ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

SoftwareX ▮ (▮▮▮▮) ▮▮▮–▮▮▮

SoftwareX

www.elsevier.com/locate/softx

INSPECT: A graphical user interface software package for IDARC-2D[☆]

Mohammad AlHamaydeh[a,*], Mohamad Najib[a], Sameer Alawnah[b]

[a] *Department of Civil Engineering, American University of Sharjah, United Arab Emirates*
[b] *Department of Computer Science and Engineering, American University of Sharjah, United Arab Emirates*

Received 8 May 2016; received in revised form 18 August 2016; accepted 21 October 2016

Abstract

Modern day Performance-Based Earthquake Engineering (PBEE) pivots about nonlinear analysis and its feasibility. IDARC-2D is a widely used and accepted software for nonlinear analysis; it possesses many attractive features and capabilities. However, it is operated from the command prompt in the DOS/Unix systems and requires elaborate text-based input files creation by the user. To complement and facilitate the use of IDARC-2D, a pre-processing GUI software package (INSPECT) is introduced herein. INSPECT is created in the C# environment and utilizes the .NET libraries and SQLite database. Extensive testing and verification demonstrated successful and high-fidelity re-creation of several existing IDARC-2D input files. Its design and built-in features aim at expediting, simplifying and assisting in the modeling process. Moreover, this practical aid enhances the reliability of the results and improves accuracy by reducing and/or eliminating many potential and common input mistakes. Such benefits would be appreciated by novice and veteran IDARC-2D users alike.

Keywords: IDARC-2D; GUI; Pre-processor; INSPECT

Code metadata

Current code version	1.0.0.0
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-16-00041
Legal Code License	BSD 3-Clause License
Code versioning system used	git
Software code languages, tools, and services used	C#, Microsoft Visual Studio 2013, SQLite ver. 3.14.1
Compilation requirements, operating environments & dependencies	.Net Framework v4.5.1
If available Link to developer documentation/manual	N/A
Support email for questions	malhamaydeh@aus.edu

Software metadata

Current software version	1.0.0.0
Permanent link to executables of this version	https://github.com/alhamaydeh/INSPECT/tree/master/Installer
Legal Software License	BSD 3-Clause License
Computing platforms/Operating Systems	Windows OS.
Installation requirements & dependencies	.Net Framework v4.5.1
If available, link to user manual — if formally published include a reference to the publication in the reference list	N/A
Support email for questions	malhamaydeh@aus.edu

☆ *INSPECT*: INelastic Seismic Performance Evaluation Computational Tool.

* Corresponding author. Dr. Mohammad AlHamaydeh, Associate Professor, Department of Civil Engineering, American University of Sharjah, P.O. Box 26666, Sharjah, United Arab Emirates. Tel.: +971 6 515 2647; fax: +971 6 515 2979.

E-mail addresses: malhamaydeh@aus.edu (M. AlHamaydeh), b00045708@aus.edu (M. Najib), salawnah@aus.edu (S. Alawnah).

http://dx.doi.org/10.1016/j.softx.2016.10.004

1. Motivation and significance

When utilizing nonlinear inelastic analysis techniques, the seismic performance of structures can be predicted, to a reasonable accuracy. As a result, the necessity for significant analytical accuracy and efficiency has dramatically increased in the past two decades or so. Carrying out such sophisticated analyses with justifiable confidence is often hindered by the difficulty of utilizing the tools and models capable of delivering the necessary level of accuracy [1,2]. For instance, developing accurate estimates of the seismic demands and the associated design factors is very crucial for the design of Reinforced Concrete (RC) frames. Such accurate estimates become even more influential when the nonlinearity effects on the seismic behavior of the RC frames are investigated [3–5]. Inelastic and nonlinear effects are rather sophisticated and thus require accurate assessment for seismic design purposes [6,7]. Unfortunately, these inelastic and nonlinear effects are not captured by the more popular and often much simpler linear elastic analysis for production design purposes. By providing more realistic estimates of structural behavior during seismic events, nonlinear analysis enables the designer to limit and control the expected levels of damage that could potentially take place. Given the modern advances in computational power, nonlinear analysis has become feasible and accessible to the design engineer more than ever before. With this increased demand for nonlinear analysis necessary for Performance-Based Design (PBD), the need for robust and reliable analytical tools and software is immediately realized. In the following paragraphs, a brief survey of some of the available software tools is presented.

Perhaps among the most widely used in the structural and earthquake engineering industry, the Computers and Structures, Inc. suite of commercial packages (ETABS [8], SAP2000 [9], and Perform-3D [10]) are all equipped with GUIs. With various differing degrees and features, they are all capable of performing sophisticated inelastic, nonlinear static and dynamic analyses. For brevity, only Perform-3D is discussed herein. Perform-3D is an evolved GUI environment for its command prompt series of ancestors DRAIN-2D, DRAIN-2DX, and DRAIN-2D+ (the latter utilized the graphic processor “VIEW2D”). It is a software platform that facilitates implementing the displacement-based design procedures specified in ASCE 41-13 [11] as well as performs capacity design checks due to seismic effects. The user-friendly GUI allows for efficient and accurate analysis of structures under strong ground motions. Some of the analysis techniques that are implemented include modal analysis, response spectrum, pseudo-static (push-over), in addition to response history for earthquake ground motion and other dynamic forces.

Zeus-NL [12] is an open source three dimensional analysis and seismic simulation platform. It too, is capable of performing nonlinear analysis and it is equipped with a user-friendly GUI. It implements several nonlinear material models as well as predefined member cross-sections in a large library. It can be used to perform different types of analysis including eigenvalue, pseudo-static (pushover), dynamic time–history analysis as well as dynamic pushover or Incremental Dynamic Analysis (IDA).

Another powerful yet relatively recent software framework is SeismoStruct [13]; it utilizes the finite element method to perform static and dynamic nonlinear structural analysis for frames under different types of loading conditions. It has a complete GUI with a very user-friendly pre/post-processing capabilities including batch mode for carrying out automated multiple analyses and IDA. Two or three dimensional models can be easily and quickly produced to predict the large displacement behavior and run eight different types of analysis including IDA, pseudo-static loading, and response spectrum analysis. Moreover, an extensive material library is available where concrete, steel and Fiber-Reinforced Polymer (FRP) material models can be implemented with various predefined section configurations. SeismoStruct is also capable of performing capacity checks in accordance to several current international design codes.

One of the powerful computational platforms that are currently available is OpenSees [14] (Open System for Earthquake Engineering Simulation). It is an open source software framework which utilizes the finite element method to simulate the seismic response of structural and geotechnical engineering systems. OpenSees offers nonlinear analysis and modeling methods with an extensive range of materials and section models available to users. Its modular design with loose coupling of analysis and model building component allows for users and developers to incorporate their own applications through incremental improvement to the framework. Fluid–structure interaction modeling [15], bridge rating applications [16] and simulation of structural response to fire [17], are some examples of the extension of OpenSees applications. Moreover, the flexible and fully programmable platform allows for deployment of hybrid simulations and control schemes. However, this flexibility is at the expense of complexity; perhaps, the main limitation of OpenSees is that it requires the use of the Tool Command Language (TCL) string-based scripting language to build models and conduct analyses. Therefore, fluent programming skills are required for users to be able to carry out their desired analyses and design checks.

To overcome this hurdle, some pre/post-processor GUIs have been developed such as BuildingTcl/BuildingTclViewer [18]. BuildingTcl is developed to provide a library of TCL commands that enables the use of OpenSees to users with little programming skills. BuildingTclViewer is a GUI that facilitates the creation of BuildingTcl databases, running OpenSees simulations, as well as viewing results, interactively. It allows access to a predefined library of different structural materials, sections and elements which can be used to generate building models and analysis data.

Another pre/post-processor GUI for OpenSees is OpenSees Navigator [19]. It is basically, a MATLAB-based tool running on multiple platforms which offers similar functionality as the BuildingTcl/BuildingTclViewer. Specifically, it furnishes parametric geometry templates that can be used to quickly model some of the common structural systems. It also comes with built-in structural steel shapes cross-sections along

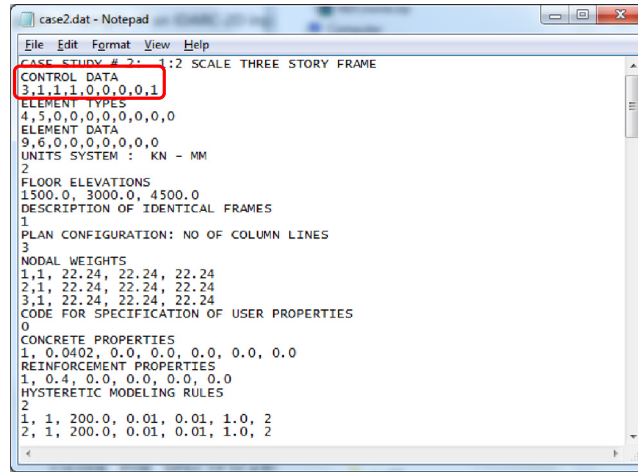
Please cite this article in press as: AlHamaydeh M, et al. *INSPECT*: A graphical user interface software package for IDARC-2D. SoftwareX (2016), http://dx.doi.org/10.1016/j.softx.2016.10.004

with their pre-defined properties; the cross-section database includes AISC (American Institute of Steel Construction) and CISC (Canadian Institute for Steel Construction). It also allows for pre/post-processing and running of OpenSees simulations without programming TCL scripts. As such, it liberates users to create two or three dimensional models in OpenSees with relative ease. In addition, the post-processing is seamlessly processed and managed using built-in toolboxes. OpenSees Navigator also serves as a great tool to visualize structural systems behavior, such as structural element forces, displacements, mode shapes, etc.

Since its original introduction in 1987 at the State University of New York at Buffalo, IDARC-2D [20] (Inelastic Damage Analysis of RC structures) has undergone many enhancements and upgrades. Its macromodel-based formulation promotes the implementation of comprehensive elements capable of exhibiting nonlinear characteristics for most structural members. Such macromodel formulations are either based on mechanics or incorporate fiber models to calculate member properties. The modeled nonlinearities accommodate flexibility (uniform vs. linear) formulations as well as concentrated or distributed plasticity with yield penetration. It utilizes a robust Newmark-Beta scheme for integrating the equations of motion in a step-by-step approach. The P-Delta effects (secondary overturning moments resulting from eccentric application of gravity forces during lateral inter-story drifts) are efficiently accounted for via equivalent unbalanced lateral forces. It is capable of performing the following analysis types combined with pre-specified static loads: (a) inelastic incremental static analysis, (b) pseudo-static "pushover" analysis, (c) inelastic dynamic analysis, and (d) quasi-static cyclic analysis. Because IDARC-2D runs in a command prompt mode (black DOS window) and does not utilize any Windows graphics, its numerical solution efficiency is quite remarkable. That feature, combined with its low desk space usage due to the text-based input/output files, makes IDARC-2D a very useful tool in earthquake engineering. Moreover, some of the features that make IDARC-2D stand out include, but are not limited to: (a) the Damage Index (DI) reporting at the component, story and overall levels, (b) the incorporation of novel passive structural control elements, such as the Rocking Column (RC) which was introduced in the current version 7.0 as well as the Triple Friction Pendulum (TFP) and Negative Stiffness Device (NSD) which are to be included in the upcoming version 8.0.

However, IDARC-2D users need to create an input text file (*.txt) that contains all the necessary parameters and information for the analysis to run. These mainly include the building information, the properties of structural elements, material, and analysis options, etc. The developers of IDARC-2D illustrated how to write/create the input text file through their readily available user's guide [20] accompanied by 12 examples and case studies. A sample of an IDARC-2D input text file is shown in Fig. 1.

At first glance, it can be quickly identified that the input data requires some deciphering. For instance, the "control data" part (inside the red box in Fig. 1) contains the general parameters of the model and analysis; these include the number of stories, number of typical (non-identical) frames, number of concrete, steel, and masonry materials. The entered values indicate a three-story structure with a single frame, one type of concrete and one type of steel reinforcement. The subsequent four zeros are indicating that no masonry material is used, P-Delta effects are to be ignored, spread plasticity and linear flexibility distribution are to be used in the analysis. The last "1" is to have IDARC-2D run in "Windows" mode as opposed to "Unix".


case2.dat - Notepad
File Edit Format View Help

```
CASE STUDY # 2:  1:2 SCALE THREE STORY FRAME
CONTROL DATA
3,1,1,1,0,0,0,0,1
ELEMENT TYPES
4,5,0,0,0,0,0,0,0,0
ELEMENT DATA
9,6,0,0,0,0,0,0,0
UNITS SYSTEM :  KN - MM
2
FLOOR ELEVATIONS
1500.0, 3000.0, 4500.0
DESCRIPTION OF IDENTICAL FRAMES
1
PLAN CONFIGURATION: NO OF COLUMN LINES
3
NODAL WEIGHTS
1,1, 22.24, 22.24, 22.24
2,1, 22.24, 22.24, 22.24
3,1, 22.24, 22.24, 22.24
CODE FOR SPECIFICATION OF USER PROPERTIES
0
CONCRETE PROPERTIES
1, 0.0402, 0.0, 0.0, 0.0, 0.0, 0.0
REINFORCEMENT PROPERTIES
1, 0.4, 0.0, 0.0, 0.0, 0.0
HYSTERETIC MODELING RULES
2
1, 1, 200.0, 0.01, 0.01, 1.0, 2
2, 1, 200.0, 0.01, 0.01, 1.0, 2
```

Fig. 1. Sample IDARC-2D input text file.

In an attempt to circumvent the deciphering process and simplify the creation of the input file, this paper introduces a GUI pre-processor package for IDARC-2D ver. 7.0. This GUI Pre-processor package is *INSPECT* (*IN*elastic ***S***eismic ***P***erformance ***E***valuation ***C***omputational ***T***ool). *INSPECT* facilitates carrying out sophisticated nonlinear analyses, without the burden of learning how to create the necessary text input file for IDARC-2D.

2. Software description

INSPECT is coded using the C# programming language through Microsoft Visual Studio 2013 [21] and ".NET" libraries for all GUI components. C# was chosen because of its technical support availability, flexibility, and small size of the generated files. This package is compatible with windows-based operating systems such as Windows 7 or higher provided that the ".NET" Framework Version 4.0 or higher is installed. Furthermore, SQLite [22] is used to save and load the user created sessions for easier access and added functionality. SQLite does not require installing any software on the client's device, contrary to some other database management systems such as MS-SQL.

The *INSPECT* program structure is very simple and consists of three main parts that are fully integrated and interconnected. The first part is an embedded database to save all required parameters of the modeled structure. For this, the SQLite database was utilized since it is a single file portable embedded database; i.e. the project file is simply the database file. The second part is a series of C#based GUI forms that provide an interactive dynamic interface between the database file and the user. Essentially, the user enters the parameters into the fields on each form and the changes are then reflected into the database file upon saving. Finally, the IDARC-2D input file generator form (see Fig. 2), which reads the structure parameters from the database, generates the input text file (*.dat), prepares the accelerogram file (ground acceleration during earthquakes), if specified, and then deploys the IDARC-2D.exe engine The input text file, IDARC-2D executable engine, and all generated output file(s) are located in the working folder within *INSPECT* for the currently executed session.

The *INSPECT* interface is divided into the same sets of the IDARC-2D user's guide to make its adoption easier for the users who are familiar with IDARC-2D. A flowchart presenting the forms with their respective sub-sections is presented in Fig. 2. These forms were created to match the general outline of the input text file required by IDARC-2D. The main form contains 11 pages/screens that are required to build the analytical model such as: frame geometries, properties of the materials, structural elements, connections, etc. and the analysis form is made of several pages depending on the analysis option specified by the user.

Developers of *INSPECT* used ".NET" GUI components such as: Checkbox, Radio Button, Numeric Up Down, Data Grid View and Combo box in a suitable manner to achieve easy and user-friendly functionality. For example, when more than one choice is possible the use of Checkbox was implemented, while Radio buttons are used wherever a single choice is to be selected. Numeric Up/Down arrows are used for easily scrolling through predefined numerical values such as integers when defining number of columns, number of beams etc. Data Grid Views are used when tabulated information is required such as when inputting story elevations or nodal weights, etc.

Toolbars are implemented in the main form to assist the user in carrying out simple tasks. Users can easily create new files, save and load current or existing files and exit the session from the "File" menu. Additionally, they are able to export the analysis run output folder to a folder of their choice, as well as erase the current session through the "Clean Workspace" option. The "IDARC-2D Manual" menu is created to direct the user to the related pages of the IDARC-2D user's guide for assistance and further information about the components of the screen (context help). This function is also enhanced with the aid of help buttons that are inserted into most pages. These convenient features were added to enhance the users experience and allow for easy accessibility to different files and resources such as the user's guide.

The "default value" feature provides the user with recommended values from the IDARC-2D user's guide for certain parameters. These include material properties such as the initial Young's modulus for concrete, strain at maximum strength of concrete, etc. Whereas for hysteretic rules, default and recommended values are provided for most related parameters such as stiffness degrading parameter, ductility and hysteretic-energy based strength decay parameters and more. For users who want to perform inelastic dynamic analysis, a generic white noise earthquake record can be generated by IDARC-2D. Alternatively, a pre-existing record could be specified through a "wave" text file.

Since the freely-distributed version of IDARC-2D has certain limits for the inputs parameters which control the problem size that can be modeled, the *INSPECT* package continuously checks the user-entered values against these limits. A full listing of these parameters and their pre-set limits can be found in Appendix C of the MCEER-09-0006 report [23]. Warning messages are immediately displayed if the user attempts to exceed any of the limits in the created project. Moreover, any discrepancies based on previously entered values are also reported. For example, if the user attempts to add information for two types of concrete materials when it has been specified that the project has only one type. Just prior to running the IDARC-2D engine, *INSPECT* will warn the user that the project exceeds any of pre-set limits for the free version of IDARC-2D. However, at the user's discretion *INSPECT* will allow the execution of the IDARC-2D engine despite the warnings.

Since often times replication of input values may be required, *INSPECT* has an embedded replication functionality for the input fields in most of the GUI forms. The small sky blue icon to the right of the field can be used to open the "RangeSelector" form, from which the user can specify a range of fields to be populated with the value of the currently selected field. Alternatively, the user may choose to replicate the current field value in all the fields, as can be seen in Fig. 3.

It is worth mentioning here that an earlier alpha version of INSPECT (then-called "INSPECT-Lite"), has been previously introduced in a two-page conference proceedings presentation [24]. The INSPECT-Lite version was inferior to the final INSPECT release in many aspects, had much limited features and lesser functionality (e.g. "RangeSelector") etc.

3. Illustrative examples

The developers of IDARC-2D created 12 case studies that are readily available as illustrative examples. These case studies help demonstrate the capabilities and components of IDARC-2D to new users. The second case study (case study #2) is arbitrarily selected to illustrate the layout and functions of the *INSPECT* software package. It is a model for a three-story building; each story has a height of 1.5 m in a single typical frame. The P-Delta effects are ignored and the spread plasticity option is opted for along with uniform flexibility distribution. The system of units is metric units (kN, mm). One concrete material (fc' = 40 MPa), and one steel reinforcement material (Fy = 400 MPa) are used. There are nine column elements in the frame, characterized by four unique types, and six beams characterized by five unique types. It includes two multi-linear hysteretic rules, one for the column elements and one for the beam elements. Both hysteretic rules are of the type: Vertex Oriented Model, and all of their parameters (stiffness degrading parameter, ductility-based strength decay, hysteretic energy-based strength decay, slip parameter) are set to their default respective values [200, 0.01, 0.01, 1]. Square columns cross-sections are utilized (250 × 250 mm), while the beams cross-sections are rectangular (300 × 150 mm), and 3 m long.

The analysis option is set as cyclic quasi-static, without long-term static loads. The displacements are only applied to story #3

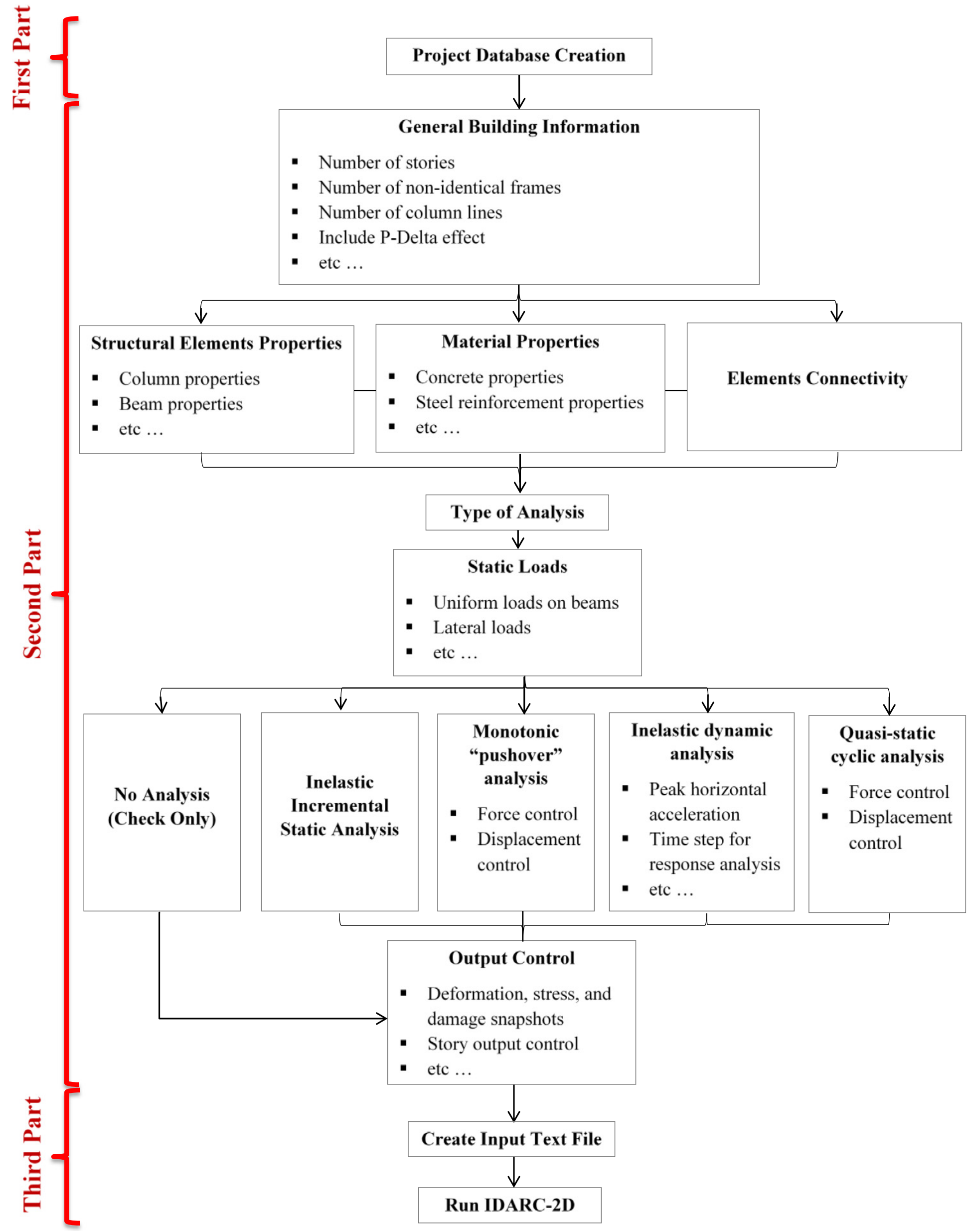


Fig. 2. *INSPECT* structure and model creation flowchart.

and the displacements profile consists of 249 points with a step size of 0.02.

The requested outputs are the three stories response histories reported at each of the quasi-static displacement intervals. Additionally, six columns (1–6) and three beams (1–3) are selected for hysteresis output. Once all the necessary information is entered correctly, clicking the “Create/Update Input File” button, shown in Fig. 4, generates the necessary input text file for IDARC-2D.

The generated text is displayed within the *INSPECT* page for the user to view and/or update as needed prior to invoking the run command for IDARC-2D. Users can run IDARC-2D of that input file from the same page by clicking on “Run IDARC_2D” button. The changes made to the generated text are directly implemented into the analysis and become immediately effective. It is worth mentioning here, that any editing performed on the input file text at this stage will immediately take effect when the “Run IDARC_2D” button is pressed. These changes will then be permanent and cannot be fed back into the database file for the model.

All available examples (12 case studies, including this representative example: Case#2) were recreated using the *INSPECT* software package during the verification stage. The recreation of the models was done from scratch using *INSPECT* based on the problems’ descriptions provided in the IDARC-2D documentation report. The verification involved rigorous comparison of the *INSPECT*-generated files against the reference files provided by the IDARC-2D developers. The comparisons were twofold: input file and output files, both of which were carried out via the WinMerge tool [25]. WinMerge is an open source differencing and merging tool that is capable of comparing both folders as well as files at

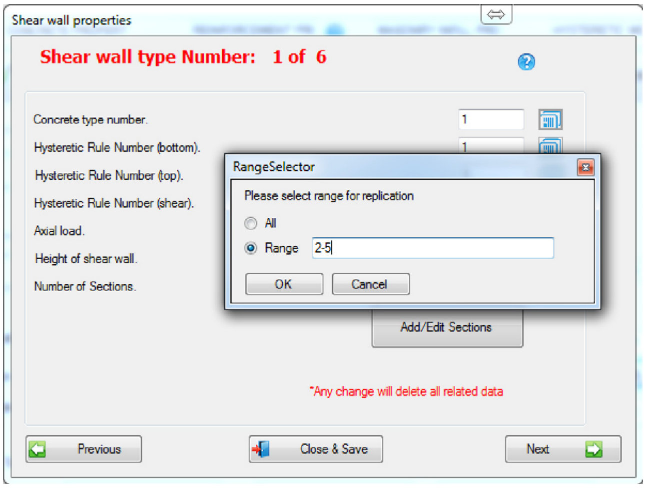


Fig. 3. The range selector form for easy replication. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

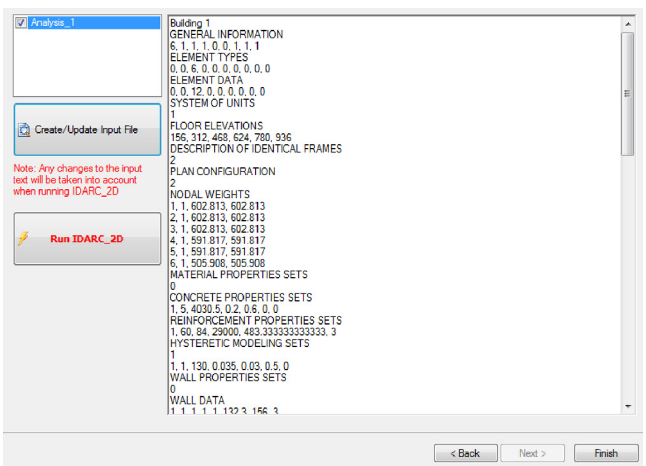


Fig. 4. Sample of *INSPECT*-generated input text file.

the character-level. In the user-provided header lines or text comments and information, etc. hard-to-trace minor variations may sometimes persist due to the associated white spaces. Additionally, the IDARC-2D free format allows for different conventional delimiters (blanks/commas, etc.) to separate input data entries. For example, some user input parameters are presented by equivalent but different set of characters, such as intermediate white spaces in-between the comma-separated input values, etc. Such variations are immaterial and do not in any way impact the validity of the IDARC-2D models. However, their existence in the input files complicates the character-level comparisons of the input files. In such cases, comparing the generated output files can supplement the input file comparisons. This is feasible because IDARC-2D ignores such immaterial user-provided text while reading the input file, and consequently does not regenerate or report them in the output files. Accordingly, the *INSPECT* verification of the 12 case studies was carried out by character-level comparison of the input files (*INSPECT*-generated against original IDARC-2D), as well as the generated output files upon running them.

Fig. 5(a) illustrates that the results of the comparison between the input files of case study #2 (original IDARC-2D versus INSPECT-generated). It can be seen that at the character-level, there are some insignificant differences that are highlighted in yellow (identical characters are left un-highlighted). The differences are due to either white spaces and/or user-entered text. On the other hand, Fig. 5(b) compares the two folders containing the input and output files. It is clear that despite the differences in the input files, the generated output files are all identical. Furthermore, Fig. 5(c) shows some of the comparison details of the main output files (“case2_IDARC-2D.out” and “case2_INSPECT.out”); identical results are reported in both output files. The damage indices are used as a demonstrative sample of the identical output since they are the final part of the output files and cannot be matching unless all other results are also identical. Further details regarding the specifics of case study #2 can be found in the IDARC-2D documentation.

4. Impact

The main advantages of using an input text file method include, easy portability, low disk space and more importantly, they can be universally accessed and edited through the Operating System (OS) without the need for additional software. On the other hand, an input text file method suffers from many drawbacks. These include the difficulty to resolve unintentional input mistakes in the text file and inability to process multiple runs easily and/or simultaneously. Unintended input mistakes in input text files that are required by software could prevent the runs from being carried out and/or may produce erroneous results. Hence, the user should be experienced and very familiar with the necessary format and parameters of the input text file. Such errors could be due to, unintentional and/or unnecessary white spaces, invalid and/or insufficient parameters, etc. which may lead the IDRAC-2D executable to either not recognize the subsequent sections correctly or even crash and abort execution. It is worth mentioning here, that the IDARC-2D user’s manual explicitly warns against introduction of unexpected white lines. Another example of invalid parameters is when the user does not abide by a specified limit or range for an input variable that is essential to the program’s operation. For instance, when the user opts to carry out a nonlinear inelastic dynamic analysis, it is essential that the analysis time step is appropriately chosen. The value of 0.005 s is recommended as a maximum time step in order to avoid having the simulation experience numerical instabilities resulting from excessive unbalanced forces. Such unbalanced forces could result in inaccuracies of the damage indices (and potentially producing values in excess of 300%) or ultimately failure to achieve convergence.

In other cases, integer values are the only valid parameter entries such as when inputting the number of column types, number of floors, etc. Other types of invalid entries that are sometimes accidentally produced and are very difficult to be detected and rectified. Of these types, there are several parameters that may potentially cause terminations if the

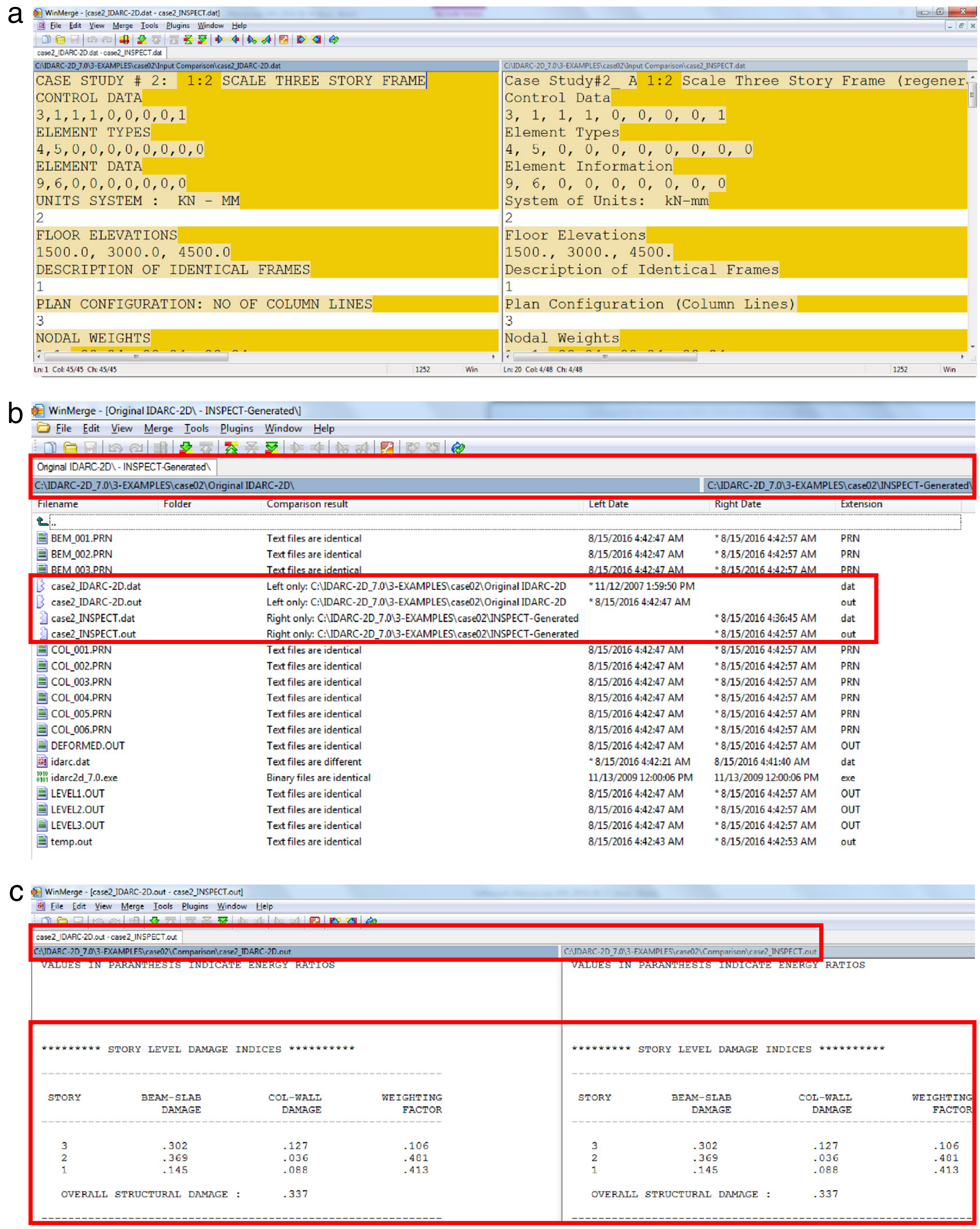


Fig. 5. Comparison of input and output files for case study #2: (a) character-level comparison of input files (b) folder comparison for input/output files (c) identical damage indices reported in output files.

subsequent pertinent section(s) do not strictly agree with the earlier specified parameters. For example, when the user specifies the number of stories (or columns, beams, etc.) to be a particular number, and in the subsequent pertinent section(s), more or less floor elevations or nodal weights, etc. are entered. Such errors in the input text file would lead to termination of the IDARC-2D session. Unfortunately, often times, without any indication of the underlying issue.

Moreover, when performing inelastic dynamic analysis for an earthquake record, IDARC-2D users need to make sure that the specified analysis time steps are matching each time incidence in the input ground motion. This is essential for attaining accurate results from the analysis. Multiple intermediate analysis steps may be specified in-between ground motion record points. Hence, IDARC-2D requires the ratio of the analysis time step to the ground motion time step to be maintained as an integer value. This check is performed by the IDARC-2D user, and thus can be quite challenging and cumbersome when several records are to be used in the considered problem. *INSPECT* users are relieved of such cumbersome checks as they are internally executed automatically; discrepancies are reported if they are found. This feature makes the process of carrying out inelastic dynamic analyses much smoother.

Furthermore, the freely-distributed version of IDARC-2D has many limitations on the problem size that can be handled, by imposing maximum limits on most input parameters (number of floors, column/beam elements, etc.). The specifics of these limits are found in Table C.1 of the technical report MCEER-09-0006 [23]. Users who are unfamiliar with these restrictions are warned by *INSPECT* when these limits are exceeded. A warning icon will pop up with the maximum permissible value displayed in a text box within the same window page. These restrictions can be overridden by simply replacing the freely-distributed executable file “IDACR-2D.EXE” with a custom built executable that allows for the desired problem size. The customized executable is easily built by users with access to the IDARC-2D source code via a standard FORTRAN compiler (editing the file IDDEFN.FOR). The process is fully explained in the IDARC-2D documentation. Users with customized executables need only to copy the newly built executable into the root folder of the INSPECT software package which was specified during installation (*e.g. C*:*INSPECT*\).

It is worth mentioning herein that this flexibility feature in the *INSPECT* software package, fosters a selective untethering option from the IDARC-2D engine and supports multi-version operation modes i.e. the option to utilizing customized IDARC-2D executables into *INSPECT* allows users to upgrade to multiple versions of the IDARC-2D engine while using an existing (or even an outdated) *INSPECT* package. This is an advantageous attribute of text-based input files. If the model to be created by *INSPECT* does not include any of the differing features and/or elements in the IDARC-2D engine version, the user can continue to use it seamlessly. Upon completion of the input file generation process, *INSPECT* makes the input text fully available to the user for complete review and/or editing prior to running the engine. Of course, the available copy/paste feature accommodates the use of any text editor for rapid and large-scale manipulation (e.g. “find”, “find and replace”, etc.). Subsequently, the updated text can be inserted back into the *INSPECT* window prior to invoking the “Run IDARC_2D” command.

In the case that the *INSPECT* user intends to use some of the upgraded features, depending on their amount, the consequences to the user may vary. For example, upon the introduction of the inelastic shear-bending beams and columns into IDARC-2D version 6.0, the “IBTYPE” input variable was introduced. This integer was to be entered in an individual line preceding the beam data set [IBTYPE = 1 meant “Regular Beam”, while IBTYPE = 2 meant “Deep Beam”]. Consequently, any input file prepared for previous versions of IDARC-2D needed to be updated with this new line for each beam data set if version 6.0 was to be utilized for the analysis. So in effect, the upgrade of existing input files to version 6.0 was carried out by simply adding a line with a single character. Such minor corrections and updates can be easily implemented into the input text prior to running the IDARC-2D engine. In similar scenarios, the use of the existing *INSPECT* package is an obvious and convenient choice until an updated version is released. This kind of wide-range practicality can be readily appreciated by any veteran user of IDARC-2D. On the other hand, if the *INSPECT* user intends to use some of the completely new elements, (e.g. Triple Friction Pendulum (TFP) or Negative Stiffness Device (NSD) elements that are to be included in the upcoming IDARC-2D version 8.0), some additional effort will be needed. Despite the apparent difficulty, including such limited text portions into a previously generated input file, can be quite simple and will require very little learning of the specifics of the relevant input data set(s).

In summary, *INSPECT* provides many automatic checks for most of the common mistakes that are typically encountered by IDARC-2D users. When combining the automatic checks with the built-in “default value” features, the model creation process is streamlined and rendered much more efficient. Thus, eliminating many of the hurdles previously faced when attempting to adopt IDARC-2D as a nonlinear analysis platform. *INSPECT* resolves the most of the difficulty associated with text-file creation for IDARC-2D. Consequently, it simplifies the input process for experienced users who are facing the challenge of creating large models. Additionally, it provides a very mild learning curve for the novice and less experienced IDARC-2D users. By doing so, it trail blazes an often less-traveled path to performance-based design. Establishing the *INSPECT* platform serves as a launching pad for a post-processor GUI that will assist in carrying out and automating performance-based evaluations. The type of evaluations that requires very large number (thousands) of runs to be performed and have their results processed and displayed in a concise manner (tabulated and/or plotted results) that is informative to the structural/earthquake engineer.

5. Conclusions

Many of the previous generation software are phasing out typing-based input methods such as code or text files and are gravitating towards GUI environments. These visual enhancements expand the popularity of currently existing tools, improve their efficiency and accuracy, as well as, simplify and expedite the learning process for new users. *INSPECT* is developed to facilitate the process of creating the input text file required by IDARC-2D to run. This visual environment simplifies the interaction of the user with IDARC-2D through graphical presentation of the required input parameters, commands or files. An added benefit of a GUI environment is the reduction and/or elimination of errors stemming from the specific requirements of the layout and format of the input text file. Moreover, the GUI environment enables detection of invalid/insufficient input parameters, as well as conflicting ones. Furthermore, error-free analysis is attained by restricting the input parameters to only one of the available options that is accepted or recommended by the software. These types of errors generally prohibit the analysis from initializing, might cause the analysis to abort, or report inaccurate results. *INSPECT* reduces the need for the novice user to have to get fairly acquainted with the numerical input parameters that are required to operate

Please cite this article in press as: AlHamaydeh M, et al. *INSPECT*: A graphical user interface software package for IDARC-2D. SoftwareX (2016), http://dx.doi.org/10.1016/j.softx.2016.10.004

IDARC-2D. This process could be extremely time-consuming and more importantly frustrating or even prohibitive and intimidating for inexperienced users. During development, this software package had undergone extensive testing and verification. It has consistently demonstrated accurate re-creation of many IDARC-2D existing input files.

Acknowledgments

The *INSPECT* developers would like to acknowledge the following individuals for their various and kind help with the *INSPECT* project: Prof. Khaled Galal, Mr. Nader Aly, Mr. Laith Elayyan, and Mr. Abdullah Sagheer. Special thanks to Prof. Andrei Reinhorn from the State University of New York (SUNY) at Buffalo, for granting the permission to redistribute the free version of IDARC-2D with the INSPECT package for additional user convenience.

References

[1] Krawinkler H. Importance of good nonlinear analysis. Structl Des Tall Spec Build 2006;15(5):515–31.
[2] AlHamaydeh M, Al-Shamsi G. Development of analytical fragility curves for representative buildings in dubai, united arab emirates. In: The international conference on earthquake engineering; skopje earthquake—50 years of European earthquake engineering, May 29–31. 2013.
[3] Al Satari MH. Estimation of seismic response demands for RC framed structures. University of Southern California; 2005.
[4] Al Satari M. Estimation of seismic response demands for R/C framed structures: An insight into the nonlinear seismic behavior. Saarbrücken, Germany: VDM Verlag; 2008.
[5] Al Satari M, Anderson J. Estimating inelastic seismic demands by elastic analysis for reinforced concrete (RC) framed structures. In: The 75th SEAOC annual convention, September 13–16; 2006, pp. 153–67.
[6] AlHamaydeh M, Abdalla J, Abdalla S, Al-Rahmani A, Mostafa A. Inelastic seismic demands for reinforced concrete frames in dubai. In: The 14th european earthquake engineering conference, Aug. 30–Sept. 3, 2010.
[7] AlHamaydeh M, Abdullah S, Hamid A, Mustapha A. Seismic design factors for RC special moment resisting frames in dubai, UAE. Earthq Eng Eng Vib 2011;10(4):495–506.
[8] Computers and Structures Inc., ETABS 2015. Computers and structures, Inc., https://www.csiamerica.com/products/etabs. Berkeley, California, USA; 2015.
[9] Computers and Structures Inc., SAP2000 ver. 18. Computers and structures, Inc., https://www.csiamerica.com/products/sap2000. Berkeley, California, USA; 2016.
[10] Computers and Structures Inc., Perform-3D – nonlinear analysis and performance assessment for 3-D structures, ver. 5.0.0. Computers and Structures, Inc., https://www.csiamerica.com/products/perform-3d. Berkeley, California, USA; 2011.
[11] ASCE/SEI, seismic evaluation and retrofit of existing buildings (ASCE/SEI 41-13 Standard). American Society of Civil Engineers (ASCE), http://dx.doi.org/10.1061/9780784412855. Reston, Virginia, USA; 2014.
[12] Elnashai AS, Papanikolaou V, Lee D. ZEUS NL – A system for inelastic analysis of structures, User's Manual, ver. 1.9.0. Mid-America Earthquake (MAE) Center, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, http://mae.cee.illinois.edu/software/software_zeusnl.html. Urbana-Champaign, Illinois, USA; 2011.
[13] SeismoSoft Inc., SeismoStruct 2016 – A computer software for static and dynamic nonlinear analysis of framed structures, ver. 7. SeismoSoft, Inc., http://www.seismosoft.com/SeismoStruct_2016_Release-4_ENG; 2016.
[14] McKenna F, Fenves G, Scott M. OpenSees – open system for earthquake engineering simulation, ver. 2.5.0. pacific earthquake engineering research center (PEER), http://opensees.berkeley.edu/. Berkeley, California, USA; 2016.
[15] Zhu M, Scott MH. Modeling fluid–structure interaction by the particle finite element method in OpenSees. Comput Struct 2014;132:12–21.
[16] Scott MH, Kidarsa A, Higgins C. Development of bridge rating applications using OpenSees and Tcl. J Comput Civ Eng, ASCE 2008; 22(4):264–71.
[17] Jiang J, Usmani A. Modeling of steel frame structures in fire using opensees. Comput Struct 2013;118:90–9.
[18] Mazzoni S. BuildingTcl/BuildingTclViewer. Pacific earthquake engineering research center (PEER), http://opensees.berkeley.edu/wiki/index.php/BuildingTcl. Berkeley, California, USA; 2010.
[19] Schellenberg A, Yang T, Kohama E. OpenSees Navigator ver. 2.5.5. Pacific earthquake engineering research center (PEER), http://openseesnavigator.berkeley.edu/. Berkeley, California, USA; 2015.
[20] Reinhorn AM, Roh H, Sivaselvan M, Kunnath SK, Valles RE, Madan A, Li C, Lobo R, Park YJ. IDARC2D Version 7.0: A program for the inelastic damage analysis of structures, User's Guide. Multidisciplinary center for earthquake engineering research (MCEER), http://civil.eng.buffalo.edu/idarc2d50/. Buffalo, New York; 2010, p. 63.
[21] Microsoft Inc., C#_Visual Studio 2013. Microsoft Inc., https://msdn.microsoft.com/en-us/library/kx37x362.aspx; 2013.
[22] SQLite Inc., SQLite ver. 3.14.1. SQLite Inc., https://www.sqlite.org/; 2016.
[23] Reinhorn AM, Roh H, Sivaselvan M, Kunnath SK, Valles RE, Madan A, Li C, Lobo R, Park YJ. IDARC2D Version 7.0: A program for the inelastic damage analysis of structures (MCEER-09-0006). Multidisciplinary center for earthquake engineering research (MCEER), http://mceer.buffalo.edu/pdf/report/09-0006.pdf, Buffalo, New York; 2009, p. 406.
[24] AlHamaydeh M, Najib M, Elayyan L. INSPECT-Lite: A GUI Pre-Processor Package for IDARC-2D Ver 7.0. In: The 2015 world congress on advances in structural engineering and mechanics, ASEM15, August 25–29; 2015.
[25] SourceForge Inc., WinMerge ver. 2.14.0. SourceForge Inc., http://winmerge.org/, 2013.