International Journal of Pressure Vessels and Piping 120-121 (2014) 141–151

Contents lists available at ScienceDirect

International Journal of Pressure Vessels and Piping

journal homepage: www.elsevier.com/locate/ijpvp

Integrated structural analysis tool using the linear matching method part 1 – Software development

James Ure [a], Haofeng Chen [a,*], David Tipping [b]

[a] Dept of Mechanical and Aerospace Engineering, University of Strathclyde, 75 Montrose Street, Glasgow G1 1XJ, Scotland, United Kingdom
[b] Central Engineering Support, EDF Energy Nuclear Generation Ltd., Barnwood, Gloucester GL4 3RS, United Kingdom

ARTICLE INFO

Article history:
Received 29 January 2014
Received in revised form
2 May 2014
Accepted 7 May 2014
Available online 23 May 2014

Keywords:
Linear matching method (LMM)
Structural integrity
Load carrying capacity
Shakedown limit
Ratchet limit
Steady state cycle

ABSTRACT

A number of direct methods based upon the Linear Matching Method (LMM) framework have been developed to address structural integrity issues for components subjected to cyclic thermal and mechanical load conditions. This paper presents a new integrated structural analysis tool using the LMM framework for the assessment of load carrying capacity, shakedown limit, ratchet limit and steady state cyclic response of structures. First, the development of the LMM for the evaluation of design limits in plasticity is introduced. Second, preliminary considerations for the development of the LMM into a tool which can be used on a regular basis by engineers are discussed. After the re-structuring of the LMM subroutines for multiple central processing unit (CPU) solution, the LMM software tool for the assessment of design limits in plasticity is implemented by developing an Abaqus CAE plug-in with graphical user interfaces. Further demonstration of this new LMM analysis tool including practical application and verification is presented in an accompanying paper.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Many Engineering components and structures in defence, aerospace, petrochemical, automobile and power industries operate under cyclic thermal and mechanical load conditions, and would include such diverse products as advanced internal combustion (IC) engine and gas turbine components, high altitude ramjet and rocket motors, chemical reactor vessels in plastics manufacture, prototype fusion reactors, power boilers, etc. In all of these applications, it is important to ensure the equipment can operate safely for the specified design life under the arduous environmental conditions. This requires engineers to identify possible failure mechanisms and guard against these during the design process [1].

Engineering structures subjected to cyclic loading histories will experience either elastic/plastic shakedown or ratchetting depending upon the applied load level. Limiting the behaviour of the structure or component to the elastic range is not an effective approach to a problem, as this leads to over-conservative design. In many applications, it is acceptable to allow limited plastic deformation to occur provided it can be shown that the structure shakes down to elastic action in the first few cycles of load. If the applied load is below the elastic shakedown limit, ratchetting and plastic shakedown will not occur under repeated loading. However in some situations, for example in nuclear power applications with cyclic thermal loading, this elastic shakedown limit can also be over-conservative. Thus an alternative approach is required to allow plastic shakedown to occur but preclude ratchetting. In ratchetting, a net increment of plastic strain occurs with each cycle of load and leads to an incremental plastic collapse over a number of cycles. Guarding against incremental plastic collapse by the determination of plastic shakedown limit or ratchet limit is crucial in any design involving cyclic thermal and mechanical loads. Under plastic shakedown condition, a low cycle fatigue (LCF) analysis would also be undertaken to ensure the structure does not fail by low cycle fatigue associated with local alternating plasticity, where the number of cycles to failure is determined by the maximum plastic strain range. A steady state cyclic analysis is often sought to evaluate the LCF life and address whether shakedown or ratchetting occurs under the defined cyclic load condition.

Design limits in plasticity for components subjected to cyclic thermal and mechanical load conditions, including both shakedown and ratchet limits, have attracted the attention of many researchers. The incremental finite element (FE) analysis [2] allows the investigation of any type of load cycle but inevitably requires

---

DOI of original article: http://dx.doi.org/10.1016/j.ijpvp.2014.05.005.
* Corresponding author. Tel.: +44 1415482036.
E-mail address: Haofeng.chen@strath.ac.uk (H. Chen).

http://dx.doi.org/10.1016/j.ijpvp.2014.05.004
0308-0161/© 2014 Elsevier Ltd. All rights reserved.

significant computer effort for complex 3D structures. There has been a recent trend towards the development of direct methods that combine the convenience and efficiency of rule based methods [1] and the accuracy of incremental FE simulation techniques. Of these simplified direct methods [3–9], the Linear Matching Method (LMM) [10–14] has become one of the most powerful numerical methods for generating approximate inelastic solutions and answering specific design related issues using standard finite element codes. The basis of the LMM is through an idea of representing histories of stress and inelastic strain as the solution of a linear problem, where the linear moduli are allowed to vary both spatially and in time. The LMM has been formulated and implemented for the evaluation of shakedown limit [10,11] and ratchet limit [12,13]. And more recently, a new LMM framework was developed to evaluate the steady state cyclic behaviour of component for the LCF assessment purpose [14].

The LMM is distinguished from the other upper bound or lower bound direct methods by ensuring that both the equilibrium and compatibility conditions are satisfied to produce exact solution at each stage of calculations [13], and is counted to be one of the methods most amenable to practical engineering applications involving complicated thermo-mechanical load history [15]. However, as many other direct methods, the LMM was initially implemented into commercial FE software Abaqus [2] using user subroutines developed by FORTRAN programming language, which is difficult for non-experts to operate. Another drawback of this implementation is the level of programming experience required to create and submit an analysis: the alterations to the subroutines required to run each analysis present issues for everyday use by engineers generally not familiar with FORTRAN. To remedy this situation and enable widespread adoption of the LMMs in industry, an integrated software tool is required to not only removes the requirement for manual subroutine alterations, but also provide additional functionality for subsequent life assessment calculations.

The main objective of this paper is to develop a new integrated structural analysis tool using the LMM framework for the assessment of design limits in plasticity, including the evaluation of limit load, shakedown limit, ratchet limit and steady state cyclic behaviour of the structure. It will deliver the LMM in a form where it can be readily used by engineers with the responsibility for design and life assessment decisions on a regular basis. The software tool would allow engineers to access the LMM solution methods without having to make any of the changes to the subroutines required to run a LMM analysis as was previously required.

The paper is organized as follows. In Section 2, preliminary considerations are discussed by examining previous LMM versions and the way of customising Abaqus. In Section 3, a re-structuring of the LMM user-subroutines for multiple CPU solution is proposed. The creation of a Graphical User Interface (GUI) via an Abaqus Plug-in is presented in Section 4. Section 5 briefly discusses the installation and testing of the LMM software tool for the assessment of design limits in plasticity. Section 6 concludes this paper.

2. Preliminary considerations

2.1. Previous LMM versions

The original incarnation of the LMM code was created as Abaqus user subroutines and has been mainly used for research purposes. A typical LMM analysis consists of two stages [12,13]. In the first stage an elastic analysis for each applied load and temperature distribution is performed using the elastic analysis UMAT subroutine. For each of these analyses the elastic stress tensor for each integration point is written to a text file, and the integration point temperature is written to a separate text file. The second stage of this analysis uses a second UMAT subroutine and these text files to perform the shakedown or ratchet limit calculation. In this second stage some changes are required to the UMAT code in order to set up the analysis. For example, the number of integration points per element and the total number of elements in the model needed to be changed so that the arrays could be sized appropriately. The code defining the load cycle also requires updating, which reads the stress and temperature text files to generate the applied stresses at each point in the load cycle.

For an analysis using this set of subroutines both the elastic analyses and the further shakedown or ratchet calculation are submitted using the Abaqus batch command i.e. the Abaqus input file for the model is required. This input file is generated using Abaqus CAE for a complete model. The majority of the content of the input file is common between the elastic and shakedown calculations (such as geometry and boundary conditions). However there are some differences which must be performed manually (such as requesting the energy outputs associated with a UMAT subroutine). A further upgrade of these subroutines was carried out [16] so that the LMM could be used with minimal code changes, where the load cycle is defined via a formatted text file which was read by the subroutines. This significantly reduces the code changes required for an analysis. However, the changes to the Abaqus input file still needed to be performed manually.

The creation of a formatted text file to configure the LMM analysis was a major step in the usability of the LMM, and in fact draws a parallel with the way in which any conventional Abaqus analysis operates. In an Abaqus analysis, a FE model needs to be created in Abaqus CAE and submitted for analysis, although other pre and post processors are often used. Abaqus CAE then creates a formatted text file (Abaqus input file) which is passed to the Abaqus solver for solution. The text file for the LMM analysis is equivalent to the input file, the only difference being that it is created manually. The creation of a text file is also adopted here as it is a simple and robust method for passing information from the LMM user interface into the subroutines. The major aim of this software tool is that the text file is generated by the user interface rather than manually.

2.2. Customising Abaqus

Abaqus [2] contains a large number of options for the user to customise a model or analysis for their particular situation. To obtain user-generated solution options the user-subroutines can be used, which is how the LMM has been implemented. In addition to this Abaqus CAE contains the option to use scripts to perform operations on the model or results databases. These scripts are written in the Python open source scripting language [2], and Abaqus has extended this language to allow operations to be performed within CAE itself. These scripts can be used to perform all operations which are available through the CAE interface (i.e. applying loads, meshing, plotting results etc.) and can also query the model or Abaqus output file (odb) for values. A typical example where scripts serve a useful function is in a sensitivity analysis, where an automatic process can vary a particular value in a model, re-submit for solution, query the results and decide whether a further iteration is required.

The use of python scripting within Abaqus is a very powerful tool, because options also exist to use this language to customise the CAE user interface itself. This can be achieved by creating either an entirely custom CAE interface or a plug-in to the standard CAE. The ability to create a custom GUI is a powerful tool as the modules and toolsets which are not desired can be removed and custom functions can be added. Abaqus Viewer is an example of this, where

Download English Version:

https://daneshyari.com/en/article/787806

Download Persian Version:

https://daneshyari.com/article/787806

Daneshyari.com