



Psyrras, N. K., & Sextos, A. G. (2018). Build-X: Expert system for seismic analysis and assessment of 3D buildings using OpenSees. *Advances in Engineering Software*, 116, 23-35.
<https://doi.org/10.1016/j.advengsoft.2017.11.007>

Peer reviewed version

License (if available):
Unspecified

Link to published version (if available):
[10.1016/j.advengsoft.2017.11.007](https://doi.org/10.1016/j.advengsoft.2017.11.007)

[Link to publication record in Explore Bristol Research](#)
PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via elsevier at <https://www.sciencedirect.com/science/article/pii/S0965997817302181>. Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/pure/about/ebr-terms>

Build-X: Expert system for seismic analysis and assessment of 3D buildings using Opensees

Nikolaos K. Psyrras¹ and Anastasios G. Sextos²

Abstract. Despite the breadth of the available finite element codes for seismic analysis and assessment, the associated complexity in use and the generality in orientation are likely to increase the epistemic uncertainty involved in the models, particularly in nonlinear analysis procedures. Thus, it is of interest to develop tools for improving the reliable use of existing structural engineering software. This paper aims to present the capabilities of Build-X, a recently developed knowledge-based system tailored to the prediction of the seismic response of 3D buildings. This expert system features a simple visual user interface that supports the structural engineer throughout the structural configuration of a building, providing expert suggestions as to critical modelling decisions, and automations that increase the reliability of the analysis and accelerate the pre-processing stage. Build-X is linked with OpenSees, a widely used script-based freeware for seismic analysis of structures, which is utilized to perform the core finite element analysis. Post-processing tasks are easy to handle through the graphical engine of the system developed. A verification study demonstrates the efficiency of the system and reliability of the results generated, pointing to the way in which Build-X may serve as a useful tool for the seismic analysis of newly designed buildings and the assessment of existing ones at reduced computational cost and modelling uncertainty.

Keywords. expert system, buildings, seismic assessment, pushover analysis, soil-structure interaction, OpenSees

¹ Ph.D. Candidate, Department of Civil Engineering, University of Bristol, United Kingdom

² Assoc. Professor, Department of Civil Engineering, University of Bristol, United Kingdom & Department of Civil Engineering, Aristotle University of Thessaloniki, Greece

INTRODUCTION

The unprecedented evolution in the field of computer science and information technology over the last decades has led to the development of numerous Finite Element software applications focusing on the numerical solution of structural and geotechnical problems. Such tools are able to offer a wide range of modeling options by boasting extensive material and element libraries, ensure algorithmic stability and solution accuracy, and some are also capable of implementing code-prescribed design procedures. By and large, nonetheless, they are characterized as general-purpose software. This implies that an average user may require considerable amounts of time to just learn their basic features and lots of frustration to master them. Modelling (or epistemic) uncertainty associated with the decisions made during pre-processing is likely to emerge as an issue of concern when the engineer seeks the response of structures under complex loading, as is earthquake ground motion. This uncertainty is further amplified by the implicit assumptions adopted by each software, often deeply hidden in long documentation files, and the case-specific FE model requirements. This denotes that analysis results are plausibly dependent on the user decisions concerning such critical modeling aspects as material constitutive laws, complex structural components, soil-structure interaction, and the selection of analysis parameters involved that may dramatically affect the response.

The need for rigorous engineering judgment becomes even more apparent in the case of seismically excited buildings, a class of problems that involve parameters of increased uncertainty. Given the multi-parametric nature of the nonlinear response of structures and the probabilistic assessment of seismic loading, even the most pertinent FE software might fail to guarantee their users will be able to represent inelastic structural response with a controlled and adequate degree of reliability. On the contrary, even the most rigorous algorithms for nonlinear structural analysis to stochastic excitation rely on the engineer to provide a reliable estimate of the several mechanical parameters, make decisions as per the boundary conditions and the effect of soil compliance and damping at the soil-foundation interface. The engineer is expected to assume own responsibility on how the software operates and interprets the assumptions made. Further, even though very advanced earthquake engineering-oriented software platforms have been developed recently (e.g. [1]), their potential effectiveness is hindered by the lack of a user-friendly interactive environment.

In the early 1990's and by virtue of the rising spread of the microcomputer, research interest moved towards the development of knowledge-based 'expert' systems that could provide supportive aid in solving specialized civil engineering problems. These novel codes, based on

119
120
121 the principles of artificial intelligence, were built to encompass domain-specific expertise,
122 convey it to unspecialized engineers-users in an interactive manner and apply it to actual
123 problem solving schemes. The area of structural design has traditionally benefitted the most
124 from this type of software, because it involves ill-structured problems by definition, where
125 heuristic knowledge is more applicable. We cite here the following notable prototype shells:
126 HI-RISE [2], SPECON (reviewed in Ref. [3]), ERDES [4]. In the subdomain of analysis, the
127 range of relevant work reported is far narrower; we identify SACON [5], an early rule-based
128 system that used backward-chaining to infer suitable analysis strategies and controlling
129 problem parameters for structural analysis problems, SesCon [6], a dedicated consultant for the
130 use of Sesame69 structural analysis program, and FEMOD [7], an assistant in FEM-specific
131 topics. Critical evaluations of expert systems applied in structural design, analysis and damage
132 and safety assessment are given in Refs. [3,8]. A comprehensive review of expert systems
133 developed to assist in the field of earthquake engineering is present in Berrais and Watson [9].
134 Dussom et al. [10] presented QUAKE, an expert tool for selecting site- and structure-specific
135 earthquake time histories. Koumoussis et al. [11] proposed a novel PROLOG-based expert tool
136 for using and better comprehending Eurocode 8 provisions. More recently, Berrais [12]
137 presented a prototype knowledge-based tool for the earthquake resistant design of reinforced
138 concrete (RC) buildings with the use of nonlinear dynamic analysis. Further, another study [13]
139 discusses the application of another three expert systems in civil engineering.

140
141 To the authors' knowledge, there has been little advancement in this research field since, and
142 no later work related to the seismic analysis of building structures. Moreover, the then
143 developed expert solutions have now become outdated considering the rapid progress in
144 information technology and in the state-of-the-art in earthquake engineering, and most
145 probably unuseable by modern computers due to incompatibility issues. In light of the current
146 state-of-practice, which requires more than ever rehabilitation of aging buildings and cost-
147 effective design against seismic forces of new ones, modern expert systems appear essential
148 to aid in conducting demanding seismic analysis of buildings in a reliable manner.

149
150 Existing conventional FEA software lack a strict internal construct to responsibly drive the user
151 throughout the process of building the structural model and specifying the seismic loading.
152 Thus, novice computer users or inexperienced structural engineers are likely to experience
153 severe difficulties and delays, particularly in cases where nonlinear behavior is examined, or,
154 even worse, end up with underestimation or overestimation of seismic response. This very gap
155 in engineering practice is attempted to be bridged by Build-X, the expert system presented
156 herein. Although the concept of expert systems is admittedly more applicable to ill-defined
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177

178
179
180 problems, as design is, efforts were made to gather reliable knowledge on numerical modeling
181 and seismic analysis methods from the state-of-the-art and state-of-practice, transform it into a
182 rule base and apply it appropriately.
183
184

185 Build-X is a front-end knowledge-based tool developed with the aim of assisting practicing
186 engineers in predicting the seismic response of 3D modeled buildings. It takes advantage of
187 the sophisticated Open System for Earthquake Engineering Simulation (OpenSees) platform to
188 provide through its graphical user interface:
189

- 190 • stepwise guidance during the pre- and post-processing stages,
- 191 • automations that accelerate the finite element model development,
- 192 • expert advice for addressing various building-specific issues that are key factors to the
193 reliable prediction of its seismic response, as will be presented in the following.
194
195
196
197

198 Build-X proceeds beyond the state-of-the-art by improving the credibility of the finite element
199 model at hand, as well as the efficiency of the analysis procedure as a whole, since it minimizes
200 the probability of modeling mistakes and cuts down on the time and effort required by the user,
201 notably in the case of multi-story buildings featuring shear wall members and compliant
202 foundation systems. Furthermore, seen as having a dual role, it can be an intelligent pre- and
203 post-processor for OpenSees, dedicated to the analysis and assessment of building structures
204 subject to earthquake effects.
205
206
207
208

209 It should be emphasized that the focus here is not on OpenSees per se. OpenSees is just the FE
210 code selected for the core problem solution; it could very well be any other script-based FE
211 solver instead. For instance, one could follow the same rationale and develop a pre/post
212 processing expert system using *.inp or Python scripting for Abaqus or APDL language for
213 ANSYS.
214
215
216

217 This text is organized as follows: first, the main features of the software are presented along
218 with the system architecture. Second, the sequence of operations is briefly described, the
219 seismic analysis methods supported by the system and a verification case study, followed by
220 conclusions.
221
222
223
224

225 SOFTWARE OVERVIEW

226 **Basic concept behind the system**

227 Build-X was developed exclusively for Microsoft Windows operating systems. Its source code
228 is written in VB.NET and was developed in the Microsoft Visual Studio environment. A
229 procedural programming language was preferred over a logic paradigm-oriented one (e.g.
230
231
232
233
234
235
236

237
238
239 PROLOG) because it fits better the needs of the software, where many mathematical
240 evaluations are to be processed and a neat GUI is to be designed. In cooperation with OpenSees,
241 it facilitates static analysis for gravity loads, Eigenvalue analysis, Modal Response Spectrum
242 analysis and Nonlinear Static (Pushover) analysis of 3D building structures. In fact, Build-X
243 operates as a real-time converter of the user's choices into a Tool Command Language (Tcl)
244 script to be used as input to the OpenSees platform. After OpenSees generates the analysis
245 output, Build-X is called back to process and present it in a visually comprehensive way.
246 Overall, the software displays the following key characteristics:
247
248

- 249 • A *wizard*-resembling sequential flow of actions that prohibits navigational disorientation
250 of the user during the modeling process. The user is encouraged to determine the
251 configuration of the structural model of the building through a series of logical and strictly
252 defined modeling steps that prevent them from skipping or missing significant aspects.
253
- 254 • *Expert knowledge* provided a priori for simulating critical components of the building in
255 a reliable manner and for selecting the most appropriate case-specific analysis parameters,
256 eliminating the probability of modeling errors.
- 257 • *Background code automations* implemented at every pre-processing step that drastically
258 reduce the time required for the Finite Element model of the building to be completely
259 defined.
- 260 • A user-friendly *visualization engine* that allows the user to inspect the Finite Element
261 model throughout its generation and review the structural response obtained by the solver.
262
263
264
265
266
267
268
269
270
271

272 **System architecture**

273 The internal software architecture differs from that of a conventional event-driven GUI
274 program. Build-X is structured according to the principles of a knowledge-based system [9],
275 consisting of seven distinct, yet interacting components, as demonstrated in Figure 1:
276
277

- 278 • *Knowledge base*: It contains a set of four elements that are called upon at specific points
279 throughout the simulation procedure to provide expert assistance based on established
280 know-how for demanding modelling and assessment tasks. The source of this knowledge
281 is well-developed seismic codes and acclaimed researchers in the field.
282
- 283 • *User interface*: It comprises the pre- and the post-processing module. The visual user
284 interface is intended to provide communication between the user and the rest of the system
285 components. Typical users are assumed to be structural engineers, office practitioners,
286 architects or engineering students with little to no expertise in seismic analysis and basic
287 computer skills. On this basis, the user interface was built to be simple and clean. No CAD
288
289
290
291
292
293
294
295

296 capabilities are present in it, but all user input essential for the definition of the structure's
297 geometry is handled by an intelligent dedicated module described in detail later on.
298
299

- 300
- 301 • *Inference engine*: Utilizing expertise from the knowledge base, the inference engine is
302 responsible for controlling the program flow and for continuously updating the context.
303
- 304 • *Context*: In essence, the context is a collection of data that accumulates in the cache in a
305 structured way according to user input and additional generated information; it represents
306 the building model at hand. The context is accessed and modified by the inference engine.
307
- 308 • *External FEA program*: Finite Element analysis is undertaken by the OpenSees platform,
309 which is linked to the system in a two-way (input-output) manner.
310
- 311 • *Explanation facility*: An essential component that provides stepwise instructions, useful
312 information and warning messages where appropriate. It is fed by the knowledge base and
313 accessed by the user through the user interface.
314
- 315 • *External MATLAB libraries*: Three MATLAB functions packaged in the form of .NET
316 assemblies are invoked by the system to address specific tasks throughout the dataflow.
317
318
319
320

321 **Flow of operations**

322 The pre-processing stage of a Build-X project consists of ten prescribed steps (Figure 2),
323 implemented as appropriately ordered window tabs. The hierarchy of the steps draws from
324 common FE modeling logic and traditional practice. In each step, a number of large-sized
325 numeric or multi-type arrays are created to store all the information describing the 3D finite
326 element model of the building, for instance node coordinates, section properties, gravitational
327 load values, etc. A brief description of the operational sequence is given below in groups:
328
329
330
331
332

- 333 1. In the first and third step, the software engine requests the description of the general
334 geometry of the building in elevation and plan (e.g. number and height of floors, number
335 and length of bays in principal plan directions), in order to produce the grid layout.
336
337
- 338 2. An intermediate step involves the determination of material behavior (linear elastic or
339 nonlinear inelastic); on this choice depend the analysis procedures to become available in
340 the last step. Reinforced concrete is the fundamental material in Build-X; however, if a
341 linear elastic material stress-strain relationship is selected, the option of a user-defined
342 material is activated as well. The constitutive material models adopted by the software for
343 unconfined and confined concrete and reinforcing steel are the ones proposed by Kent and
344 Park [14], Scott et al. [15] and Menegotto and Pinto [16], respectively. The criteria for these
345 choices were merely the reliability of the models and the simplicity in formulation.
346
347
348
349
350
351
352
353
354

- 355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
3. Next are the steps to create the actual structural members of the building. First, the user is asked to insert shear wall members, if desired; no expertise in discretization is required here, as this task is taken over by the expert system which automatically estimates the geometric and stiffness properties of the frame finite elements according to the popular Equivalent Frame Method [17]. Secondly, position and cross-section of column members of the building need be set by the user. Subsequently, the horizontal beam members are generated by defining the connectivity between the columns and thereafter their sections are selected from a predefined library. In the case that nonlinear response is sought, each of the above three modeling steps is followed by a sub-step whereby the user has to define the reinforcing steel configuration in every structural member, according to given typical layouts. Finally, position and thickness of slabs remain to be determined.
 4. To model the foundations of the building, two options are provided: homogeneous boundary conditions for the ground nodes and compliant supports. The first option allows the user to select between fixed and pinned supports, while the second one allows two commonly used foundation types: rigid spread footings and mat foundation. Both options require the user to enter the geometry and basic mechanical properties of the soil.
 5. The next modeling step is the determination of the gravity loads acting on the structure. Surface loads applied on slab members are automatically distributed to supporting members. Masses are assumed to be lumped at floor levels and are automatically calculated by the system.
 6. The last pre-processing step is related to the analysis methods and the analysis parameters to be used. In the case of linear elastic simulation, static analysis for gravity loads, Eigenvalue analysis and Modal Response Spectrum analysis are activated as options. If inelastic material behavior is turned on, the available analysis procedures are nonlinear static analysis for gravity loads, Eigenvalue analysis and Standard Pushover analysis. The option to consider geometric nonlinearities in the form of global second-order (P- Δ) effects is also present.

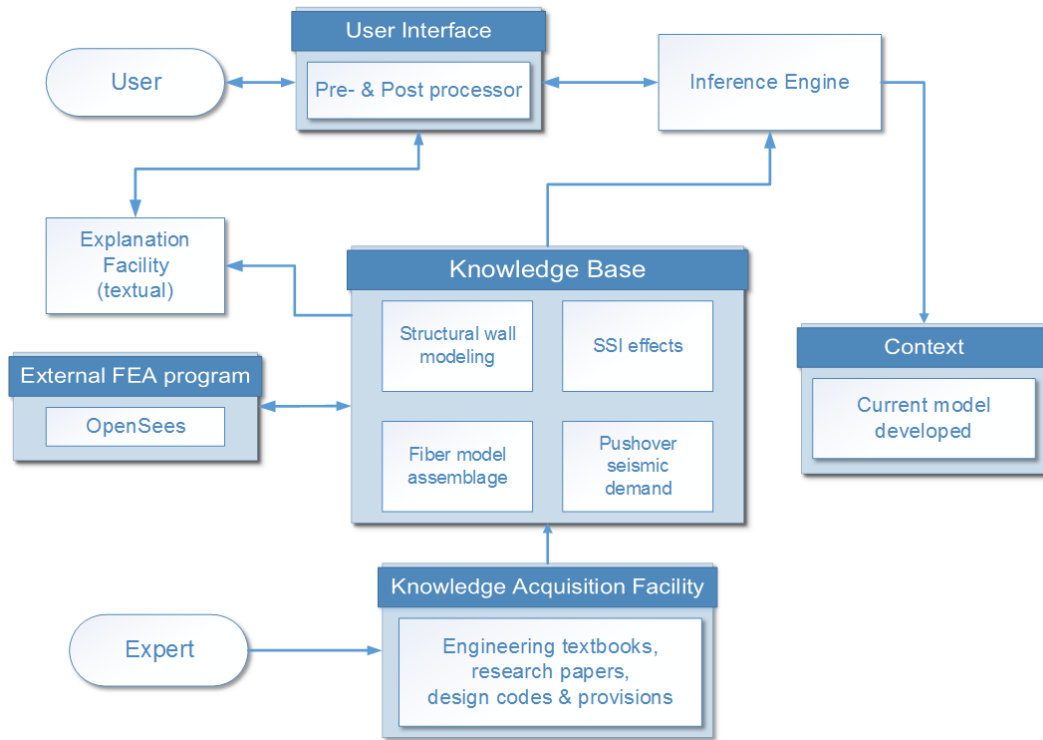


Fig. 1. System architecture

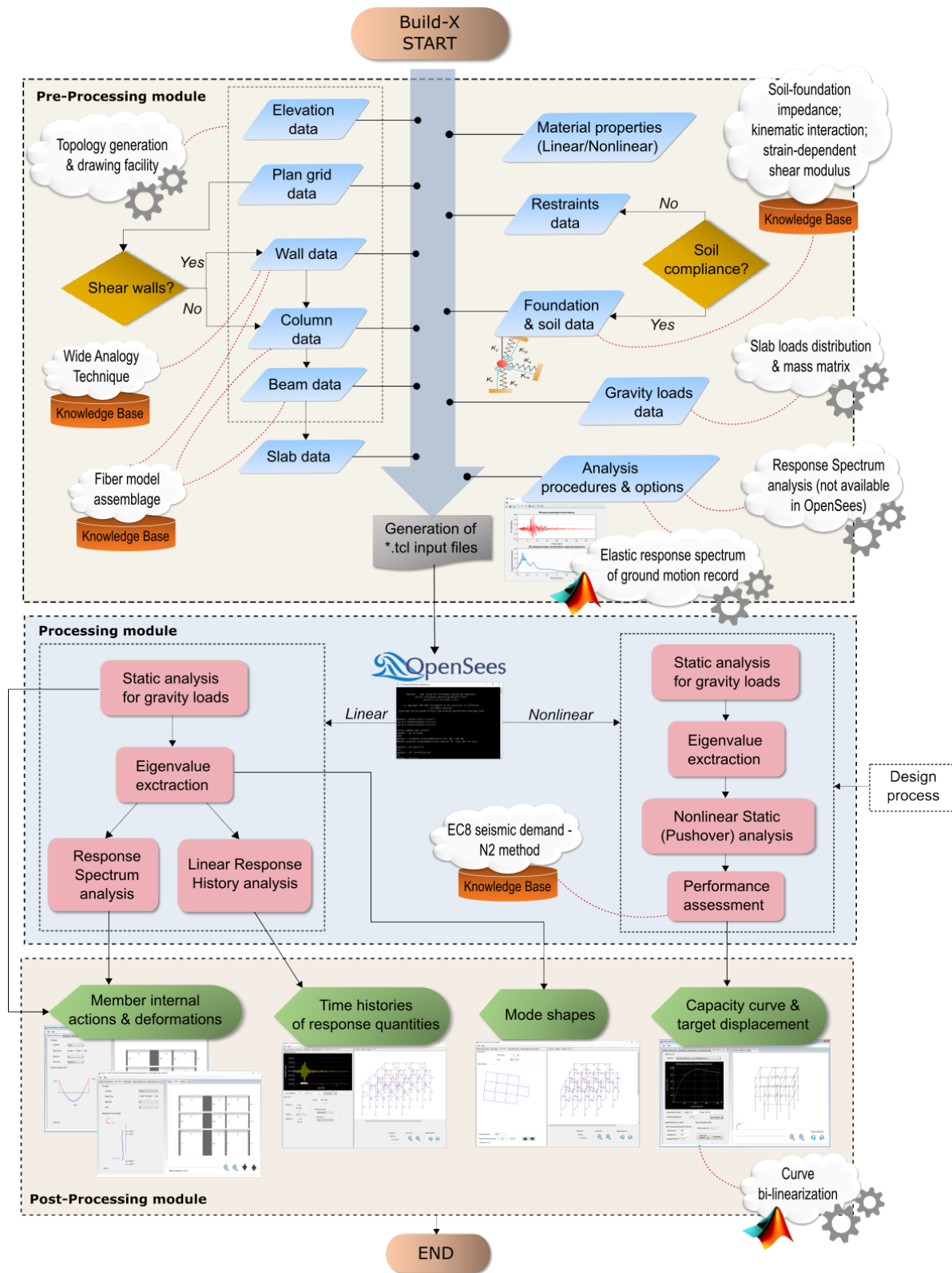


Fig. 2. Data flowchart

7. Finally, the variables stored in memory are manipulated to assemble appropriately formatted *.tcl* files that are subsequently imported to OpenSees. In the background, Build-X triggers the execution of the OpenSees command shell with these particular files as input

532
533
534 and, after all requested analyses are complete, post-processes the output files produced by
535 OpenSees and switches to the results graphical environment. Code snippets of this
536 communication are provided in Listing 1, where select OpenSees commands are created
537 (Table 1 and 2 supplements the explanation of some data variables), and Listing 2. Note
538 that all missing variable declarations are implied elsewhere in the source code. In the post-
539 processing module of the system, the user may navigate freely to review graphically
540 illustrated information regarding member and frame internal forces diagrams, deformed
541 shapes, Eigenmodes and the Capacity curve, if applicable. This module also includes the
542 evaluation of the seismic performance of the building.
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590

```

591
592
593 1 Private Sub Analyze_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
594 2     Handles Analyze.Click
595 3     SaveFileDialog1.Filter = _
596 4         "Tool Command Language File (*.tcl)|*.tcl"
597 5     SaveFileDialog1.InitialDirectory = _
598 6         My.Computer.FileSystem.SpecialDirectories.MyDocuments & "\Build-X_files"
599 7     If SaveFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
600 8         userName = IO.Path.GetFileNameWithoutExtension(SaveFileDialog1.FileName)
601 9         directoryName = Path.GetDirectoryName(SaveFileDialog1.FileName)
602 10        userName = userName.Replace(" ", "_")
603 11        file = directoryName & "\" & userName & ".tcl"
604 12        'Open text file for I/O operations
605 13        FileOpen(1, file, OpenMode.Output)
606 14        dirRec = directoryName & "\" & userName & "\recorders"
607 15        System.IO.Directory.CreateDirectory(dirRec)
608 16        Dim OpenSeesDir As String = dirRec.Replace "\", "/"
609 17        PrintLine(1, "set osdir {" & OpenSeesDir & "}")
610 18        PrintLine(1, "set shortName [file attributes $osdir -shortname]")
611 19        PrintLine(1, "cd $shortName")
612 20        'Start building OpenSees model
613 21        PrintLine(1, "model BasicBuilder -ndm 3 -ndf 6")
614 22        '...
615 23        'Generate nonlinear vertical elements (columns/walls) in the case of a Pushover analysis
616 24        For i = 1 To UBound(columns, 1)
617 25            For j = 1 To numNodes
618 26                For z = 1 To numNodes
619 27                    'Perform loop checks to find matching end nodes for each column/wall member
620 28                    If columns(i, 2) = nodes(j, 1) And columns(i, 3) = nodes(j, 2) _
621 29                        And columns(i, 5) = nodes(z, 1) And columns(i, 6) = _
622 30                            nodes(z, 2) And columns(i, 4) = nodes(j, 3) And _
623 31                                columns(i, 7) = nodes(z, 3) Then
624 32                        If Not columns(i, 0) = Nothing Then
625 33                            PrintLine(1, "element forceBeamColumn " & element & _
626 34                                " " & j & " " & z & " 5 " & 5000 _
627 35                                    + CInt(columns(i, 19)) & " 1")
628 36                        Else
629 37                            PrintLine(1, "element forceBeamColumn " & element & _
630 38                                " " & j & " " & z & " 5 " & CInt(columns(i, 19)) & " 1")
631 39                            End If
632 40                            columns(i, 15) = j
633 41                            columns(i, 16) = z
634 42                            element += 1
635 43                        End If
636 44                    Next z
637 45                Next j
638 46            Next i
639 47        '...
640 48        Threading.Thread.Sleep(1000)
641 49        'Call OpenSees
642 50        OpenSeesRun("OpenSees.exe", file)
643 51    End If
644 52 End Sub

```

Listing 1. Subroutine called on ‘Analyze’ button click; select lines of code are shown that translate internal data about columns and shear walls into TCL commands compatible with OpenSees.

Table 1. Format of the matrix ‘columns’ that stores the information about the vertical (columns/walls) members; this table shows the data type representations in each array column (1 through 21).

	0	1	2	3	4	5
--	---	---	---	---	---	---

Column or wall	ID	x_i	y_i	z_i	x_j
6	7	8	9	10	11
y_j	z_j	x-dimension	y-dimension	z-dimension	Number of transverse y stirrup legs
12	13	14	15	16	17
Number of transverse z stirrup legs	Confinement bar size	Confinement bar spacing	Longitudinal bars	Longitudinal bar size	Volumetric transverse reinforcement ratio
18	19	20	21		
Material ID	Fiber section ID	Member direction	Core wall ID (if applicable)		

Table 2. Format of the matrix ‘nodes’ that stores all node coordinates

1	2	3
x	y	z

```

671 Module OpenSeesManagement
672   'Check if application process has terminated
673   Sub CheckifClosed(name)
674     Dim allprocess
675     Dim isOn As Boolean = True
676     While isOn
677       isOn = False
678       allprocess = GetObject("winmgmts:")
679       For Each Process In allprocess.InstancesOf("Win32_process")
680         If (InStr(UCase(Process.Name), name) = 1) Then
681           isOn = True
682           Exit For
683         End If
684       Next
685     End While
686   End Sub
687   'Start OpenSees with .tcl script as input argument
688   Sub OpenSeesRun(openseesUrl, filename)
689     Dim ProcessProperties As New ProcessStartInfo
690     ProcessProperties.FileName = openseesUrl
691     ProcessProperties.Arguments = filename
692     Dim myProcess As Process = Process.Start(ProcessProperties)
693     CheckifClosed("OPENSEES.EXE")
694   End Sub
695 End Module

```

Listing 2. Module that controls the linkage between Build-X and OpenSees; control is passed to Build-X post-processing module after OpenSees has terminated its operation.

Expert knowledge highlights and code automations

Various modeling phases are facilitated by the expert knowledge library of the software. As an expert system essentially simulates a decision-making procedure normally undertaken by a highly specialized professional, Build-X carries and applies a priori a set of relevant expertise on specific modeling issues, and is equipped with a series of automation algorithms, discussed in detail in the following.

Model geometry configuration

709
710
711 For the sake of simplifying the development of the system, we decided to omit its integration
712 with a complete CAD module. Instead, the design of the floor plans is accomplished with the
713 use of a smart facility that is based on certain assumptions and restrictions, in the expense of
714 course of the unlimited freedom that CAD could allow when it comes to topology configuration.
715 These limitations are mentioned in the appropriate section of this text. The subroutines that
716 handle the generation of the model geometry make use of conventional Visual Basic user
717 interface controls to process the user's actions (e.g., DataGridView and ComboBox for user
718 input, CheckBox for possible member connectivity, Buttons for action confirmation) and of
719 the .NET built-in GDI+ API to draw graphics. The produced 2D and 3D views of the building
720 at hand are displayed alongside the main input tab.

721
722 The sequence of actions is as follows. The user is first asked to configure the global elevational
723 attributes of the building. A typical story height and a separate first story height can be set.
724 Irregularity in elevation is possible to introduce in a subsequent step. The next action is to
725 define an auxiliary $m \times n$ orthogonal grid. This grid is central in defining the general topology
726 of the building, because the system admits structural members only along these orthogonally
727 laid grid lines. Exception to this rule are core walls, which are allowed to have wall components
728 off the grid lines. Upon drawing the grid sketch, the system initiates the procedure of generating
729 load-bearing structural members upon user's request: shear walls and core walls, columns,
730 beams and slabs. This sequence partly represents the overall importance hierarchy of each
731 member type in the structure. Shear walls can be inserted arbitrarily on the grid lines, forming
732 T- or L- shaped assemblages. U- or double U-shaped walls (core walls) are introduced through
733 a separate user form. There follows the creation of columns; every possible column location is
734 identified by the system on grid intersection points that are not occupied by shear walls. The
735 available column 'slots' are indicated by checkboxes on the grid; a total of $m \times n$ columns can
736 be created. The user is initially called to specify a reference column pattern, which can be
737 customized in the next step, where columns are grouped level-wise and displayed in tabular
738 format. In this stage, they can assign column cross-sections one-by-one or discontinuities
739 (option 'Void'). Listing 3 demonstrates how the column generation and drafting works. In a
740 similar fashion, the system identifies and presents in Checkbox form all possible beam
741 members on the grid by performing compatibility checks against the previously created vertical
742 members. Subsequently, the internal subroutines are invoked again to automatically detect all
743 potential slab locations considering the existing grid of beams.
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767

This CAD substitute scheme is certainly not the most elegant one, but it vastly accelerates the topology generation process, while allowing for introduction of geometric setbacks in elevation and plan. The grid orthogonality assumption cannot be violated in any case.

```

768
769
770
771
772
773
774
775
776
777 1 'Import necessary namespaces
778 2 Imports System.Drawing.Graphics
779 3 'Subroutine to generate CheckBoxes for all column slots on XY canvas
780 4 Sub createColSlots()
781 5   For i = 1 To yGrid
782 6     For j = 1 To xGrid
783 7       colCheck(i, j) = New CheckBox
784 8       Me.TabPage1.Controls.Add(colCheck(i, j))
785 9     Next
786 10  Next
787 11 End Sub
788 12'...
789 13 Private Sub drawCols_Click(ByVal sender As System.Object,ByVal e As System.EventArgs) _
790 14   Handles drawCols.Click
791 15   'Erase all CheckBox controls on XY canvas
792 16   For Each C As Control In xydraw.Controls
793 17     If TypeOf C Is CheckBox Then
794 18       Me.TabPage1.Controls.Remove(C)
795 19     End If
796 20   Next
797 21   'Create necessary GDI+ API variables
798 22   Dim bitmap As New Bitmap(CInt(xyPicture.Width), CInt(xyPicture.Height))
799 23   Dim xyGraphics As Graphics = Graphics.FromImage(bitmap)
800 24   counter = 1
801 25   For z = 1 To numFloors
802 26     For j = 1 To xGrid
803 27       For i = 1 To yGrid
804 28         If colCheck(i, j).Checked = True Then
805 29           'Draw 12x12 column outlines in plan
806 30           xyGraphics.FillRectangle(Brushes.Silver, gridP_Yscaled(i) - 6, _
807 31             gridP_Xtrans(j) - 6, 12, 12)
808 32           'Add the relevant label
809 33           If z = 1 Then
810 34             xyGraphics.DrawString("C" & counter,Font,Brushes.RoyalBlue, _
811 35               gridP_Ytrans(i) + 6, gridP_Xtrans(j) + 6)
812 36           End If
813 37           columns(counter, 1) = "C" & counter
814 38           columns(counter, 2) = gridP_X(j)
815 39           columns(counter, 3) = gridP_Y(i)
816 40           columns(counter, 4) = gridP_Z(z)
817 41           counter += 1
818 42         End If
819 43       Next i
820 44     Next j
821 45   Next z
822 46   xyPicture.Image = bitmap
823 47   'Initialize and populate story change ComboBox with admissible entries
824 48   whatFloor = New ComboBox
825 49   For i = 1 To numFloors
826 50     whatFloor.Items.AddRange(New Object() {i})
827 51   Next
828 52   whatFloor.SelectedIndex = 0
829 53   'Assign event handling subroutine for ComboBox selection change
830 54   AddHandler whatFloorCmbx.SelectedIndexChanged, AddressOf floorChange
831 55 End Sub

```

Listing 3. VB.NET code snippet illustrating a part of the graphical facility of the system; the first subroutine is used to create the candidate columns and the second one is invoked when the user chooses the initial column configuration

Shear wall model generation

In earthquake prone regions, construction practice favours the use of shear walls towards enhancing the lateral stiffness of buildings. To avoid the excessive computational cost in representing shear walls with continuum finite elements, the Wide Column Analogy (or Equivalent Frame Method as is known alternatively) is adopted in this instance [17]. This simplified approach suggests that the planar shear wall can be substituted by an assembly of 3D frame elements that equivalently describe its mechanical behaviour: a ‘wide’ column placed at the centroid of the wall section, characterized by the actual stiffness properties of the original wall section, and two practically rigid links to satisfy Euler-Bernoulli beam theory hypotheses for the wall and restore the wall continuity with adjacent coupling beam members. To extend it to 3D space of non-planar core walls, multiple individual planar wall units are interconnected appropriately at the rigid link ends. In this case, attention has to be paid to the value of the torsional stiffness of the core wall; herein, the following formula is adopted [18].

$$J_T = h_w t_w^3 \left[\frac{1}{3} - 0.21 \frac{t_w}{h_w} \left(1 - \frac{1}{12} \frac{t_w^4}{h_w^4} \right) \right] \quad \text{for rigid beam elements} \quad (1)$$

$$J_T = 0 \quad \text{for the wide column} \quad (2)$$

To ensure the reliable implementation of the methodology, the user is only asked to define the position and the cross-sectional dimensions of the wall. The system’s internal algorithms are responsible for extracting the joint topology and element stiffness properties automatically. The option of modeling U-shaped wall cores is also available. Based on previous studies [19], the adopted technique can be deemed suitable for modeling structural walls of rectangular and U-shaped section in ordinary, low to medium-rise buildings, achieving a fair compromise between accuracy of results and computational cost.

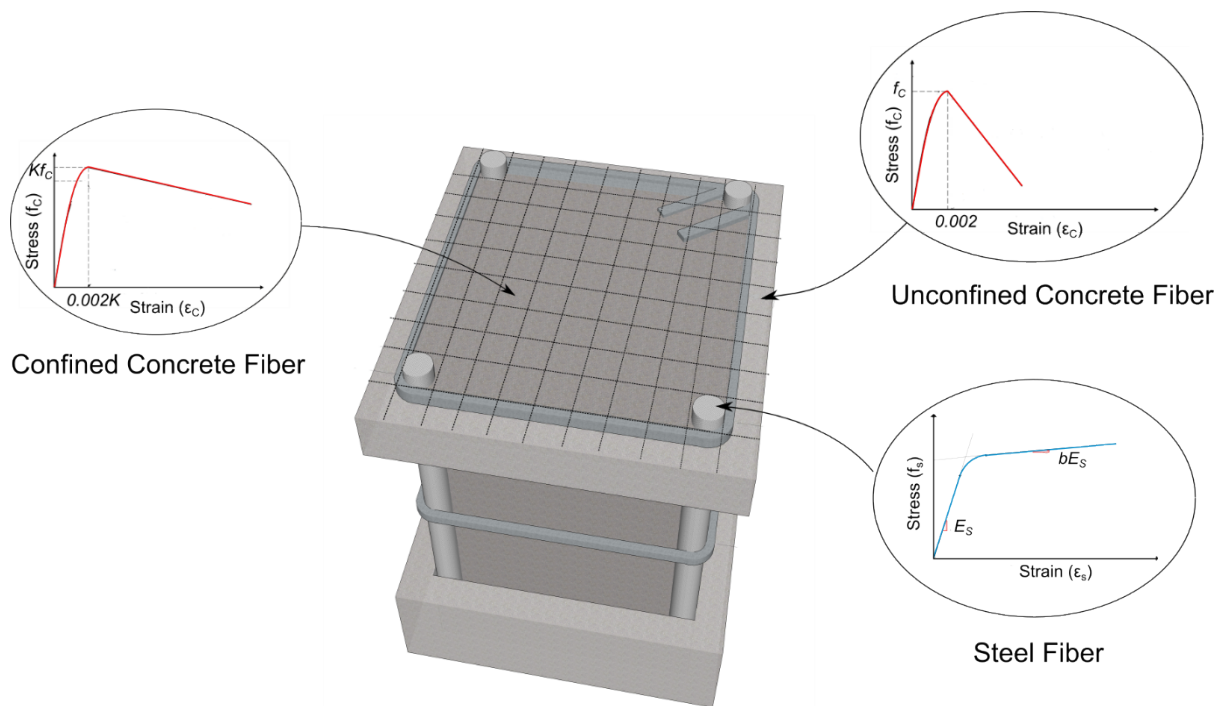


Fig. 3. RC column discretized in fibers according to the fiber section model

Fiber model development

The inelastic deformations of the RC structural members of the building can be estimated by means of a Standard Pushover analysis, utilizing the force-based distributed plasticity beam-column element [20] and fiber section implemented by OpenSees. The fiber beam-column model has the advantage of permitting gradual spread of the plastic strains both across the section depth and over the member length, over the more idealized lumped model that assumes the development of localized plastic hinges at predefined locations along the member primary axis. RC sections are particularly fit for this methodology, because they are non-homogeneous and thus nonlinearity is more conveniently described at the material level. Additionally, coupling of axial force and moment response is directly accounted for. Lumped plasticity models can be computationally more efficient as they generally require fewer input parameters. However, these parameters are not always straightforward to determine and usually need calibration.

The main benefit for the designer is that a complete description of the post-yield characteristics of the structural members is possible with the use of only intuitive and easy to determine input. The visual interface of the system is specifically designed to this direction. The user is required to define the uniaxial stress-strain relationship of groups of fibers-materials and the detailing of each member according to standard layouts provided, i.e. number and size of longitudinal bars, number, size and spacing of confinement bars, section clearance etc. Based on this input,

945
946
947 the expert system automatically and reliably creates the fiber element model for every structural
948 member, and this prevents the user from messing with intricate calibrations that depend on
949 special information. The Gauss-Lobatto integration scheme is used for the derivation of the
950 element stiffness matrices, with 5 integration points for columns and 4 for beams.
951

952 *Soil-structure interaction*

953
954
955 As shown by Mylonakis and Gazetas [21], site conditions may substantially affect the seismic
956 response of a structure. Buildings founded on deformable soil have a longer fundamental period
957 of vibration than the respective fixed-based ones and, as a result, may respond differently to
958 earthquake ground motion depending on the dynamic interplay between the frequency content
959 of the soil, the structure and the ground motion. For the case of ordinary buildings, which cover
960 the vast majority of day-to-day design practice, seismic codes worldwide typically prescribe
961 conventional analysis methods, such as Lateral Load (equivalent static) analysis and Response
962 Spectrum analysis. In both cases, damping and the frequency-dependence of soil-structure
963 interaction are irrelevant. The same applies to the Nonlinear Static (Pushover) analysis, which
964 is the prevalent method for the assessment of existing structures. For the above reasons, Build-
965 X focuses on static and equivalent linear modelling of SSI effects.
966

967
968
969 In light of this approach, three frequently preferred in design practice soil-foundation systems
970 are available: surface spread footings, embedded spread footings and mat foundation (Figure
971 4). The inertial part of the interaction is implemented through the evaluation of springs
972 connecting the ground nodes to the soil medium. The modeling procedure involves computing
973 the six static impedance components for each degree of freedom according to the formulae
974 proposed by Gazetas [22]. In the case of a mat foundation, the base slab is assumed to follow
975 a rigid body motion implemented implicitly by enforcing appropriate constraints for the base
976 nodes. Thus, it is sufficient to calculate and assign springs only at the master node representing
977 the center of mass of the base slab. The designer then defines the geometry of each individual
978 spread footing or of the basemat, i.e. plan dimensions and depth of embedment, if applicable,
979 in addition to a pair of mechanical parameters characterizing the soil halfspace (small strain
980 shear modulus and Poisson's ratio). This completed, the expert system computes the stiffness
981 constants of all the translational and rotational springs (Table 3).
982

983
984
985 For the quantification of the influence of the kinematic effects, Build-X incorporates the
986 simplified procedure proposed by FEMA 440 [23], which is based on the work by Kim and
987 Stewart [24].
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003

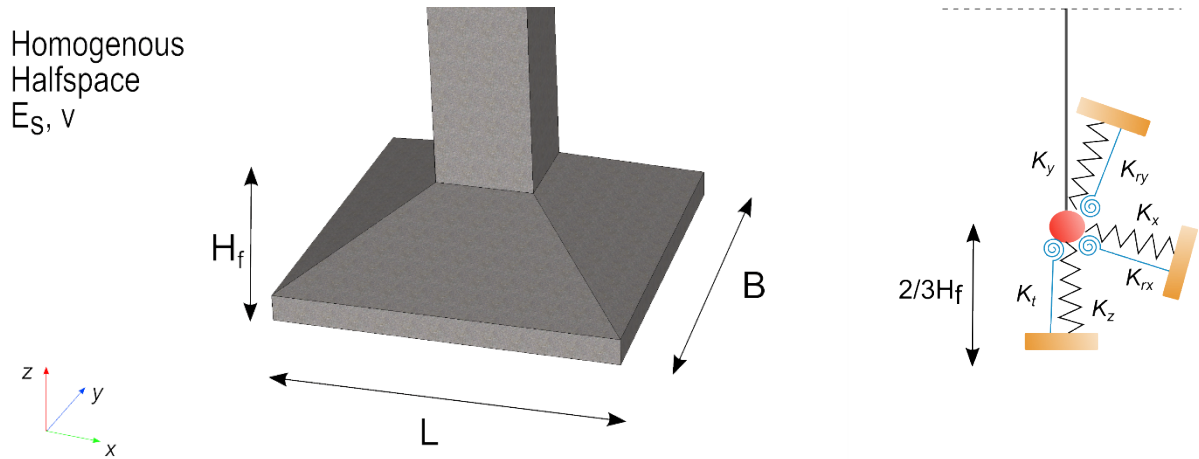


Fig. 4. Embedded spread footing and the respective 3-D spring model at the soil-footing interface

Table 3. Static stiffness components of rectangular shaped foundation on homogenous halfspace surface after Gazetas

Spring DOF	Static stiffness
Vertical z	$K_z = \frac{GL}{1-\nu} (0.73 + 1.54\chi^{0.75})$ with $\chi = \frac{A_b}{4L^2}$
Horizontal y	$K_y = \frac{GL}{2-\nu} (2 + 2.5\chi^{0.85})$
Horizontal x	$K_x = K_y - \frac{0.2}{0.75-\nu} GL \left(1 - \frac{B}{L}\right)$
Rotational about x	$K_{rx} = \frac{G}{1-\nu} I_{bx}^{0.75} \left(\frac{L}{B}\right)^{0.25} \left(2.4 + 0.5\frac{B}{L}\right)$
Rotational about y	$K_{ry} = \frac{G}{1-\nu} I_{by}^{0.75} \left[3\left(\frac{L}{B}\right)^{0.15}\right]$
Torsional	$K_T = GJ_T^{0.75} \left[4 + 11\left(1 - \frac{B}{L}\right)^{10}\right]$

According to it, kinematic interaction may be taken into account by means of the ratio of the acceleration response spectral ordinates corresponding to the foundation level to the response spectral ordinates specified for the free-field conditions. This ratio is used to estimate a reduced response spectrum that is consistent with the foundation input motion ('FIM') calculated analytically with the use of transfer functions [25]. In FEMA 440, different expressions are proposed for considering base-slab averaging and embedment effects separately. In the code

presented herein, only the former effects can be accounted for and the respective ratio is estimated from the following expression:

$$RRS_{bsa} = 1 - \frac{1}{14100} \left(\frac{b_e / 0.3048}{T} \right)^{1.2} \geq \text{the value for } T = 0.2 \text{ sec} \quad (3)$$

where $b_e = \sqrt{BL}$ with B, L standing for the foundation dimensions in plan view and T is the predominant eigenperiod, elongated due to inertial interaction effects. When performing a performance-based analysis, such as a pushover analysis [26], the ratio of response spectra due to base-slab averaging effects is evaluated using the fundamental vibrational period in the corresponding direction of excitation and then the acceleration demand is obtained by multiplying this ratio with the free-field response spectral ordinate that corresponds to the foregoing period value. Base-slab averaging is expected to emerge as a form of kinematic interaction when a foundation system with sufficient in-plane stiffness is modelled or rigid diaphragms are assumed at floor levels [23]. In Build-X, these conditions might occur: the former in the case of a RC mat foundation, while the latter in the case of independent, non-interconnected spread footings. The previously described methodology is believed to be a rational modelling approach when the seismic performance of a building on compliant soil needs to be evaluated.

Another important modeling decision concerns the dependence of the soil stiffness on the expected strain levels. Under the design seismic action, the supporting soil is expected to exhibit nonlinear inelastic behavior. This can be approximated by a reduction in the initial (low-strain) soil shear modulus G_{\max} as a function of the selected level of PGA. To tackle the error introduced by completely ignoring this effect, Build-X makes use of the following curve-fitting function [27], which has been derived to fit the relevant values proposed by Eurocode 8 - Part 4 [28]. This function gives an approximation of the shear modulus reduction factor of the soil in the cases of Modal Response Spectrum analysis and Standard Pushover analysis:

$$G / G_{\max} = 41.6a_g^3 - 17.5a_g^2 - 0.66a_g + 1, \quad 0.1g < a_g < 0.3g \quad (4)$$

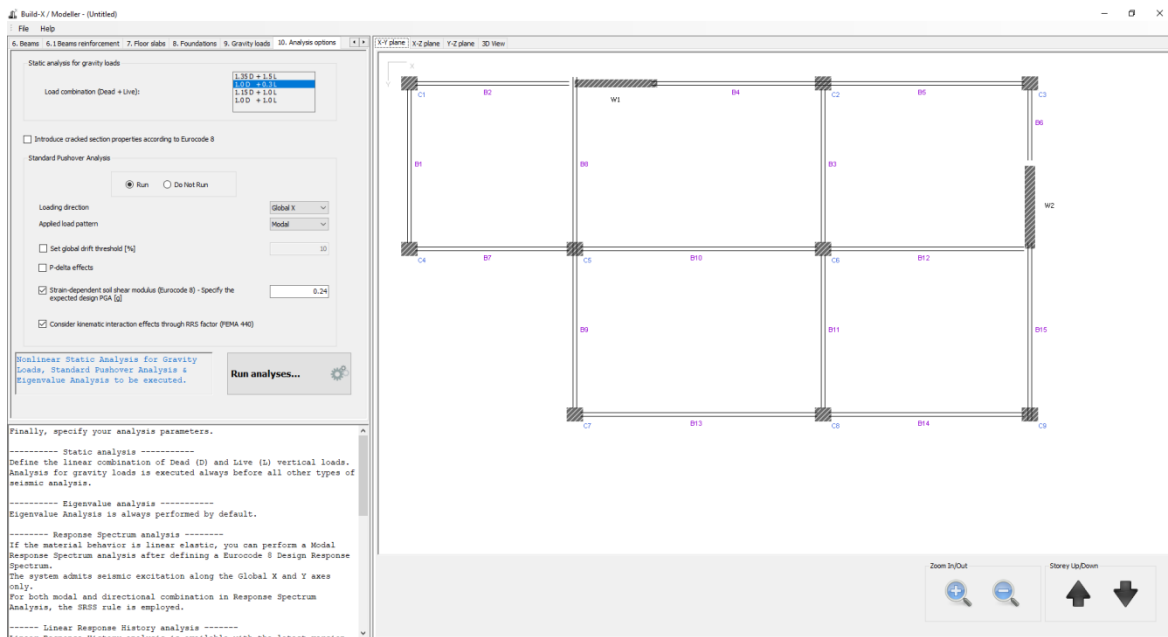


Fig. 5. Options to introduce kinematic interaction effects and strain-dependent soil shear modulus

where G_{max} is the initial soil shear modulus. The options to account for kinematic interaction effects and strain-dependent soil shear modulus are illustrated in Figure 5.

Allocation of gravitational loads and mass matrix generation

By selecting the automatic mode to introduce gravity loads on the building, the system computes the dead loads acting on frame members. Since slabs are not explicitly modeled with shell finite elements, rather rigid diaphragms are assumed at the floor level, a well-known approximate methodology is employed to allocate the surface gravity loads imposed on slabs to the supporting beams and shear walls. Based on slab dimensions and support conditions, tributary lines are drawn on the slab surface dividing it into triangular and trapezoidal sections; these are in effect the influence load areas for the underlying structural elements of the slab under consideration (Figure 6). The polygonal distributions along the slab boundaries are automatically substituted by equivalent uniform ones that yield identical shear forces. Besides the self-weight loads, the user has the option to introduce additional dead and live loads on the slabs. In the automatic mode, Build-X is also capable of considering the self-weight of non-structural infill walls resting on beam members.

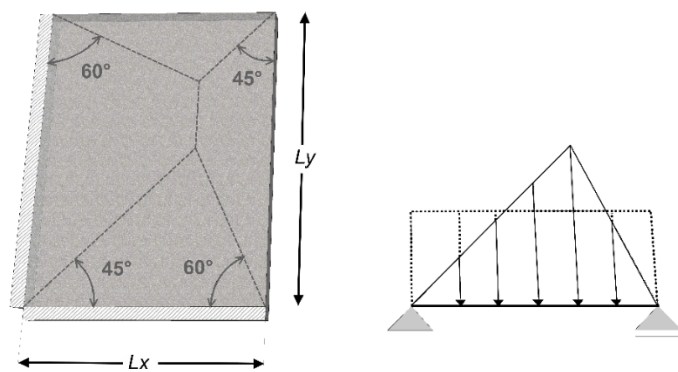


Fig. 6. Definition of influence areas on a typical two-way slab with two simple side supports and equivalent uniform distribution on the boundary

The mass matrix of the structure is assembled by the program's algorithms, without any interference by the user. Lumped masses are idealized at the geometric centroids of floor levels, assuming mass is uniformly distributed over the floor surface. Their values are estimated according to the Eurocode 8 load combination for quasi-permanent actions $G + 0.3Q$, with G being the dead and Q the live loads acting vertically on the floors [29].

Seismic performance assessment using the N2 Method

One of the most important aspects featured in Build-X is that it permits the projection of the performance level achieved by the building in a given seismic scenario. The procedure is implemented as specified in Annex B of Eurocode 8-part 1 and relies on the N2 method proposed by Fajfar [30]. Standard Pushover Analysis (SPA) is a modern variation of the classical 'collapse' analysis [31] that predicts the hierarchy of structural damage up to the onset of collapse. The interested reader is referred to the pertinent references for more information about the the analysis method and the assessment procedure prescribed by Eurocode 8.

Of note, the bilinear approximation of the capacity curve of the equivalent SDOF required by the EC8 seismic assessment procedure in terms of an elastic-ideally plastic curve is achieved completely automatically with the use of a Microsoft .NET -compatible MATLAB library, compiled with MATLAB Compiler SDK. The generated .NET assembly includes a class with one method, i.e., an interface function that performs the least-squares bilinearization. This function makes explicit use of the 'Free-knot spline approximation' function developed by Bruno Luong, available from the MATLAB online file repository. A VB code excerpt of this functionality is presented in Listing 4 and the deployed MATLAB function is displayed in Listing 5.

```

1240
1241
1242 1 'Import useful namespaces/elements
1243 2 Imports BilinLib
1244 3 Imports MathWorks.MATLAB.NET.Arrays
1245 4 Imports MathWorks.MATLAB.NET.Utility
1246 5 'Declare necessary MATLAB API array class variables
1247 6 Dim uMW, fMW As MWNumericArray
1248 7 Dim outArray() As MWArray
1249 8 'Subroutine invoked to perform the bilinearization
1250 9 Public Sub DoBilinearFit()
1251 10     'Cast native .NET variables storing the transformed SDOF
1252 11     'displacement and force to MATLAB array variables
1253 12     uMW = CType(uSDOF, MWNumericArray)
1254 13     fMW = CType(fSDOF, MWNumericArray)
1255 14     'Create instance of MATLAB library .NET class
1256 15     Dim bilinear As New bilinClass
1257 16     'bilinearization: the MATLAB function name
1258 17     outArray = bilinear.bilinearization(2, uMW, fMW)
1259 18     'assign bilinear function coefficients to array
1260 19     coeffs = outArray(0).ToArray
1261 20     'assign bilinear function breakpoints to array
1262 21     brakPnts = outArray(1).ToArray
1263 22     'Slope of the ascending branch; yield force
1264 23     slope = coeffs(0, 0): Fy_sdof = coeffs(1, 1)
1265 24     'Yield displacement; ultimate displacement
1266 25     Dy_SDOF = brakPnts(0, 1): Dult_SDOF = brakPnts(0, 2)
1267 26     'Compute equivalent SDOF period and display it in TextBox
1268 27     Tsdof = 2 * Math.PI * Math.Sqrt(mSDOF * Dy_SDOF / Fy_sdof)
1269 28     TextBox.Text = Format(Tsdof, "#.###")
1270 29 End Sub
1271 30 'Subroutine that draws the bilinear fit to the existing Capacity curve
1272 31 Public Sub PlotBilinFit()
1273 32     With pushoverCurve.Series(1).Points
1274 33         .AddXY(0.0, 0.0)
1275 34         .AddXY(Dy_SDOF, Fy_sdof)
1276 35         .AddXY(Dult_SDOF, Fy_sdof)
1277 36     End With
1278 37 End Sub

```

Listing 4. VB.NET subroutines computing and plotting the bilinear fit on the Pushover curve.

```

1271 1 function [coeff, breakPnts] = bilinearization (u, f)
1272 2 % INPUT args:
1273 3 % - u: displacement array
1274 4 % - f: force function of u to be fitted
1275 5 % OUTPUT:
1276 6 % - coeff: matrix array containing the first-order polynomial coefficients
1277 7 %         of the two linear branches
1278 8 % - breakPnts: vector array containing the breakpoints of the bilinear fit
1279 9
1280 10 % BSKF function settings for elastic-perfectly plastic approximation:
1281 11 nknots = 2;
1282 12 % Enforce point-wise constraints on the fitting curve:
1283 13 pointConstraints(1) = struct('p', 0, 'x', 0., 'v', 0.);
1284 14 pointConstraints(2) = struct('p', 1, 'x', max(u), 'v', 0.);
1285 15 % Structure to consider point-wise constraints
1286 16 options = struct('animation', 0, 'knotremoval', 'none', 'pntcon', pointConstraints);
1287 17 % Invoke actual curve-fitting function (for more info on input args,
1288 18 % please refer to the dedicated website)
1289 19 pp = BSKF(u, f, 2, nknots, [], options);
1290 20 coeff = pp.coefs;
1291 21 breakPnts = pp.breaks;
1292 22 end

```

Listing 5. The MATLAB function invoked from within the VB.NET code

Figures 7 through 9 illustrate characteristic stages during the pre- and post-processing modules of a typical inelastic simulation of a building model featuring two shear walls, resting on deformable soil and analyzed by means of a Pushover analysis. In accord with Eurocode 8

directions, Build-X offers two options for the vertical distribution of the applied lateral loads in each horizontal direction: a uniform pattern, implying a distribution proportional to floor masses, and a modal pattern, suggesting lateral forces proportional to the product of floor mass and the displacement shape vector of the dominant mode in the considered direction.

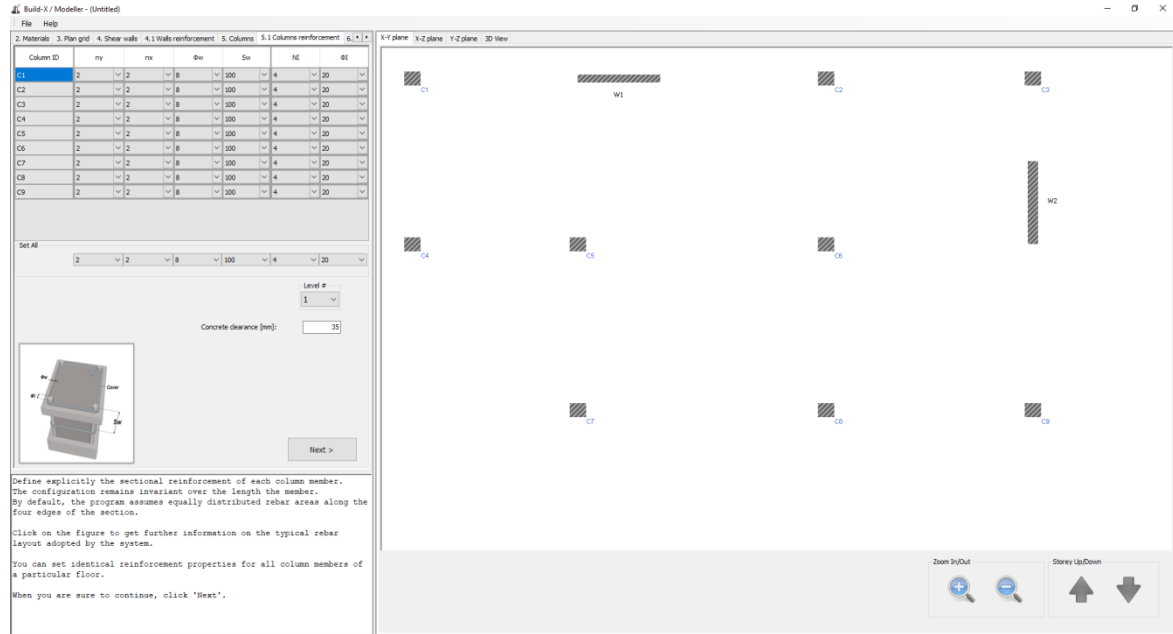


Fig. 7. View of the columns reinforcement definition tab in the pre-processing module of the system.

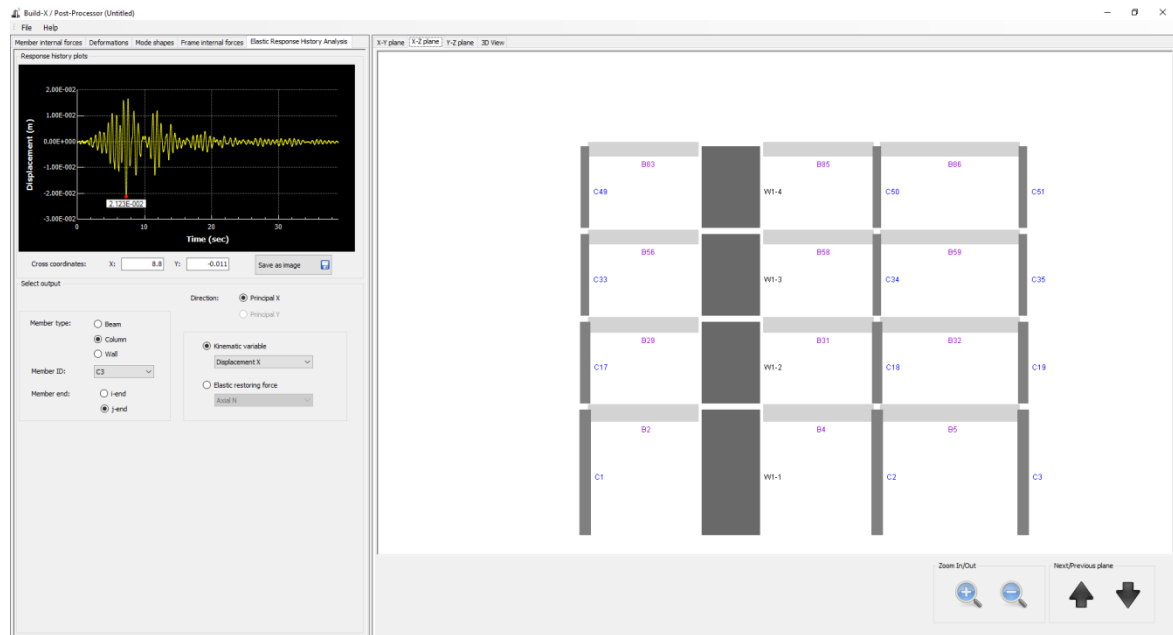


Fig. 8. View of the LRHA results tab in the post-processing module of the software.

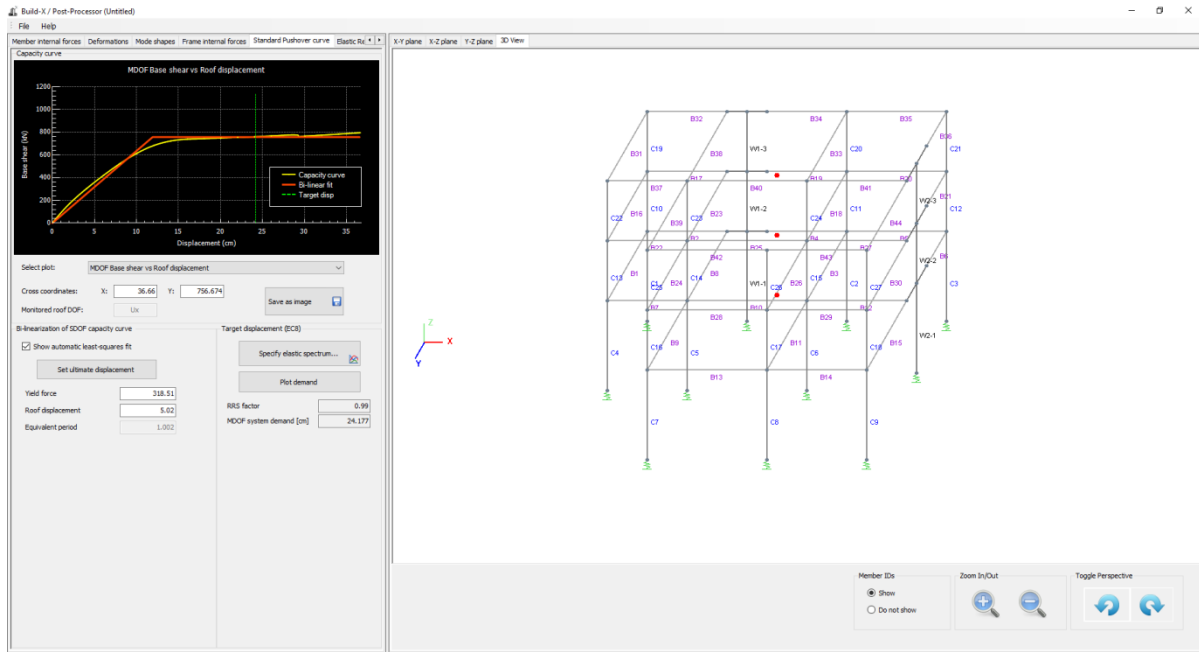


Fig. 9. View of the Capacity Curve tab in the post-processing module of the software.

The analysis proceeds until one of the following scenarios is reached: (a) the structural system becomes unstable, indicating collapse; (b) the monitored displacement reaches a user-defined threshold magnitude, (c) numerical convergence issues arise. Prior to the application of the lateral load pattern, the nonlinear response of the building due to gradually imposed gravity loading in a static manner is computed.

In addition to Standard Pushover Analysis, the option of EC8-compliant Response Spectrum Analysis is provided but it is not elaborated herein as it has been traditionally used by the engineering community for decades and hence the implementation is deemed well-known.

Overall, the expert structure of Build-X ensures the implementation of the foregoing seismic analysis procedures in a reliable and code-compliant way. Without doubt, this very software structure introduces several limitations to the user, rendering the range of simulation options and parameterization narrower compared to other finite element analysis packages. However, no margin for errors pertinent to analysis parameters is left to the user, while the system is capable of providing a realistic enough portrayal of the earthquake response of the building, employing both force-based and performance-based methods.

1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475

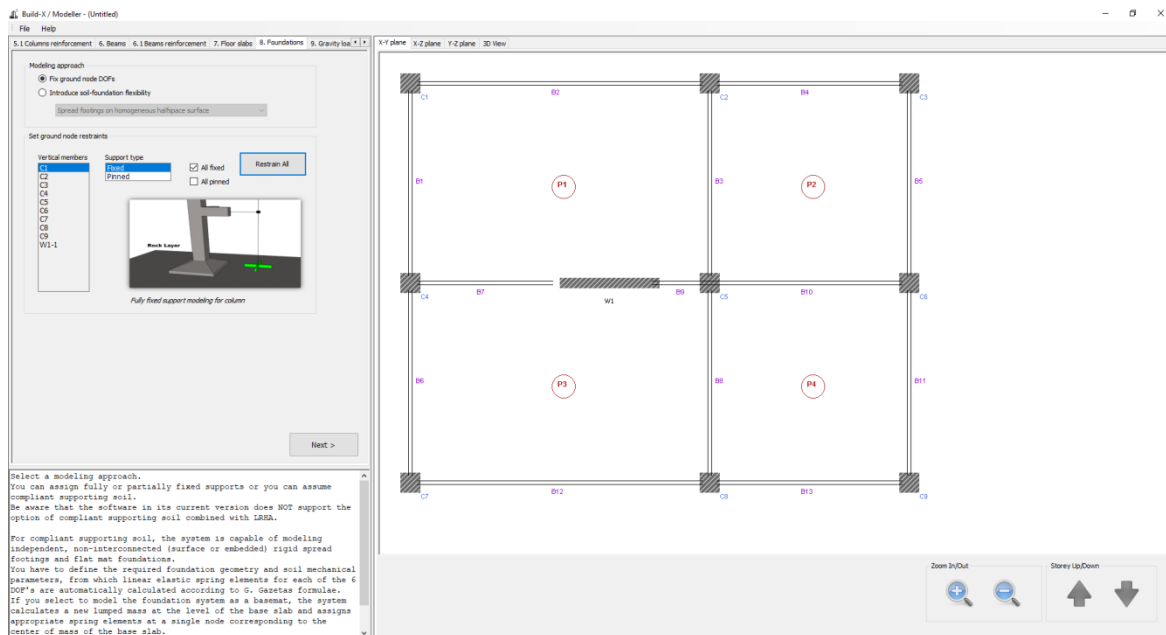


Fig. 10. Plan view of the first storey of the case study building in Build-X.

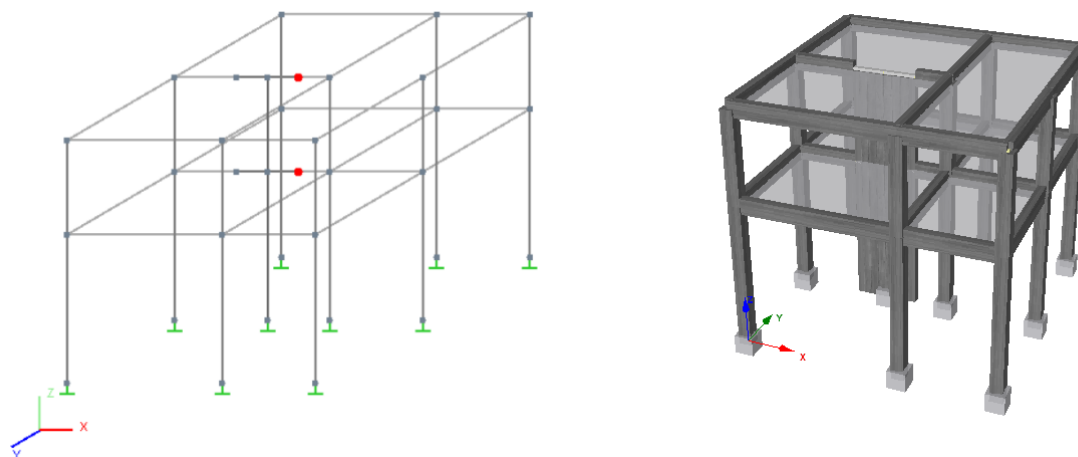


Fig. 11. 3D finite element model of the case study building as displayed in the graphical environments of Build-X (on the left) and SeismoStruct (on the right).

VERIFICATION OF THE SOFTWARE

To assess the fidelity of the building model produced by the developed expert system and to illustrate its overall efficiency and assumption validity, an example seismic performance assessment application is presented. The objective here is to show whether Build-X, as an OpenSees front-end, produces the same analysis results as the standard manual scripting approach that one would conventionally use, under the same modelling assumptions. Verification is performed against SeismoStruct [32], a widely used code for earthquake engineering applications within the research community and by practitioners. SeismoStruct was selected for the comparison on the grounds that it features distributed plasticity fiber

1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534

element models for capturing the nonlinear response of structural systems, similarly to OpenSees, hence a comparison under like modeling assumptions is possible. The structure under consideration is a two-story RC building exhibiting regularity in plan and elevation. It is first subjected to incremental gravity loading; in a second loading step, Standard Pushover analysis in both principal plan directions is performed.

Structural configuration

The building of interest is of rectangular plan, symmetric with respect to the global X axis, and has a total floor area of 64.0 m² (Figure 10). The heights of the first and second story are 4.73 m and 3.00 m, respectively. It features a dual frame-wall resisting structural system, consisting of a single shear wall (2000x250mm) with its stiff axis lying parallel to the global X axis. Rectangular cross-sections are used for columns (400x400mm) and beams (250x500mm). The concrete class is C20/25 (characteristic unconfined compressive strength $f_{ck} = 20N / mm^2$) and the reinforcing steel class is B500C (yield strength $f_y = 500N / mm^2$). It should be mentioned that SeismoStruct's material library does not contain the Kent and Park concrete model, as adopted by default in Build-X; to this end, Mander's concrete model [33] was used. Plastic strain evolution of reinforcing steel is represented by the Menegotto and Pinto model in both programs. Strain hardening is assumed zero. Reinforcement details are as follows: 6Ø20mm bars for the longitudinal reinforcement and Ø8/100mm confinement bars for the transverse reinforcement of the confined edges of the wall sections, Ø8/150mm web grid along the width edges of the core; 4Ø20mm longitudinal bars and Ø8/100mm confinement bars for the columns; 4Ø20mm longitudinal bars for the beams. Concrete cover depth was assumed to be 35mm. To ensure maximum possible matching between the parameters of the two models, vertical members were assumed to be fixed at their base and vertical loads were introduced to the building in the form of uniformly distributed transverse forces on beam members, with a magnitude of 30 kN/m. A 3D render of the building model as generated in both programs is given in Figure 11.

Comparison of Standard Pushover analysis results

By performing Standard Pushover analysis in each of the horizontal directions, the nonlinear response of the building is obtained in terms of a roof displacement versus base shear force curve, as shown in Figure 12. In general, it is observed that a high level of agreement is attained as to the predicted response quantities of interest and the overall shape of the curves between the two solutions.

1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593

Regarding the capacity curve for load application in X direction, initial gradients of the two solutions, denoting identical initial stiffnesses. The first concrete tensile cracks can be traced at approximately the same point of the two curves ($d_{cr}^{BW} \approx 0.02m$, $V_{cr}^{BW} \approx 820kN$ and $d_{cr}^{SS} \approx 0.02m$, $V_{cr}^{SS} \approx 833kN$). As the building approaches the first yield region, more striking differences between the two curves are observed. Yield strength in Build-X exceeds by 7.4% the one in SeismoStruct (1265kN compared to 1178kN), while yield displacement is found to be 19.6% larger (0.061m compared to 0.051m). Stiffness degradation is captured almost identically in terms of the slope of the post-yield curve branch. The deviation found in ultimate strength is of the order of 2.1% (1336kN compared to 1308kN). Strength softening is successfully captured by both solutions, although SeismoStruct's solution produces a sudden strength drop. The ductility capacity $\mu = d_u / d_y$ derived from Build-X is estimated to be 3.56 compared to the value of 2.83 obtained by SeismoStruct, indicating a 25% discrepancy.

In the case of Pushover analysis in Y direction, a comparison is clearly easier to derive (Figure 12, right). The two curves follow an almost identical path until the building enters the plastic region. The latter is reflected on the initial stiffness, first crack displacement, yield strength and yield displacement values, with deviations that did not exceed 2.44%. Within the plastic region, the curve obtained by Build-X maintains an almost zero tangent stiffness, while the one obtained by SeismoStruct displays a negative slope, reflecting the strength softening effect. Ultimate strength and displacement deviations between the two solutions amount to 2.16% and 4.16%, respectively. Ductility capacity is calculated to be $\mu = 7.075$ for the Build-X solution and $\mu = 6.99$ for the SeismoStruct solution, a 1.2% difference. It has to be noted that a series of convergence difficulties were encountered during the analysis in SeismoStruct, resulting in capacity curves with acute jaggedness. For this reason, tuning of some parameters concerning the fiber model was necessary. Specifically, the number of fibers used for the cross-section of vertical members was set to 81 and the number of integration sections was set to 4. These settings for beam members were 64 and 3, respectively. Additionally, the load pattern was applied in 75 steps up to a global drift equal to 10%.

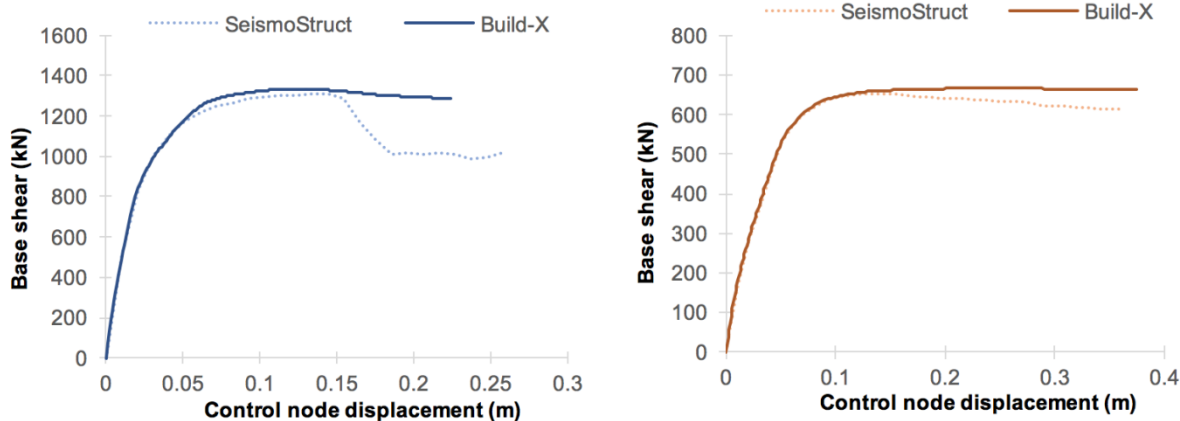


Fig. 12. Pushover curves of the considered building in global X direction (left) and Y direction (right) using the developed (Build-X) and the reference (SeismoStruct) software.

LIMITATIONS

For the sake of completeness, it is appropriate to state the most notable limitations and assumptions inherent to the software’s functionality.

- The system assumes geometrically prismatic frame elements. This means that a frame element with variable cross-section is not admissible.
- Only rectangular and circular cross-sections are available for frame members, on the grounds that these constitute the most popular choice by far in RC buildings. The future resolution is to enrich the section library with standard steel profiles, in order to expand the applicability of the software to structural steel buildings.
- The already mentioned grid orthogonality excludes the creation of obliquely-oriented straight members or curved members. Moreover, no structural members may lie outside the grid outline.
- Geometry definition is based on member centerlines. Build-X does not admit beam-column joint offsets.
- Shear walls are assumed to run uninterrupted from the foundations up to the top level of the building. In other words, no discontinuities can be introduced for this member type.
- Beam-to-beam connections are not supported.
- The Equivalent Frame Model may underestimate the torsional stiffness of the core walls, and parasitic moments due to torsion-induced shear may arise. For this reason, attention has to be paid to torsionally sensitive buildings having core walls.

- 1653
1654
1655
1656
1657
1658
1659
1660
- The user should be aware that Build-X does not provide unlimited potential for arbitrary input manipulation (going back and forth within the visual user interface). Most of the choices, once confirmed, are considered final.

1661 CONCLUSIONS AND FUTURE WORK

1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689

Capabilities and key features of an expert system designed for the seismic analysis and performance evaluation of buildings are presented in this paper. Offering an expert knowledge library which addresses modeling of critical building-specific issues designers often neglect or are unaware of, Build-X eliminates possible mistakes and ensures robustness in the modeling process. Code automations are designed to facilitate various pre- and post-processing tasks, not only improving the credibility of the finite element model developed, but also minimizing the total elapsed time before analysis results become available. This became particularly evident in the verification study: the time required to completely assemble and analyze a fairly simple inelastic building model in SeismoStruct was measured over 25 minutes; Build-X was over 8 times faster (approximately 3 minutes). It is expected that this time reduction will be significantly greater when compared to FE software that employ lumped plasticity models, whereby users have to engage in the time-consuming task of defining plastic hinge parameters. A verification case study against the reference software for a two-story RC building with one shear wall shows that no algorithmic error exists as per the use of the proposed software in preparing, calling and executing the OpenSees engine. Minor deviations between the two solutions may be partly attributed to the different constitutive laws used for confined concrete by the FE solvers.

1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702

The previous remarks, in conjunction with the fact that Build-X provides a simple and user-friendly visual interface with continuous guidance, manifest the overall improved efficiency of the system. Serving as a unique complementary tool to the GUI-lacking yet powerful OpenSees platform, Build-X has the potential to facilitate the design engineer in readily implementing seismic analysis and assessment procedures for 3D buildings, ensuring low modeling uncertainty levels and operating times. The system may prove an appealing solution especially for the stage of the preliminary seismic design, whereby different conceptual model designs are cyclically tested in order to choose the one that satisfies certain constraints.

1703
1704
1705
1706
1707
1708
1709
1710
1711

The system architecture is such that easily allows for new feature additions and improvements that could elevate its expert character. For instance, an upgrade that is currently under consideration is an implementation of a simplified global collapse capacity assessment methodology [34] for planar frame structures based on Pushover analysis. Other features

1712
1713
1714 currently being designed to complement the dynamic procedure are: (a) selection of appropriate
1715 frequency-dependent dynamic impedance functions from the literature to account for soil-
1716 foundation flexibility, (b) optimal selection of ground motion sets following spectral-matching
1717 procedures (this could be instantly accomplished by linking Build-X with ISSARS [35], a
1718 ground-motion selection tool that improves the reliability of statistical measures of the obtained
1719 structural response), (c) expansion of the LRHA framework to model nonlinear material
1720 behaviour where optimal choices of cyclic nonlinear material laws will be supported.

1721 The setup files of the software are free to download from <https://www.buildx4opensees.eu/>
1722
1723

1724 ACKNOWLEDGEMENTS

1725
1726 Special credits deserve to be attributed to D. Panagiotopoulos and N. Gantsidis, whose
1727 preliminary work on the development of *Building Wizard* served as the firm basis to this
1728 software. Their contribution is greatly appreciated.
1729

1730 REFERENCES

- 1731
1732
1733
1734
1735
1736
1737
1738 [1] McKenna F, Fenves G, Scott MH. Open System for Earthquake Engineering Simulation.
1739 PACIFIC Earthq Eng Res Cent 2000.
1740 [2] Maher ML, Fenves SJ. Role of Expert Systems in High-Rise Building Design. In: Beedle LS,
1741 editor. Second Century Skyscraper Counc. Tall Build. Urban Habitat, Boston, MA: Springer
1742 US; 1988, p. 571–89. doi:10.1007/978-1-4684-6581-5_47.
1743 [3] Sriram D, Maher ML, Fenves SJ. Knowledge-based expert systems in structural design. *Comput*
1744 *Struct* 1985;20:1–9. doi:10.1016/0045-7949(85)90047-1.
1745 [4] Syrmakizis CA, Mikroudou GK. ERDES - An expert system for the aseismic design of buildings.
1746 *Comput Struct* 1997;63:669–84.
1747 [5] Bennett J, Creary L, Engelmores RS, Melosh R. SACON: A Knowledge-based Consultant for
1748 Structural Analysis. Stanford, CA, USA: Stanford University; 1978.
1749 [6] Fjellheim R, Syversen P. An Expert System for Sesam69 Structural Analysis Program Selection.
1750 1983.
1751 [7] Chen JL, Hajela P. FEMOD: A consultative expert system for finite element modeling. *Comput*
1752 *Struct* 1988;29:99–109. doi:10.1016/0045-7949(88)90175-7.
1753 [8] Furuta H, Tu K-S, Yao JTP. Structural engineering applications of expert systems. *Comput Des*
1754 1985;17:410–9. doi:10.1016/0010-4485(85)90288-X.
1755 [9] Berrais A, Watson AS. Expert Systems for Seismic Engineering: the State of the Art. *Eng Struct*
1756 1993;15:146–54.
1757 [10] Dussom KB, Hadj-Hamou T, Bakeer RM. Quake: An expert system for the selection of design
1758 earthquake accelerogram. *Comput Struct* 1991;40:161–7. doi:10.1016/0045-7949(91)90468-2.
1759 [11] Koumoussis V.K. GC. GP. DC. Enhancing the Usability of Eurocode No 8 through Hypertext
1760 and Expert Systems. EG-SEA-AI 1st Work 1994;23:69–81.
1761 [12] Berrais A. A knowledge-based expert system for earthquake resistant design of reinforced
1762 concrete buildings. *Expert Syst Appl* 2005;28:519–30. doi:10.1016/j.eswa.2004.12.013.
1763 [13] Farinha F, Portela E, Domingues C, Sousa L. Knowledge-based systems in civil engineering:
1764 Three case studies. *Adv Eng Softw* 2005;36:729–39. doi:10.1016/j.advengsoft.2005.03.019.
1765 [14] Kent DC, Park R. Inelastic behavior of reinforced concrete members with cyclic loading. *Bull*
1766 *New Zeal Soc Earthq Eng* 1971;4:108–25.
1767 [15] Scott BD, Park R, Priestley MJN. Stress-Strain Behavior of Concrete Confined by Overlapping
1768
1769
1770

- 1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
- Hoops at Low and High Strain Rates. *ACI J* 1982;79:13–27.
- [16] Menegotto M, Pinto PE. Method of Analysis for Cyclically Loaded R. C. Plane Frames Including Changes in Geometry and Non-Elastic Behavior of Elements under Combined Normal Force and Bending. *Proc IABSE Symp Resist Ultim Deform Struct Acted by Well Defin Loads* 1973:15–22. doi:<http://dx.doi.org/10.5169/seals-13741>.
- [17] Clough RW, King IP, Wilson EL. *Structural Analysis of Multi-storey Buildings*. J Struct Div ASCE 1964;90:19–34.
- [18] Avramidis I, Athanatopoulou K, Anastasiadis K. *Prototype Structural Analysis Examples (in Greek)*. Thessaloniki: Aivazis; 2003.
- [19] Krawinkler H, Seneviratna GDPK. Pros and cons of a pushover analysis of seismic performance evaluation. *Eng Struct* 1998;20:452–64. doi:10.1016/S0141-0296(97)00092-8.
- [20] Taucer F, Spacone E, Filippou F. A fiber beam-column element for seismic response analysis of reinforced concrete structures, Report No. UCB/EERC-91/17, Earthquake Eng. Research Center, University of California at Berkeley, CA, 1991. 1991.
- [21] Mylonakis G, Gazetas G. Seismic Soil-Structure Interaction: Beneficial or Detrimental? *J Earthq Eng* 2000;4:277–301. doi:10.1080/13632460009350372.
- [22] Gazetas G. Formulas and Charts for Impedances of Surface and Embedded Foundations. *J Geotech Eng* 1991;117:1363–81. doi:10.1061/(ASCE)0733-9410(1991)117:9(1363).
- [23] FEMA. *Improvement of Nonlinear Static Seismic Analysis Procedures*. FEMA 440, Fed Emerg Manag Agency, Washingt DC 2005.
- [24] Kim S, Stewart J. Kinematic Soil-Structure Interaction from Strong Motion Recordings. *J Geotech Geoenvironmental Eng* 2003;129:323–35. doi:10.1061/(ASCE)1090-0241(2003)129:4(323).
- [25] Veletsos A, Prasad A. Seismic Interaction of Structures and Soils: Stochastic Approach. *J Struct Eng* 1989;115:935–56. doi:10.1061/(ASCE)0733-9445(1989)115:4(935).
- [26] Lagaros ND, Fragiadakis M. Evaluation of ASCE-41, ATC-40 and N2 static pushover methods based on optimally designed buildings. *Soil Dyn Earthq Eng* 2011;31:77–90. doi:10.1016/j.soildyn.2010.08.007.
- [27] Sextos AG, Pitilakis KD, Kappos AJ. Inelastic dynamic analysis of RC bridges accounting for spatial variability of ground motion, site effects and soil-structure interaction phenomena. Part 1: Methodology and analytical tools. *Earthq Eng Struct Dyn* 2003;32:607–27. doi:10.1002/eqe.241.
- [28] European Committee for Standardization (CEN). *Eurocode 8: Design of structures for earthquake resistance—Part 4: Silos, tanks and pipelines (EN 1998-4: 2006)*. Eur Comm Norm Brussels 2003;2.
- [29] European Committee for Standardization (CEN). *Eurocode 8: Design of structures for earthquake resistance—Part 1: General rules, seismic actions and rules for buildings (EN 1998-1: 2004)*. Eur Comm Norm Brussels 2004;1.
- [30] Fajfar P, Gaspersic P. The N2 Method for the Seismic Damage Analysis of RC Buildings. *Earthq Eng Struct Dyn* 1996;25:31–46.
- [31] Kunnath S. Identification of Modal Combinations for Nonlinear Static Analysis of Building Structures. *Comput Civ Infrastruct Eng* 2004;19:246–59.
- [32] Seismosoft. *SeismoStruct v7.0.4 – “A computer program for static and dynamic nonlinear analysis of framed structures”*, available from <http://www.seismosoft.com> 2013.
- [33] Mander JB, Priestley MJN, Park R. Theoretical Stress-Strain Model for Confined Concrete. *J Struct Eng* 1988;114:1804–26. doi:10.1061/(ASCE)0733-9445(1988)114:8(1804).
- [34] Adam C, Jager C. Simplified collapse capacity assessment of earthquake excited regular frame structures vulnerable to P-delta. *Eng Struct* 2012;44:159–73. doi:10.1016/j.engstruct.2012.05.036.