF (Fig. 7; b curve for Criterion C2). However, the tangent stiff-
ness matrix will cease to be positive-definite and the software may report negative eigenvalues beyond this point, hinting at the presence of a lower-energy state on an ignored secondary path [46]. This problem is traditionally countered by introducing a very minor mesh perturbation in the form of the critical buckling eigenmode (i.e. 1st LBA mode [24,25,47,48]) which turns the bifurcation into a limit point and ensures a continuous response. Buckling then becomes easily detectable by KC1. However, this procedure does not always reliably ensure a switch at the lowest bifurcation, and certain systems may require such a deep perturbation that the initial tangent stiffness is affected. If the first bifurcation is missed, the analysis risks dangerously overpredicting the buckling resistance. The authors suggest that a further ‘KC4’ should be implemented which terminates the analysis once the tangent stiffness matrix stops being positive-definite and negative eigenvalues are detected for the first time, even if no switch in the load path occurs. However, this does not appear to be currently possible within ABAQUS or other software.

7. Automated processing of output

7.1. Identification of dimensionless groups governing geometric nonlinearity

Rotter [2,16] emphasises the importance of isolating the effect of geometric nonlinearity ($\alpha_C$) for any RRD characterisation, as this permits complete capacity curves to be established for ‘constant geometric nonlinearity’ where $\alpha_C$ is manifest as a straight vertical line in $R_c / R_{pl}$ vs $R_c / R_{pl}$ space (Fig. 1b). The analysis hierarchy (Fig. 2) shows that the effect of geometric nonlinearity may be isolated through a sub-study of GNAs performed at Level 3 and above, leading to a characterisation of $\alpha_C$ that is a function of geometric variables only (e.g. $r/t$ and $L$). In a number of shell systems investigated so far, a wise application of background theory has enabled the construction of a single dimensionless group, effectively a ‘multi-Level’ variable, that captures the entirety of $\alpha_C$’s dependence on the shell geometry. Examples include:

- Cylinders under axial compression [42,49]: both $\omega = L/\sqrt{r/t}$ and the Batdorf [50] parameter $Z = (L^2/(r/t))(1-\nu^2)$ have been used as dimensionless variables to characterise this system.
- Cylinders under bending [30,51]: $\omega = L/\sqrt{r/t}$ captures the local ‘short-wave’ buckling of shorter cylinders, while $\Omega = (L/r)/(t/r)$ captures the global ‘long-wave’ cross-sectional ovalisation of longer cylinders.
- Cylinders under wind loading [52]: $\xi = (L/r)(t/r)^{1/2}$ has been established computationally as the governing dimensionless geometric group.

It appears that, with the exception of $Z$, the above are all special cases of a general ‘Level 3/4’ dimensionless variable that controls...
Algorithm 3 Object classes in pseudocode for automated capacity curve processing.

Class Capacity_Curve(object):
  Method_1 to initialise instance of object representing a capacity curve. The curve
  is defined by a vector of the Level 3 geometric variable and a vector
  of GMNIA load factors, for constant Level 0, 1, 2 and 3/4 variable values.
  Method_2 to build ‘modified’ capacity curve (see Fig. 1b) of Rk/Rρ using computed GMNIA, LBA and MNA load factors for constant Level 0, 1, 2 and 3/4, and deduce accurate initial estimates for RRD parameters.
  Method_3 to build ‘traditional’ capacity curve (see Fig. 1a) of Rk/Rρ using α using computed GMNIA values for constant Level 0, 1, 2 and 3/4, together with estimates from Method 2, perform a constrained least-squares fit using the functional form of Eq. 1 to obtain ‘best’ RRD parameters. Save each fitted RRD parameter as a property of the Capacity_Curve object.

Class Level_3(object):
  Method_1 to initialise instance of object representing the contents of a Level 3 folder. This will contain vectors of GMNIA Rk, load factors paired with the corresponding Level 4 variable value. The load factors are read in from .sta ‘Riks’ analysis files.
  Method_2 to build a vector of the Level 3/4 variable.

Main Programme:
Specify Level 0, 1 and 2 variable values for identification.
Specify ranges of Level 3 and 4 variable values as used in FE analyses.
Establish the desired working range of the level 3/4 variable.

For each value of the Level 3 variable:
  Call Level_3.Method_1 to create a Level 3 object for each level 3 folder.
  Call Level_3.Method_2 to build vector of the Level 3/4 variable.
  Store each Level_3 object in an array.

For each value of the desired level 3/4 variable:
  Search all Level_3 objects for GMNIA Rk, load factors corresponding to the
desired Level 3/4 variable value and add to an array e.g. this results in an
array of GMNIA Rk/Rρ vs. varying r/t for constant ζ.
  Call Capacity_Curve.Method_1 to initialise an instance of the object using
the above data, and add it to an array of these objects for storage.
  Call Capacity_Curve.Method_2 and .Method_3 and save fitted RRD parameters.
  Plot the resulting RRD parameters as function of the Level 3/4 variable.

αC for uniform cylindrical geometries:
\[
ζ_a = \frac{L}{r} \left( \frac{r}{L} \right)^a
\]
where \(a = 1/2\) gives \(ζ_a\), \(a = -1/2\) gives \(Ω\), \(a = 4/7\) gives \(ε\), while
\(a = 0\) or \(1\) give simply \(L/r\) or \(L/t\) respectively. It is suggested that \(αC\)
for many cylindrical shell systems may be expressed compactly in
terms of \(ζ_a\) with an appropriately specialised value of \(a\), and
general forms akin to Eq. 4 are likely to exist for other shell geometries.
Establishing the governing dimensionless geometric group is
an important investment, as it reduces by one the number of variable
dependencies on all RRD parameters. For example, Level 1
RRD parameters \(β, η, λ_0\) and \(χ_β\) then become a function of only
three physical variables (i.e. \(E_0, δ/t\) and \(ζ_a\) - directly representing
material nonlinearity, imperfections and geometric nonlinearity),
rather than four (i.e. \(E_0, δ/t, r/t\) and \(L\)).

7.2. Extracting capacity curves at ‘constant geometric nonlinearity’

The outline of a script in pseudocode to process the data generated
by a programme of GMNIAs performed according to the analysis
hierarchy in Fig. 2 is presented in Algorithm 3. In keeping with
the preceding discussion and Algorithm 1, it is advantageous to
perform the processing at Level 3 to avoid the complexity of many
nested loops. The code to read buckling load factors from text
data files generated by the FE software is easily written and benefits
from an ordered arrangement of Level 4 model files within Level 3
folders as suggested by Algorithm 1 and 2. In ABAQUIS, \(R_0\) may be
obtained as the lowest positive eigenvalue read from a .dat file,
while \(R_ρ\) and \(R_k\) may be obtained as the highest LPF read from a
.sta file created by a ‘Riks’ analysis.

Algorithm 3 suggests making use of the powerful data abstractions
available in an object-oriented language such as Python or Matlab [53]
to organise the significant amount of information

generated during a programme of GMNIAs. The contents of each
Level 3 directory of completed GMNIAs (e.g. for constant r/t) can
be stored in a custom-written Level_3 object. These consist of
equal-length lists of \(R_0\) load factors, the corresponding Level 4
variable value (e.g. varying \(L\)) and the constructed Level 3/4 variable
value (e.g. varying \(ζ_a\)). Continuing the illustration in Algorithm 1,
for example, there would be 199 instances of the Level_3 class
associated with the branch choices made at Levels 0 to 2.

The capacity curve (Fig. 1) is a conceptual abstraction that lends itself
particularly well for implementation as a data object. Since
these should be established for ‘constant geometric nonlinearity’
[26], an instance of the Capacity_Curve class suggested in
Algorithm 3 should be created at each desired value of the con-
structed Level 3/4 \(ζ_a\) variable which controls \(αC\). To gather the
data necessary to create a Capacity_Curve object at a partic-
ular Level 3/4 variable value, it is sufficient to search all Level_3
objects within the same branch of Levels 0 to 2 for GMNIA \(R_0\)
load factors at that Level 3/4 value. This gives sufficient information
to construct both a ‘traditional’ (Fig. 1a) and ‘modified’ (Fig. 1b)
capacity curve, which may then be used to perform a constrained
least-squares fit using Eq. 1 as the functional form and thus estab-
lish a set of RRD parameters specific to those Level 0, 1, 2 and
3/4 variable values. In this light, it is advantageous to design the
LVL4 vector in Algorithm 1 to always generate the same vector of
desired Level 3/4 variable values for any constant Level 3 variable
value, as illustrated in the final section.

7.3. Illustration on perfect cylinders under uniform bending

The logic of Algorithm 3 was implemented in Matlab [53] and is
used here to illustrate how RRD parameters may be established
for perfect cylindrical shells under uniform bending. Rotter et al.
[30] found that separate phenomena govern the behaviour at dif-

25
A.J. Sadowski et al./Advances in Engineering Software 109 (2017) 15–30
Fig. 10. Illustration of Algorithm 3 where the extraction of ‘modified’ capacity curves is performed at constant Level 3/4 parameter \( \Omega = (L/r) \sqrt{t/r} = \zeta_{1/2} \).

Fig. 11. Extracted RRD parameters for ‘transitional-length’ perfect cylinders under global bending with minimal and without strain hardening governed by Level 3/4 parameter \( \Omega = (L/r) \sqrt{t/r} = \zeta_{1/2} \).

different length ranges with \( \alpha_c \) characterised by two separate Level 3/4 variables, \( \omega = \zeta_{1/2} \) and \( \Omega = \zeta_{1/2} \). For brevity, only the behaviour dominated by pre-buckling cross-sectional ovalisation controlled by \( \Omega \geq 0.5 \) is illustrated.

Fig. 10a shows an excerpt from approximately three thousand GMNIA \( R_k \) load factors arranged in a three-dimensional \( R_k/R_{pl} \) vs \( R_k \) vs \( \Omega \) space, where the resistance \( R \) is the applied end bending moment \( M \). These were generated automatically and computed over only 48 hours on a single unsupervised workstation powered by a modest Intel® Core™ i7-6700 processor (3.4 GHz clock speed) with 16 GB of RAM assuming Batches = 4, CPUs = 2 (Algorithm 1) and automated analysis termination using GN_Riks_killer.f. For the range \( \Omega \geq 0.5 \) accurate analytical results exist for \( R_{pl} \) and \( R_{pl} \) [27] thus they did not need to be obtained using Level 3 LBAs or MNAs. The Level 4 vector of lengths was designed to maintain steps of 0.5 in \( \Omega \). The coloured curves on Fig. 10a parallel to the \( \Omega \) axis are saved by Matlab in memory as Level_3 objects. The solid black curves in the planes perpendicular to the \( \Omega \) axis represent ‘modified’ capacity curves (Fig. 1b) saved in memory as Capacity_Curve objects (shown in detail on Fig. 10b for \( \Omega = 0.5, 1.0, 1.5, 2.0, 2.5 \) and 3.0).

A typical plot of the dependency of the RRD parameters on the Level 3/4 \( \Omega \) variable in the range \( 0.5 \leq \Omega \leq 7 \) (extracted from 2 \( \times \) 22 capacity curves) is shown in Fig. 11 for a perfect cylinder and two different initial strain hardening moduli; \( E_t = 0 \) (ideal elastic-plastic) and 0.1% of \( E = 200 \) GPa (see Sadowski et al. [34]). An enhanced formulation of the Level 1 RRD parameter \( \eta \) was used [20] which assumes a linear variation with \( \lambda \) between \( \eta_0 \) at \( \lambda_0 \) and...
The extracted $\alpha_G$ parameter follows the relationship with $\Omega$ [30,51] showing that ‘Brazier’ ovalisation is fully-developed for $\Omega > 3.5$ and responsible for an approximate 50% reduction in the nonlinear elastic buckling resistance. It is entirely unaffected by $E_p$.

The $\beta$ parameter, representing the first visible onset of the effects of plasticity, is only very slightly affected by $E_p$. As these illustrations have been performed for a perfect shell only, there is as of yet no information to extract $\alpha_G$ which would require a full range of Level 2 branch choices. For the case of no post-yield strain hardening ($E_p = 0$), the $\chi_b$ and $\lambda_0$ RRD parameters were constrained to be fixed at unity and zero respectively. This has a significant effect on the shape of the curves for low $\lambda$, which exhibit very different $\eta_b$ and $\lambda_0$ values depending on whether strain hardening is present or not. A complete RRD characterisation of cylinders under uniform and various non-uniform bending moment distributions is currently being carried out using the methodology presented in this paper, and will be published at a later stage.

8. Conclusions

Reference Resistance Design (RRD) is a method of analysis recently devised by Rotter [1] and approved for implementation in the Eurocode on the design of metal shells EN 1993-1-6. It permits a designer to take full advantage of the accuracy and complexity of an advanced nonlinear FE analysis of a metal shell without bearing the significant expense of having to perform it. The information about the behaviour of any system is encoded within six physical algebraic parameters and two physical resistances built around a base ‘capacity curve’ functional form. Establishing complete databases of RRD parameters for the most important shells systems is a significant challenge in modern shell buckling research.

This paper has presented a computational strategy to aid researchers in designing a programme of parametric FE analyses to establish complete RRD characterisations of metal shell structural configurations and load cases. The research community is particularly urged to embrace the possibilities for mass automation already present in modern FE software to handle the many thousands of fully nonlinear calculations necessary to establish RRD characterisations. A strategy for automated generation, submission, termination and processing of analyses is presented, built around a novel formal analysis hierarchy. Annotated Python code fragments and object-oriented pseudocode are presented to illustrate the concepts, whose logic may be specialised for any shell system and in any programming language.

A crucial capability is the automated termination of an ongoing FE analysis once a ‘criterion of failure’ has been met. This paper explores the many different failure criteria possible for metal shells depending on the analysis type, and proposes computational procedures for the automated detection of each one.

Fully-functional implementations written in FORTRAN are presented for immediate use as ABAQUS user subroutines, but the logic may be implemented in any FE software with a scripting functionality. Makers of FE software may wish to carefully consider incorporating some of the proposed functionality into their products, as the applications extend far beyond RRD and shell buckling.

Acknowledgements

This work was partly funded by the UK Engineering and Physical Sciences Research Council (EPSRC) with grant contract EP/N024060/1. Mr Fajuyitan’s contribution was funded by the Petroleum Technology Development Fund (PTDF) of Nigeria. The authors are very grateful to Professor J. Michael Rotter for his valuable comments on the draft manuscript.

Appendix A. FORTRAN code

Below are two FORTRAN user subroutines URDFIL to enable the automatic termination of a ‘Riks’ analysis, adapted from Section 11.49 of the ABAQUS User Subroutines Reference Guide for v. 6.14–2 [11]. This additionally requires the following lines to be present in the submitted .inp file, which assumes that a set called RIKS_NODE has been defined containing a single ‘tracker’ node.

```fortran
*NODE FILE, GLOBAL=NO, NEWT="RIKS_NODE"

if (a different coordinate system has been chosen, e.g. cylindrical, set GLOBAL=NO and apply the "TRANSFORM keyword previously. The DOF that is to be used as the horizontal axis of the equilibrium path is to be specified within the subroutine itself.

Subroutine 1 is intended for the automated termination of a materially nonlinear (but geometrically linear) MNA when a projection of the plastic collapse load is assumed to be sufficiently accurate on the basis of Convergence Indicator Plot (CIP) of Doerich and Rotter [38], invoked in Algorithm 1 as MN_Riks_killer.f. The CIP projection only initiates once the Modified Southwell (MS) plot is deemed to have sufficiently deviated from a vertical line (Fig. 6b). The kill condition (KC) is when the absolute relative difference between two adjacent CIP estimates of $R_{pl}$ is less than a tolerance. Subroutine 2 is intended for the automated termination of a geometrically nonlinear (possibly with material nonlinearities and/or imperfections) GNA, GNIA or GMNIA when one of three KCs is met, invoked in Algorithm 1 as GN_Riks_killer.f. The subroutines are available for download with extended commentary and functionality on the authors’ ResearchGate profiles where they are periodically updated.
Subroutine 1. ABAQUS user subroutine MN_Riks_killer.f for the automated termination of materially nonlinear ‘Riks’ MNAs based on the Convergence Indicator Plot (CIP) of Doerich and Rotter [38].
SUBROUTINE URDFILL(LSTOP, LOVRWRT, KSTEP, KINC, DTIME, TIME)
  C ABAQUS User subroutine to automate GN Riks analysis termination based on many failure criteria.
  C Invoked as 'GN Riks killer.f'. Below are preliminaries inherited from the example code.
  INCLUDE 'ABA_PARAM.INC'
  DIMENSION ARRAY(513), JARRAY(NPREDC, 513), TIME(2)
  EQUVALENC (ARRAY(1), JARRAY(1, 1))
  C
  ALL XXX are arrays (default size 100) to store LPF, DOF and RCURV (radius of curvature) values
  C
  CHANGE RCURV is the absolute relative change in the incremental radius of curvature
  C
  TOL INC LPF is the tolerance on the incremental CURR LPF - PREV LPF (here 1e-7)
  C
  TOL CHANGE RCURV is the tolerance on CHANGE RCURV (here 100)
  C
  DOUBLE PRECISION, DIMENSION (100) :: ALL_LPFI, ALL_DOF, ALL_RCURV
  DOUBLE PRECISION CHANGE_RCURV, S1, S2, S3, A1, A2, PRODUCT_TRIANGLE_SIDES, AREA_TRIANGLE
  DATA ALL_LPFI/100.0.D0/, ALL_DOF/100.0.D0/, ALL_RCURV/100.0.D0/
  PARAMETER (TOL_INC_LPFI=0.1D-7, TOL_CHANGE_RCURV=1.0D0)
  C
  DO K = 1, 9999999
    CALL DBFILE(0, ARRAY, JRCD)
    IF (JRCD.NE.0) GO TO 110
    KEY = JARRAY(1, 2)
    IF (KEY.EQ.2000) THEN
      ALL_LPFI(KINC) = ARRAY[-]
    END IF
    DDOFs are stored in Record Key 101 attributes 2-7 + offset 2 = array positions 4-9
    C
    ELSE IF (KEY.EQ.101) THEN
      ALL_DOF(KINC) = ABS(ARRAY(XX))
    END IF
    C
    The radius of curvature requires the last 3 points on the equilibrium path
    IF (KINC.GE.3) THEN
      S1 = SQRT((ALL_LPFI(KINC)-ALL_LPFI(KINC-1))**2+(ALL_DOF(KINC)-ALL_DOF(KINC-1))**2)
      S2 = SQRT((ALL_LPFI(KINC)-ALL_LPFI(KINC-2))**2+(ALL_DOF(KINC)-ALL_DOF(KINC-2))**2)
      S3 = SQRT((ALL_LPFI(KINC)-ALL_LPFI(KINC-3))**2+(ALL_DOF(KINC)-ALL_DOF(KINC-3))**2)
      A1 = (ALL_DOF(KINC)-ALL_DOF(KINC-1)) * (ALL_LPFI(KINC-1)-ALL_LPFI(KINC))
      A2 = (ALL_DOF(KINC)-ALL_DOF(KINC-2)) * (ALL_LPFI(KINC-2)-ALL_LPFI(KINC))
      PRODUCT_TRIANGLE_SIDES = S1*S2*S3; AREA_TRIANGLE = 0.5D0*ABS(A1-A2)
      ALL_RCURV(KINC-2) = 0.25D0*PRODUCT_TRIANGLE_SIDES/AREA_TRIANGLE
    END IF
  END IF
  END DO
  110 CONTINUE
  C
  Kill Condition 1 (KC1) is when Current LPF < Previous LPF
  IF (KINC.GE.2) THEN
    IF (ALL_LPFI(KINC).GE.ALL_LPFI(KINC-1)) THEN
      CONTINUE
    ELSE
      LSTOP = 1
      PRINT *, "Kill Condition 1 met."
    END IF
  END IF
  C
  Kill Condition 2 (KC2) is when Current LPF - Previous LPF < TOL_INC_LPFI
  IF (ABS(ALL_LPFI(KINC)-ALL_LPFI(KINC-1)).GE.TOL_INC_LPFI) THEN
    CONTINUE
  ELSE
    LSTOP = 1
    PRINT *, "Kill Condition 2 met."
  END IF
  END IF
  C
  Kill Condition 3 (KC3) is when CHANGE_RCURV > TOL_CHANGE_RCURV, delayed to at least increment 10
  IF (KINC.GE.10) THEN
    CHANGE_RCURV = ABS(ALL_RCURV(KINC-3)-ALL_RCURV(KINC-2))/ALL_RCURV(KINC-3)
    IF (CHANGE_RCURV.LE.TOLCHANGE_RCURV) THEN
      CONTINUE
    ELSE
      LSTOP = 1
      PRINT *, "Kill Condition 3 met."
    END IF
  END IF
  END IF
  LOVRWRT = 1
  RETURN
END

Subroutine 2. ABAQUS user subroutine GN_Riks_killer.f for the automated termination of geometrically nonlinear 'Riks' GNAs, GNIs or GMNIs.
References

[1] Rotter JM. The new method of Reference Resistance Design for shell structures. In: Proceedings SD5S 2016, International Colloquium on Stability and Ductility of Steel Structures; 2016. p. 623–30.

[2] Rotter JM. Advances in understanding shell buckling phenomena and their characterisation for practical design. In: Gizejowski MA, Kozłowski A, Marcinowski J, Ziółko J, editors. Recent Progress in Steel and Composite Structures. London: CRC Press, Taylor and Francis; 2016. p. 2–15.

[3] Flügger W. Statik und Dynamik der Schalen. 1st Edition. Berlin: Springer Verlag; 1934.

[4] Timoshenko S, Woinowsky-Krieger. Theory of Plates and Shells. 1st Edition. New York: McGraw-Hill; 1940.

[5] Kutter WT. The stability of Elastic Anisotropic PhD Thesis. The Netherlands: Technical University of Delft; 1945.

[6] Bushnell D. Computerized Buckling Analysis of Shells. Dordrecht: Martinus Ni- jhoff Publishers; 1985.

[7] Teng JG. Buckling of thin shells: recent advances and trends. Appl Mech Rev 1996;49(4):263–74.

[8] Teng JG, Rotter JM. Buckling of thin shells: An overview. In: Teng JG, Rotter JM, editors. Buckling of Thin Metal Shells. London: Spon; 2004. p. 1–41.

[9] Bushnell D. BOSORS — Program for buckling of elastic-plastic composite shells of revolution including large deflections and creep. Comput Struct 1976;6:221–39.

[10] Teng JG, Rotter JM. Elastic-plastic large deformation analysis of axisymmetric shells. Comput Struct 1989;31(2/3):231–33.

[11] ABAQUS (2014) “ABAQUS v. 6.14-2” Commercial FE Software and Documentation. Dassault Systèmes, Simulia Corporation, Providence, RI, USA.

[12] OpenSees. OpenSees v. 2.4.0. Computer software. UC Berkeley; 2006 http://opensees.berkeley.edu.

[13] EN 1993-1-6. Eurocode 3: Design of Steel Structures. Part 1-6: Strength and Stability of Shell Structures. Brussels: Comité Européen de Normalisation, CEN; 2007.

[14] Publication No. 125Revised 2nd Impression ECCS EDRS#2. Buckling of steel shells — European Design Recommendations. European Convention For Constructional Steelwork. Rotter JM, Schmidt H, editors. 5th Edition ECCS, Brus- sels; 2012.

[15] Rotter JM. Shell buckling design and assessment and the LBA-MNA methodology. Stahlbau 2011;80(11):791–803.

[16] Rotter JM. Stability and plasticity in structural analysis: a new conceptual framework. In: Shen ZY, Li GQ, Chan SL, editors. Advances in Steel Structures. Elsevier; 2005. p. 1815–26.

[17] Rotter JM. A framework for exploiting different computational assessments in structural design. In: Proceedings of the 6th International Conference on Steel and Aluminium Structures, Oxford, UK; 2007.

[18] Rotter JM. New segment in Annex E of EN 1993-1-6 on cylindrical shells under global bending and spherical shells under external pressure. Amendment AM-1-6-2013-13 to EN 1993-1-6 approved by CEN/TC250/SC3, November Zurich, Switzerland; 2013.

[19] Rotter JM. The practical design of shell structures exploiting different methods of analysis. In: Pietraszkiewicz W, Szymczak C, editors. Shell Structures: Theory and Applications. London: Talor and Francis; 2005. p. 71–86.

[20] Diederich C, Rotter JM. Generalised capacity curves for stability and plasticity: application and limitations. Thin-Walled Struct 2011;49:1132–40.

[21] Sadowski AJ, Rotter JM. Steel silos with different aspect ratios: II — behaviour under eccentric discharge. J Constr Steel Res 2011;67:1545–53.

[22] Sadowski AJ, Rotter JM. Exploration of novel geometric imperfection forms in buckling failures of thin-walled metal silos under eccentric discharge. Int J Solids Struct 2013;50:781–94.

[23] EN 1993-1-6:2005. Eurocode 3: Design of Steel Structures. Part 1-6: Strength and Stability of Shell Structures. Amendment A1. Brussels: Comité Européen de Normalisation, CEN; 2017.

[24] Teng JG. Song CY. Numerical models for nonlinear analysis of elastic shells with eigenmode-affine imperfections. Int J Solids Struct 2003;40(24–25):3253–80.

[25] Tafreshi A. Buckling and post-buckling analysis of composite cylindrical shells with cutouts subjected to internal pressure and axial compression loads. Int J Press Vessels Pip 2002;79(5):351–9.

[26] Hübner A, Teng JG, Saal H. Buckling behaviour of large steel cylinders with patterned welds. Int J Press Vessels Pip 2006;83:13–26.

[27] Chen L, Doerich C, Rotter JM. A study of cylindrical shells under global bending in the elastic-plastic range. Steel Constr Des Res 2008;1(1):59–65 Stahllbu.

[28] Limam A, Lee L-H, Corona E, Kyríaides S. Inelastic wrinkling and collapse of tubes under combined bending and internal pressure. Int J Mech Sci 2010;52:637–47.

[29] Silvestre N, Gardner L. Elastic local post-buckling of elliptical tubes. J Constr Steel Res 2011;67:281–92.

[30] Rotter JM, Sadowski AJ, Chen L. Nonlinear stability of thin elastic cylinders of different length under global bending. Int J Solids Struct 2014;51:2826–39.

[31] Buchanan C, Gardner L, Liew A. The continuous strength method for the design of circular hollow sections. J Constr Steel Res 2016;118:207–16.

[32] Gardner L, Yun X, Macorini L, Kucukler M. Hot-rolled steel and steel–concrete composite design incorporating strain hardening. Structures 2017;9:21–8.

[33] Sadowski AJ, Rotter JM, Reineke T, Ummenhofer T. Statistical analysis of the material properties of selected structural carbon steels. Struct Safety 2015;53:26–35.

[34] Sadowski AJ, Rotter JM, Stafford P, Reineke T, Ummenhofer T. On the grade of the yield plateau in structural carbon steels. J Constr Steel Res 2017;130:120–30.

[35] Kobayashi T, Mihara Y, Fuji F. Path-tracing analysis for post-buckling process of elastic cylindrical shells under axial compression. Thin-Walled Struct 2012;61:180–7.

[36] Sadowski AJ, Rotter JM. Modelling and behaviour of cylindrical shells structures with helical features. Comput Struct 2014;133:90–102.

[37] Riks E. An incremental approach to the solution of snapping and buckling problems. Int J Solids Struct 1979;15:529–51.

[38] Doerich C, Rotter JM. Accurate determination of plastic collapse loads from finite element analyses. J Press Vessel Technol 2011;133 pages 011202-1 to 011202-10.

[39] Intel. Intel® Parallel Studio XE 2016. Commercial Software. Intel Corporation; 2016 http://software.intel.com/en-us/intel-parallel-studio-xe.

[40] Doerich C. Effect of geometric nonlinearity of locally-supported cylinders. In: Structures And Granular Solids — From Scientific Principles To Engineering Ap- plications. Taylor and Francis; 2008. p. 183–97.

[41] Sadowski AJ, Rotter JM. Buckling of very slender metal silos under eccentric discharge. Eng Struct 2011;33:1187–94.

[42] Yamaki N. Elastic Stability of Cylindrical Circular Shells. North-Holland: Elsevier Science; 1984.

[43] Rotter JM, Teng JG. Elastic stability of cylindrical shells with weld depression. ASCE J Struct Eng 1989;115(5):1244–63.

[44] Fajuyitan OK, Sadowski AJ, Wadee MA, Rotter JM. Nonlinear behaviour of short elastic cylindrical shells under global bending. Under review 2017.

[45] Sadowski AJ, Rotter JM. On the relationship between mesh and stress field ori- entations in linear stability analyses of thin plates and shells. Finite Elem Anal Des 2013;73:42–54.

[46] Chapter 1 Knight NF Jr. Finite element techniques for nonlinear post buckling and collapse of elastic structures. Structural Dynamic Systems Computational Techniques and Optimisation — FE Analysis. Lenedes CT, editor. Chapter 1 Gordon & Breach Science Publishers; 1998.

[47] Stephens WB, Starnes JH, Almroth BO. Collapse of long cylindrical shells under combined bending and pressure loads. AIAA J 1975;13(1):20–5.

[48] Song CY, Teng JG, Rotter JM. Imperfection sensitivity of thin elastic cylin- drical shells subject to partial axial compression. Int J Solids Struct 2004;41(24-25):7155–80.

[49] Rotter JM. Cylindrical shells under axil compression. In: Buckling of Thin Metal Shells. London: Spon Press; 2004. p. 42–87.

[50] Fatdorost SR. A simplified method of elastic stability analysis for thin cylindrical shells II — modified equilibrium equation, Virginia: National Advisory Commit- tee for Aeronautics: 1947. Report No. 1342.

[51] Calladine CR. Theory of Shell Structures. Cambridge: University Press; 1983.

[52] Chen L, Rotter JM. Buckling of anchored cylindrical shells of uniform thickness under wind load. Eng Struct 2012;41:199–208.

[53] Matlab (2016) “Matlab R2016a programming environment” Mathworks, Natick, MA, USA.