archi|DOCT

The e-journal for the dissemination of doctoral research in architecture.

Supported by the ENHSA Network | *Fueled by the* ENHSA Observatory

10

February **2018**
www.enhsa.net/archidoct
ISSN 2309-0103

RISK

Lifelong Learning Programme

european network enhsa of heads of schools of architecture

European Observatory of Doctoral Research in Architecture

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

Reliability associated with the use of building structural analysis and design software

Ivan Cabrera i Fausto, Ernesto Fenollosa, Verónica Llopis and Ana Almerich-Chuliá //

Department of Continuum Mechanics and Theory of Structures,
Higher Technical School of Architecture //

Universitat Politècnica de València, Spain

Abstract

Architecture education in Spain presents a vast technical component which includes the capacity for analyzing and designing building structures. The current architectural context encompasses buildings with an increasing structural complexity which demands the necessary use of computer tools. Professionals looking for a building analysis and design software face substantial difficulties when trying to find not only a roster of commercial products available on the market, but also comprehensive studies about their characteristics, especially regarding their functionality and reliability. National codes neither standardize this type of computer tools' performance, nor supply guidelines which enable the comparison of their features and performance or, at least, which enable to validate the functioning of a certain one. Simultaneously, available international bibliography is scarce, dispersed and clueless about how to confront these tasks independently. However other neighboring scientific fields have already met the challenge of assessing the implicit risk when using certain products and tools by developing well defined methods with theoretical background, detailed principles and development criteria and quantitative assessment. Most of these methods are based in the comparison of results obtained when using a certain product or tool under controlled conditions with reference results obtained by other means. This way of working is clearly related with test beds and is easily translatable to the field of building structures analysis and design. A testbed based on a conventional building structure would enable current or future users of certain software to compare the results obtained with this application with exact values provided by other tools or methods with proven reliability which strictly apply the current regulations and codes. Thereupon practitioners would have not only an appreciation but even a quantitative assessment of the underlying risk in the use of this software application.

Keywords

Architecture; Risk; Reliability; Structural analysis; Structural Design; Building Structures; Software; Testbed.

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

1. Architecture and structural analysis and design:

1.1. Architecture's technical dimension:

In 25 BC the Roman architect and engineer Marcus Vitruvius Pollio wrote a treatise on architecture and building techniques in Rome which received the title of "De architectura". It was divided in ten books and dedicated to Augustus. In the third chapter of the first book when referring certain public buildings Vitruvius wrote:

> All these should possess solidity, utility, and beauty. Solidity arises from carrying down the foundations to a good solid bottom, and from making a proper choice of materials without parsimony. Utility arises from a judicious distribution of the parts, so that their purposes can be duly answered, and that each has its proper situation. Beauty is produced by the pleasing appearance and good taste of the whole, and by the dimensions of all the parts being duly proportioned to each other (Vitruvius 25 BC).

What would be later on known as Vitruvian triad and popularized by Claude Perrault in the summary published in 1673 should occur simultaneously, with no exception and only when perfectly balanced success could be ensured.

1.2. The structure:

The word "structure" is a cultured term, transcription and translation from the past participle of the Latin verb "*struere*", which means to build. It was initially used for architectural constructions and frequently limiting its meaning to the resistant part of the building. Nowadays, we can define the structure as the set of objects or elements placed within a building and linked among them in such a way that are able to resist and transfer to the soil the applied forces. To this end, the requirements of strength, stiffness, stability and durability should be met, as long as the aforementioned prerequisites of solidity, utility and beauty.

1.3. The structural analysis and design:

Structural analysis and design is the subject which deals with predicting and determining a structure's response when receiving a set of external stimuli usually denominated forces: static or dynamic loads, changes of temperature, building inaccuracies, etc. If the structure is being projected, its objective will be to produce an adequate structural design for the demands and goals of the project. To this end, the dimensions of the elements are to be defined and building materials are to be chosen in such a way that the structure can withstand the external forces that will be applied on it. In other words, choices have to be made in such a way that stresses and deformations do not exceed those limits that would produce failure of the structure or that would turn it unable for its function. If the structure is already built, its objective will be to check the dimensions of the existing elements by verifying the inner stresses.

In either instance, the process to be followed is composed by:

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

- Preliminary design:
  Alternative proposals and structural system selection

- Analysis of the selected structural system:
  Once a particular option has been selected, the design has to be developed in a sequence composed by three stages that would be repeated until a satisfactory final result is achieved:

  - Preprocessing or modelling:
    A theoretical model needs to be elaborated. It has to be sufficiently simple, but it has to be close enough to the actual structure and suitable to be analyzed by means of available analysis techniques. The model definition encompasses identifying the building bearing structure by means of determining its geometry and the sizing of the different components which constitute it; selecting the materials; interpreting the behavior of the different element joints and bearing points; and, finally, assessing the loads by means of determining their magnitude, position (figure 1), frequency, origin and grouping them in types and establishing the convenient load combinations.

  - Processing or structural analysis:
    Once the underlying assumptions have stablished, the model will be analyzed by means of an adequate method whose equations will provide us the structural response of the different components regarding internal forces and movements. In structures modelled with bars, forces in member ends and joints displacements will be obtained (figure 2). Then and based on the principles of mechanics of materials, the study of the internal forces and deformations for the different elements will be carried out. Results should be assessed and interpreted according to the adopted model in order to determine the feasibility of the selected structural system.

  - Postprocessing or final design:
    Once the internal forces and deformations for any structural component have been determined, ultimate limit states and serviceability limit states need to be checked for all of them in order to verify the validity of the initial dimensions for these elements. In the case of insufficiency, the sizing of the section should be increased. On the contrary, if the initial dimensions reveal an excess, the sizing should be reconsidered by diminishing it so as to be true to the utmost economy principle. Either way, it will be necessary to carry out the processing again, since the section properties of the different elements do affect the internal forces distribution and the displacement values for the whole structure. Not only the scarcity but also the excess when notorious might even suggest a partial or global reconsideration of the structural typology chosen. Design will be over when acceptable values for the resistant capacity and the deformability and habitability have been reached for the different structural components and for the structure as a whole.

  - Deliverables management and works supervision:
    The results obtained must be transferred to the calculations report and to the necessary blueprints to enable the structure completion. The works should be supervised in order to check that the structure final characteristics correspond to those considered during the analysis and desi **1. Architecture and**

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

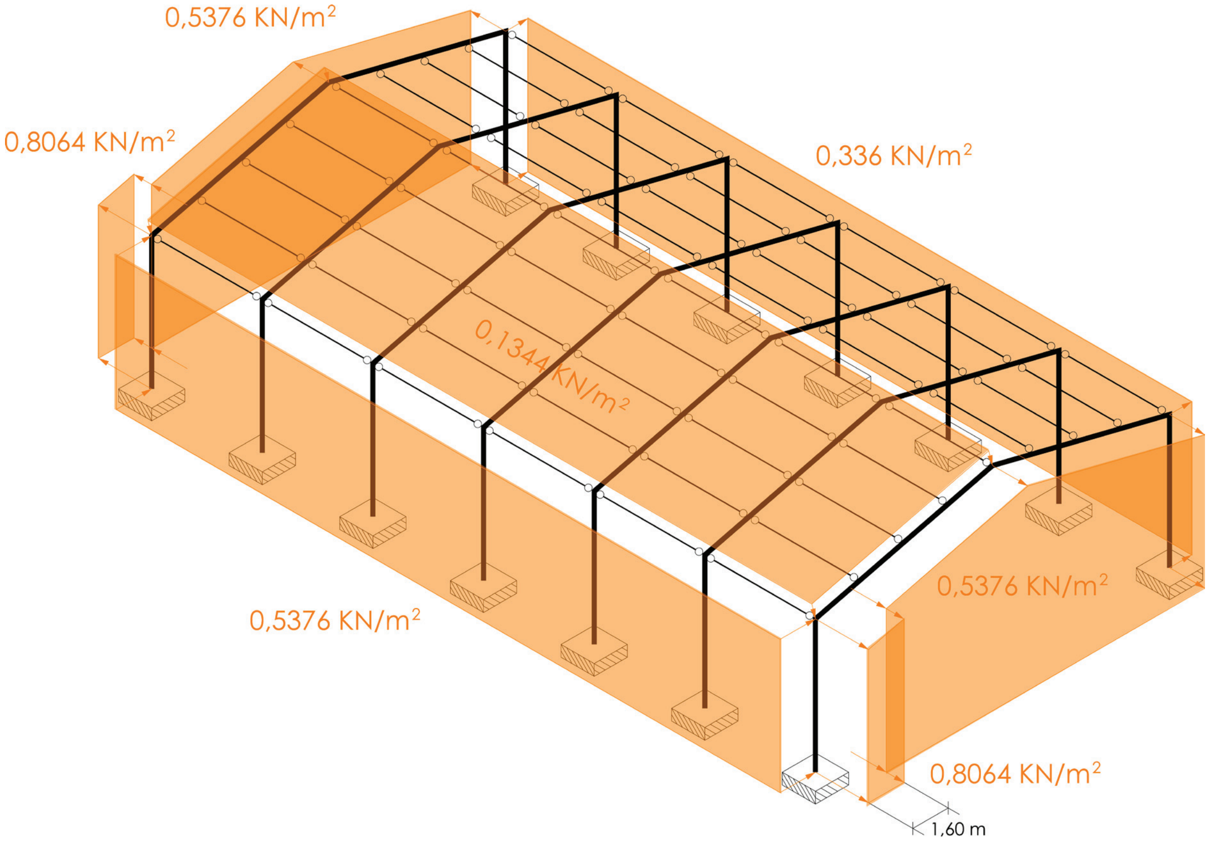

Figure 1.
Western wind loads applied on a structural model.

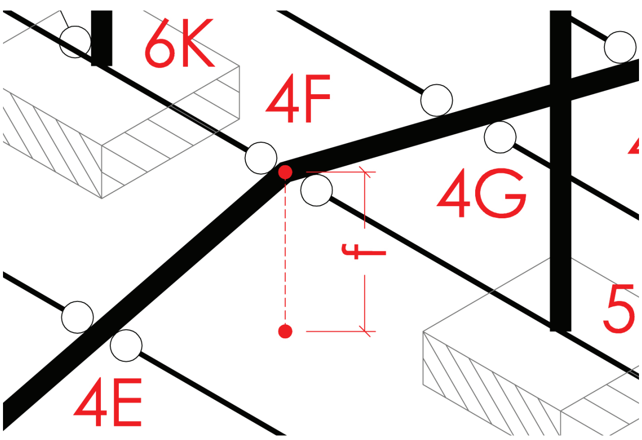

Figure 2.
Measurement of a joint vertical displacement in a structure modelled with bars.

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

2. Computer-assisted structural analysis and design:

2.1. The components of the evolution of structural analysis and design:

The evolution of structural analysis and design along history has been determined by the breakthroughs in different subjects including:

- The development of Mathematics and Physics, especially in sciences such as Mechanics
- The introduction of new materials such as iron, reinforced concrete, prestressed concrete, timber, etc.
- The creation of new structural shapes based on the development of new building technologies
- The development of new procedures and instruments for experimentation
- The formulation of computer science techniques and the development of computers that ease the solution of the equations needed in the different analysis and design methods

2.2. Origins of the computer-assisted structural analysis and design:

In 1946 a machine called ENIAC (Electronic Numerical Integrator And Calculator) was built at the University of Pennsylvania. With more than 18.000 valves and weighing up to 30 tones, it required a room of 200 square meters (Langlais 1985). It is considered the first large scale, electronic digital computer for generic use. Anyhow, it had been preceded by many others small scale machines such as the ABC, the Colossus, the Zuse's 73, and even by other some never built examples such as the analytical machine of Babbage (Rojas & Hashagen 2000). Later on, the invention of the transistor in 1948 and of the silicon chip in 1959 kicked off an uninterrupted evolution that would stir up even the most unexpected aspect of human life. Computers have always been instruments endowed with the capacity to learn how to do any task as long as there is a way to teach them. Their development undeniably fostered the subject of structural analysis and design (Ramírez de Dampierre 1981). Since the first software for structural analysis and design was elaborated and performed in the mid-1950s, computers have been helpful for reducing the time required for calculations and have improved the design processes efficiency (Rojiani et al. 1994).

2.3. Evolution:

The development of specific software for structural analysis and design was made namely at the universities and research centers. The United States were the main hub for the development of this kind of products. We could cite plenty of examples from these times. But among them the most remarkable one is the structural analysis and design software by then called STRUDL (STRUctural Design Language) which was developed by the Massachusetts Institute of Technology for IBM and that became the reference software for many offices around the world. Starting from its original version, the Georgia Tech developed between 1975 and 1977 the GTSTRUDL whose daily use would spread over more than two decades (Rojiani et al. 1994).

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

In the 1980s, the emerging expertise in the use of structural analysis and design software extended the application beyond conventional preconceived structural cases. The first studies on structural optimization appear by means of using computers as machines for analyzing, designing, reanalyzing and redesigning as many times as needed. These procedures were supposed to be automatized in such a way that computers would turn into generators of optimal structures for any building (Isreb 1984).

In the early 1990s most of the structural analysis and design software products required the data input alphanumerically with the subsequent effort and risk of committing mistakes during this process. The lack of a visual check hindered an efficient software use and demanded a significant effort in order to check the exactitude and correctness of the structural model. By then a new generation of structural analysis and design software arose with a more significant functionality since the user was able to input the data with a graphic interface making easier the structural model construction and the visualization of the analysis results (Rojiani et al. 1994). Graphical interfaces were designed so as to provide information about the structural typology, the type of structural components and the loads applied on them. It was also possible to check how the different components were assembled, which were de dimensions of each section and it was even possible to get some comments about the design allowing the user to accept or to reject recommendations suggested by the system (Biederman 1996). In the mid-1990s the cost of personal computers which could run structural analysis and design software kept the downward trend. Their progressively improving features and capacities of their central processing units opened the door to all kind of design software, even for those which implied the use of finite elements (figure 3) whose theoretical origin had been initiated several decades ago, but whose more intense development had been boosted just some years ago (Gendron 1997).

The computerization of the structural analysis and design was by then almost complete and a significant part of the vast commercial offer was also able to perform the different verifications and to design according to many specific national codes (Biederman 1996). The relentless improvements in the graphic interaction with users fostered the development of many structural analysis and design software of this typology (Rojiani et al. 1994). Any practitioner employed by then a computer in his daily duties akin to building design (Biedermann 1996). Even though the theoretical development of object-oriented programming dates back to the 1950s and 1960s at the Massachusetts Institute of Technology, it wouldn't become popular in the market until this decade. Its approach simplified even more structural analysis and design software usage since employed components resembled noticeably to their peers in the actual structure. As a result, the structural model of a building (figure 4), always being a quite complex physic-mathematic entity, could be implemented as the addition of structural elements, groups of them, joints, etc. Each part of this addition could be visualized as an independent object with different levels of abstractions (Biedermann 1996).

Structural analysis and design is subjected to a significant level of uncertainty because of the imprecise values for loads, material properties, components geometry or bearing points performance. The need for tools which could handle and compute these uncertainties gave place to the stochastic calculus software. The quick advance in computer applications in general terms during the eighties and nineties favored the development of this sort of software which was applied to a great range of problems with academic and engineering interest (Pellissetti & Sueller 2006). The effective numerical procedures behind these software considered structural uncertainties and quantified results in terms of probability. In the late 1990s, these methods were implemented in calculus environments in such a way that their management was rather easy for any practitioner (Schueller 2000).

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

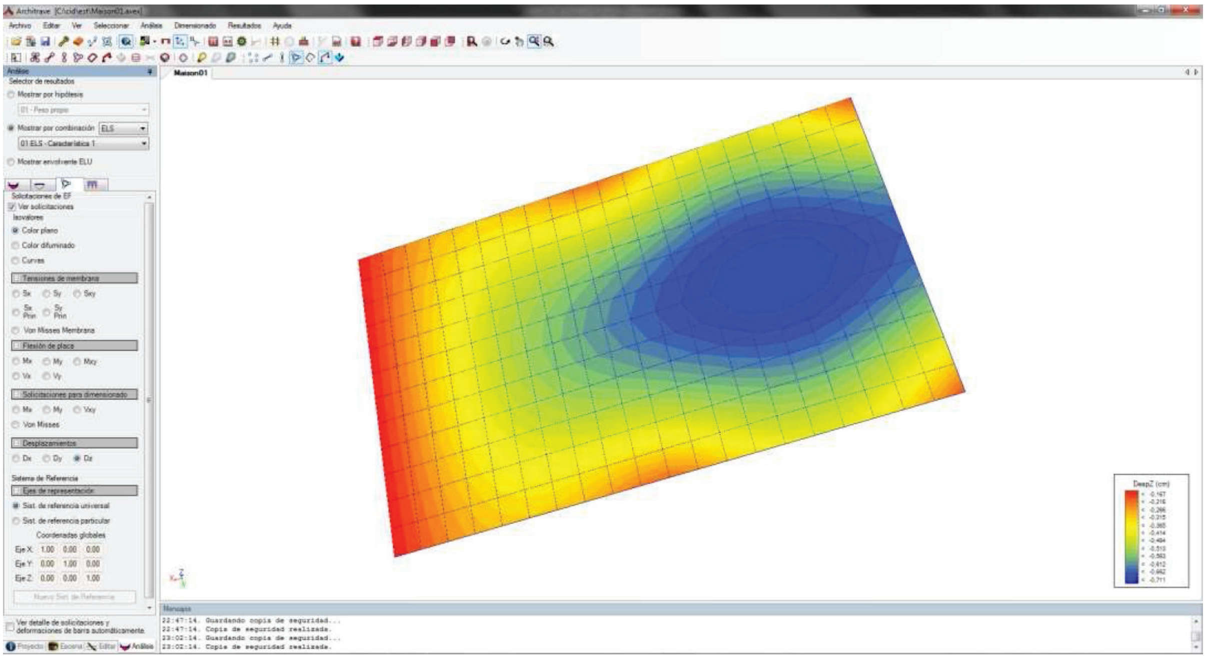

Figure 3.
Slab vertical displacements on a finite element model obtained employing the software Architrave®.

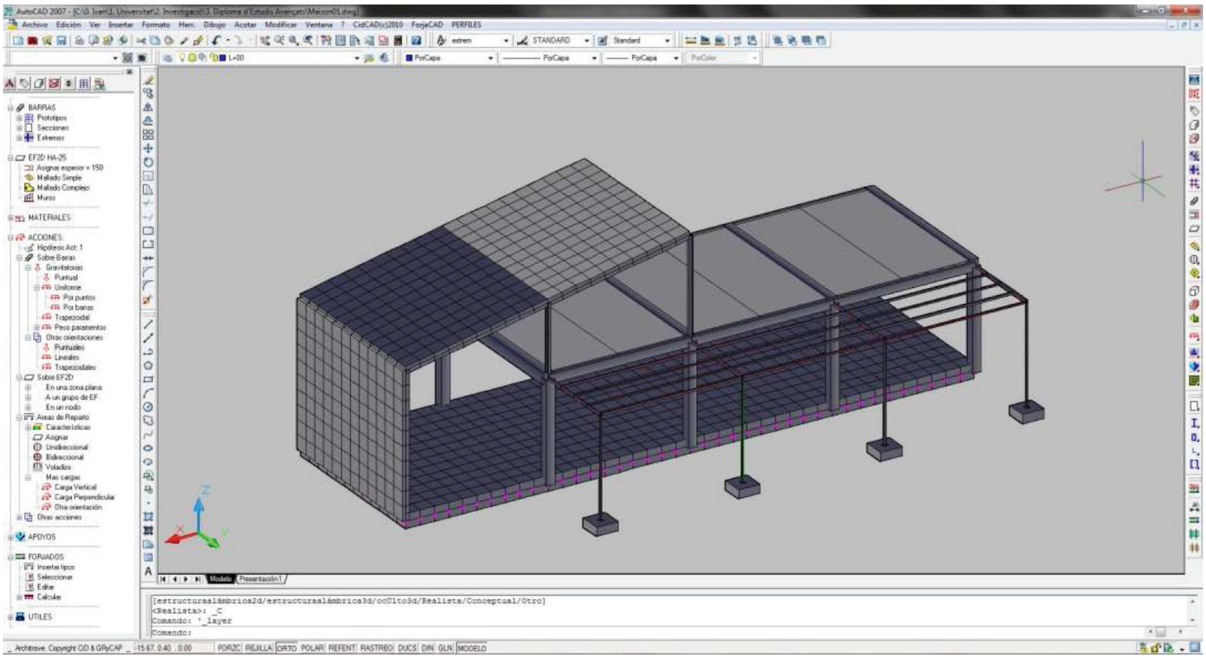

Figure 4.
Structural model of a family house during the input data process employing the software Architrave®.

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

Nowadays, structural analysis and design demanding high-performance is still run with software installed in individual computers, running counter to the trend of computer engineering market which has migrated most of its applications to the cloud. The emerging "Cloud computing" is a service delivery innovative model which utilizes the internet as a distribution channel (Gracia & Bayo 2013). The system structure enables users to access to a catalogue of services where only those used should be paid. Some of the products have already been designed and even incipiently commercialized identifying their modules with the traditional stages of structure preprocessing, processing, postprocessing and deliverables generator.

Finally, the wide dissemination which architectural design software based on Building Information Modelling (BIM) has nowadays, is approaching step by step the field of building structures analysis and design. If all the components within a building have been defined in the computer model, the structure as a part of the building should also be. Software capable of identifying the structural elements within these building computer models and capable of obtaining all the necessary data to define the structural model are around the corner. Having run the analysis and design stages, blueprints of the structure will be obtained normally.

2.4. Particularities and modules:

When describing any software both producers and users structure their speech in four clearly defined issues which exactly correspond to the four aforementioned stages of structural analysis and design: preprocessing or modelling, processing or analysis, postprocessing or design and checking of elements, and deliverables making (Rojiani et al. 1994).
During the preprocessing, the accuracy in the computer structural modelling performed by users has a crucial transcendence (Zhao & Zhen 2013). Making a simplified representation of the real structure involves great ability and expertise, turning this stage into the maybe the riskiest phase of the whole process as will be detailed later on.

As previously explained, during the processing the software will analyze the structure determining the internal forces, movements, stresses and deformations in the different elements that make up the structure. Its exactitude and validity perhaps is the most delicate point in the whole process. Employing a poor-quality structural analysis and design software is extremely dangerous for users, especially if we take into account that they are responsible for the software that they have chosen (Emkin 1988). Most of the software packs for building structures analysis and design are composed by closed systems which do not allow users to verify those assumptions and computes made during each stage of the process (Kao & Yeh 2014). Hence, internal forces diagrams, depictions of the deformed structure, and stresses and deformation graphics, constitute some of the few chances that practitioners have so as to verify the correctness of not only the model but also of the calculations made on it. Unfortunately, only those structural analysis and design enthusiasts assiduously and thoroughly check these diagrams and graphics, perfectly aware of their power for revealing potential mistakes. On the contrary, many others do not take the time for examining this information and just rush to get the final results of a postprocessing that might have been made on a processing full of hidden mistakes.

After the postprocessing carried out with the dynamics and limitations of the corresponding national code, the deliverables making is usually understood as the structure blueprints generation and plotting. However, plenty of structural analysis and design software can supply other informa-

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

tion related to the foundations and structure that might be included in the project such as the regulatory compliance report, the terms of reference, etc.

At international level and nowadays, there are numerous software packs developed for building structures analysis and design. Although preprocessing and processing modules have been developed with scientific criteria universally accepted and hence valid for any country (figure 5), post-processing modules should adapt to the specific legislation of each state. While there are products that offer users the chance to develop this third stage depending on different national codes, the most frequent case is that comprehensive products specifically designed for their national codes are available in any country. In the specific case of Spain, the range of products is significant, but their establishment is very uneven with a reduced number of software controlling most of the market.

3. The selection of a structural analysis and design software:

3.1. The need:

The current architectural context encompasses buildings with an increasing structural complexity. New structural typologies and the arousing development of the existing ones have produced structures highly statically undetermined which have complicated gradually their analysis and design by hand. This circumstance has been especially accentuated in those structures with a significant presence of two-dimensional elements which cannot be modelled by means of bars but by means of finite elements and whose analysis without computer assisted methods is unthinkable (Gensichen & Lumpe 2008).

Therefore, nowadays practitioners must choose a software application for building structures analysis and design and incorporate it to the set of tools employed in their daily practice. Learning how to use this kind of software must happen in the schools of architecture, along with the learning of the theoretical concepts that underlie in the performance of these applications, and along with the solving of simple cases by hand that will provide students with order of magnitude and intuition about the results to be expected.

3.2. Casuistry and quandary when choosing:

Not only practitioners who look for a software application as a working tool but also university professors that need it for their teaching must face the difficulty of buying a product that fulfills their requirements. Both will find serious obstacles to find a rigorous roster of all the products available in the market and should compose an initial list with information that would come namely from three sources: commercials in specialized journals, conversations with other colleagues and compilations displayed in certain websites. It is difficult to assess the reliability and exhaustiveness of the information that can be obtained by each of these three mechanisms, but most experiences reveal that each one has certain features that absolutely characterize them.

Commercials included in architecture journals, both in the first pages or in the back cover, usually exhibit the product virtues and focus in the last update. Indeed, these commercials frequently seem much more addressed to users of previous versions than to possible new users. Only in those commercials formats with more capacity such as brochures or centerfolds, we will be able to find

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

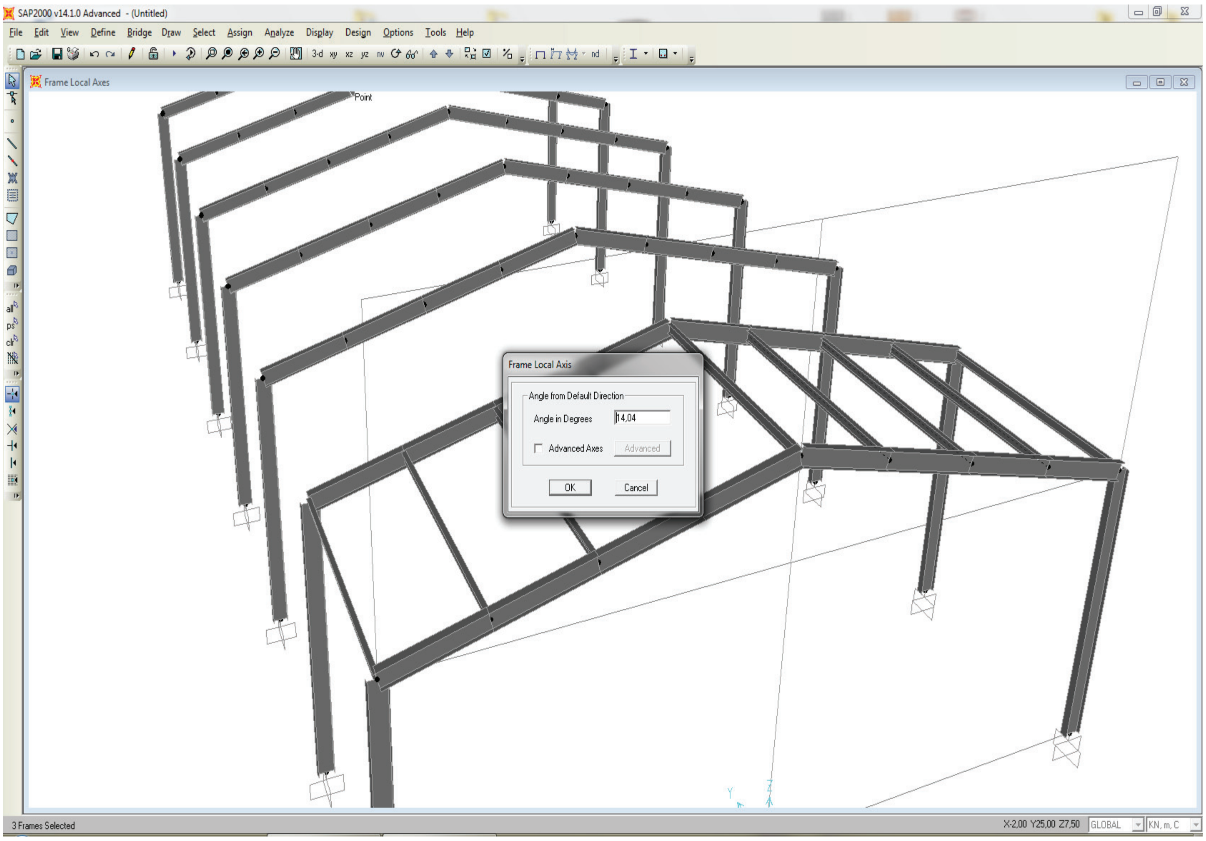

Figure 5.
Structural model preprocessing employing the international software SAP2000®.

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

a more detailed explanation of the application features, which, obviously, enhances its merits and mutes its scarcities. Anyhow, after a thorough analysis we will have no clue about which capacities make it better or worse than other similar products and the comparison tasks will be our responsibility.

The opinion of our colleagues might frequently be too subjective. When they made their selection it is quite possible that they didn't choose the optimal software but the one that fit their particular needs the most. Those needs in the case of a university professor might be related to the academic skills of its student and the type of lectures that must be taught. And in the case of a practitioner, those needs might be conditioned by the usual typology of commissions which receives. In any case, the final choice might have been conditioned by something completely unconnected to quality such as its price. As it happened when checking commercials, there is no reliable comments which compare different options. Anybody will justify why he or she chose one product, but mainly providing comments excessively focused in their current choice. When occasionally comments about other previously used products are provided, a narration about a bad experience that resulted in the give up of this product will show up.

Finally, rosters compiled in specialized but frequently non-official websites represent the only source where different structural analysis and design software applications are displayed on equal terms. These lists are usually limited to a more or less exhaustive enumeration of different systems and a brief description which seldom include comments comparing the features of different products. These critical analyses are namely found in opinion forums where the anonymity of users and the vehemence employed when defending certain points of view produce some skepticism.

Therefore, users eager to purchase a software application for structural analysis and design for their daily duties are rather devoid of conclusive evidence or arguments in order to choose the right option. A individual strategy will become a must taking into account the two fundamental factors when choosing a software application of any kind: functionality and reliability, which is to say the minimization of risks of any kind.

3.3. Functionality of a software application:

Any software application functionality is related to several aspects such as the facility for managing it, the capacity for limiting mistakes and the adjustment to the user's needs. Initially and apart from reliability, functionality is the feature that distinguishes one application from another and, most of the times, is the key factor when choosing one product over another (Rojiani et al. 1994). Each year industry creates a variety of applications which when already launched to the market make customers doubt about these products being adequate for their daily routines or not (Lam 2007). According to S. Redwine and W. Ridle (1985), there is a series of critical factors when the moment to use a new technology comes:

- Seriousness of the product: technology must be well developed.
- Clear acknowledgement of the need: technology must provide a response to a well-recognized and well defined need.
- Possibility to be customized: technology must be adaptable to the user's needs.

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

- Existence of previous positive experiences: a positive background must verify a good balance between the cost and the benefits obtained.
- Guidance: the product should include a user's guide with a large number of examples, especially in the case that new concepts come into play.

Simultaneously, there are many factors that may slow down the use of new technology and that consequently dissuade customers from buying and employing it:

- Too wide range of ways of using the product
- Difficult understanding of the new technology that involves high costs
- Large number of alternatives with different strengths and weaknesses

The final balance of all these parameters made by the customer will guide the final decision about buying or not a new software application.

3.4. Reliability of a software application:

We can understand reliability as the antonym of risk. When it comes to talk about the reliability of a structural analysis and design software application it is indispensable to clearly define three concepts which are absolutely different: reliability of the structure, reliability of the computerized analysis and design process and reliability of the software application itself.

3.4.1. Reliability of the structure:

According to Holick and Vrouwnvelder (2004), it is impossible to quantify a structure performance with precision. Thereupon, any building structure analysis and design implies uncertainty when results are displayed. Nowadays, there are techniques and methods commonly accepted in order to assess the reliability of a building structure. The most refined one is the Partial Safety Coefficients Method, based on the experience gained and in probabilistic concepts of structural reliability. Fundamental concepts on reliability are collected in the different national codes and in the International Standard ISO 2394 "General Principles on Reliability for Structures".

Reliability of the structure takes account of three fundamental factors: its strength, the service to be provides and its durability. Moreover, there are additional factors such as fire safety and other accidental situations (Holick and Vrouwnvelder 2004).

Civil structures analysis and design wouldn't be possible without taking into account the fact that there is a risk of failure during their life cycle. Assuming that circumstance, International Standard ISO 2394 defines reliability in a similar manner as many other national codes do: it is the capacity of the structure to accomplish determined requirements during its intended life cycle and under specific conditions. Therefore, in terms of quantity, reliability could be explained as the complementary concept of the probability of failure (ISO 1998).

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

3.4.2. Reliability of the computerized analysis and design process:

Since the 1990s, the use of IT resources to solve structural analysis and design problems is something usual among practitioners. Despite the fact that the benefits of employing a computer are undeniable, users never know for sure if results provided by the machine are reliable (Bell 1997). In this section we are not talking about the use of a stochastic or probabilistic method that aims to determine the reliability of a structure. Now we are thinking of, once any method has been chosen, how accurate are the results that software provides to practitioners at the end of the process.

Disparity between results and the values that would be obtained after a precise application of the chosen method might have been produced fundamentally by three different types of reason. The first one comes with the errors produced by a deficient employ of the structural analysis and design software. In other words, the application calculates perfectly a model which is not correct or the variables to be developed weren't properly selected. This is the type of reason that we will analyze in this section later on. The second reason comes with the mistakes produced by a deficient calculation run by the software. That is to say that the application doesn't calculate properly a model that was well defined. This casuistic will be analyzed in the next section. The third, last and most dangerous reason comes with a combination of the two previous ones which means that the model is not well adjusted to the actual structure and additionally the calculus has been faultily run. It is the most dangerous case because we would be designing a structure to withstand a set of situations that have nothing to do with those that will be faced in fact.

If the structural model doesn't represent properly the real structure, results after running the calculations might lead to a deficient structural performance and even to its partial or total collapse. All of it could be avoided if the person in charge for the modelling process and later analysis and design has the adequate theoretical and technical knowledge. In the case of having unwittingly committed any mistake or inexactitude, practitioners should be capable of running a verification of the results provided by computers. This verification should be carried out by applying the rationale so as to check that results should the practitioner's intuition and order of magnitude (Bell 1997).

A careful management of software and a critical analysis of results also provide practitioners with an estimation of the sensibility of the analysis results to the different approximations and simplifications done during the necessary discretization and idealization processes performed during the modelling (Rojiani et al. 1994). It is extremely convenient that practitioners are instructed in developing with correctness the modelling and in verifying results, since too rough rounding or approximations might lead to results that perhaps are not distorted enough to be easily detected (Melosh & Utku 1988).

3.4.3. Reliability of the software application:

As has been discussed in the previous point, even if the structural model has been carefully defined and the approximations and simplifications introduced during its preparation are reasonable enough so as not to produce a model which performs differently from the real structure, there is a last factor of risk related to the software not running the calculations well. Regardless of our trust in the software performance, once results have been obtained, practitioners should always run a series of quick calculations by hand in order to make comparisons. This verification will determine the judgment and sufficiency of results (Bell 1997). After all, computers operate following instruc-

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

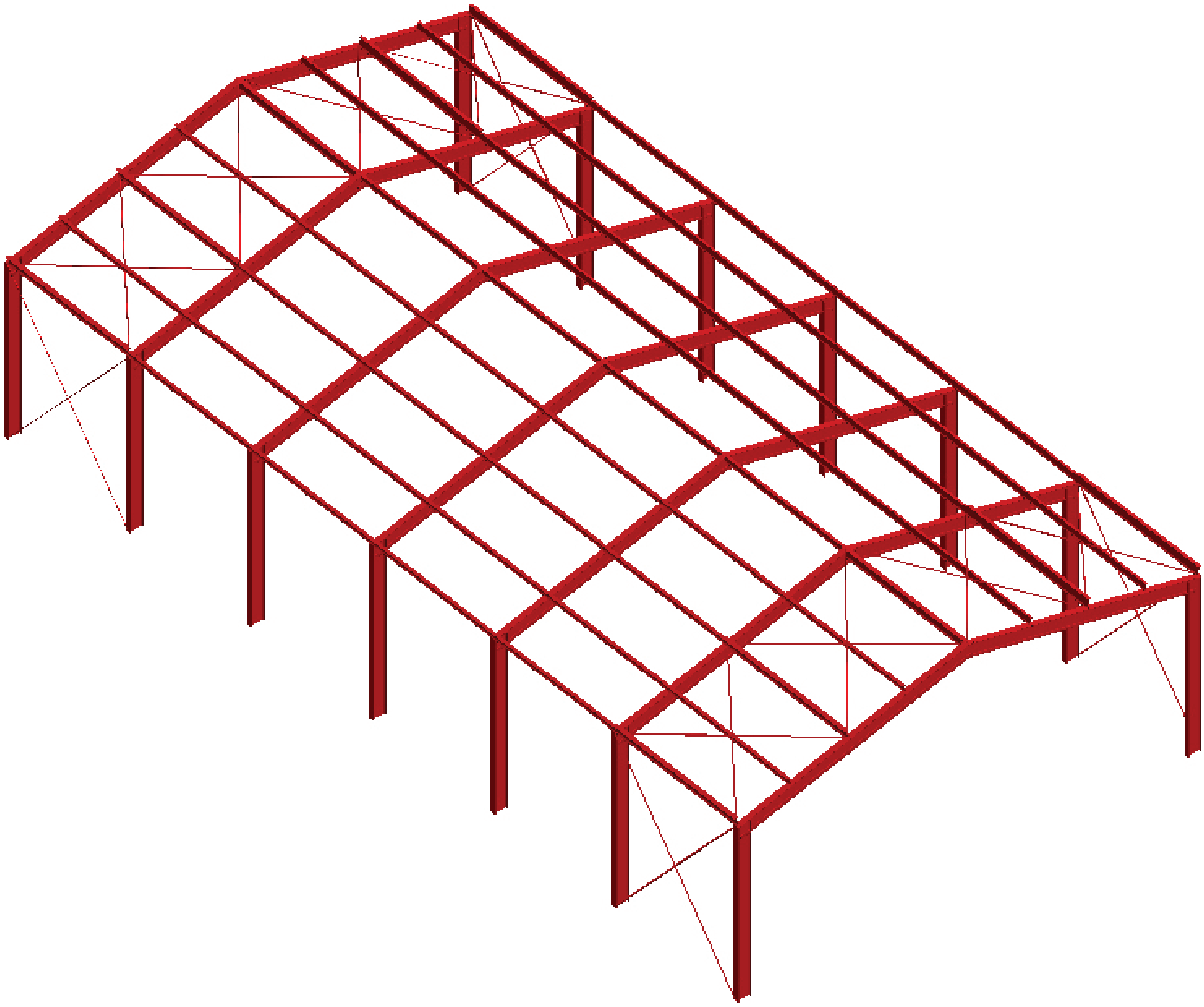

Figure 6.
Archetype of industrial plant structure employable in a testbed of reliability of structural software.

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

tions introduced by programmers in the structural software application and latent defects might take quite long to be detected (Ramírez de Dampierre 1981).

Against this background, it is very important to carry out a correct verification and assessment of this kind of software prior to its marketing. Likewise, the existence of national regulations and control agencies which take responsibility for the quality of software products for structural analysis and design is extremely important (Rojiani 1994). Meeting the principles compiled in these regulations and applied by the aforementioned agencies would become not only a reference to prevent the market debut of defective products, but also a tool for future users when trying to compare the reliability of already commercialized software products. Different results for the same structure provided by two different applications gave practitioners a terrible sense of unease.

We must accept that it is impossible to warrant a complete reliability for any structural analysis and design software application. Just as no scientific hypothesis can be certainly proved or discredited independently of the amount of experiments performed, we cannot warrant that any application is completely free of errors independently of the amount of practical demonstrations that we run and verify. Hence, assuming that reliability cannot be guaranteed quantitatively but qualitatively, it is interesting to develop certain tests that can foster as much as possible our trust in the accuracy of specific computer software (Melosh & Utku 1988). In this sense, a testbed based on a conventional building structure (figure 6) would enable software current or future users to compare the results obtained with this software with exact values obtained with other tools or methods with proven reliability which strictly apply the current regulations and codes.

3.5. Studying the reliability of structural analysis and design software:

3.5.1. Verification, validation and evaluation:

Verification, validation and evaluation play a fundamental role in the establishment of any software application reliability. Apart from the dynamics and methodology employed when making the product, verification and validation are the only tools that can warrant the quality of the product to the smallest detail (Wallace & Fuji 1989a). Both analyze and test the software to determine if it runs properly for its intended purposes, guaranteeing the absence of poor functioning and measuring the quality and reliability of results (Rojiani et al. 1994).

Verification is normally run by the developer of the software application and happens at any single stage of the product developing process (Rojiani et al. 1994).

Validation is the last test of a product when its development process is complete and aims to guarantee acceptable results (Wallace & Fuji 1989b). Validation implies a series of tests where the software application should solve a series of problems whose results will be compared with others whose reliability has been proven because of having been published, because of being widely known solutions, because of having been obtained safely by hand, or because of having been obtained with similar products with renowned prestige. Software can be validated when providing similar results to others available from any of the aforementioned sources (Emkin 1988).

Evaluation is focused on quality and functionality, identifying strengths and weaknesses of the software application that are not strictly technical (Rojiani et al. 1994). It is especially interesting for final

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

consumers since it provides information about the product ease of use (Priest 1988). Evaluation endows future users with the capacity to distinguish between one application which is friendly and another which is not and, therefore, might mislead practitioners (Vora 1986). Evaluation is developed by comparing the software pack performance in different scenarios with the performance of similar tools. It is not a simple process at all since these scenarios and categories to be compared depend quite a lot on the needs of each user. Moreover, final statements are rather subjective and almost impossible to quantify (Eskenasi 1989).

3.5.2. Responsibility on the study of structural analysis and design software reliability:

A special emphasis on guaranteeing the reliability of structural analysis and design tools is indispensable, since almost no regulation compiles a set of detailed tests to be developed (Melosh & Utku 1988). Consequently, there are few standardized case types in the field of structural analysis and design with available results to be used as benchmarks. Those people responsible for the production of this kind of software never took over seriously of the validation of their products and have traditionally trusted practitioners in determining the validity of results obtained with their software applications and their validity range. Likewise, big enterprises focused on analysis and design are understandably reluctant to devote sources for testing thoroughly structural software since testing is an intense task, that takes a lot of time and with few economic incentives. Therefore, many applications are in the market without verification or testing, or have been just roughly checked by their producers. That fact raises serious questions about their reliability (Rojiani et al. 1994).

3.5.3. Bibliography on reliability of structural analysis and design software since 1985:

Existing literature on reliability of structural analysis and design software is rather scarce. If we check the Web of Science about this topic, results are reduced an rather confusing since the term "reliability" refers to many different aspects of the analysis and design process such as the stochastic methods, the reliability of the process or the reliability of the software itself, being the last one the topic that we are really interested in now.

Therefore, a thorough review of precedents and their correct contextualization demands on one side employing as keywords for the search very generic terms, and on the other side examining carefully all the abstracts of the different papers so as to establish the final roster of documents that deal with this question.

If we check the Web of Science looking for documents which contain in its subject the terms "structural", "analysis" and "software", then refine the search filtering it only for these categories that may include papers of our field such as "Engineering Civil", "Engineering Mechanical", "Materials Science Multidisciplinary", "Computer Science Interdisciplinary Applications", "Mechanics", "Construction Building Technology", "Engineering Multidisciplinary", "Materials Science Characterization Testing", "Materials Science Composite", "Engineering Industrial" and "Architecture", and finally refine it again by filtering the results to those of areas with identic purposes such as "Engineering", "Materials Science", "Computer Science", "Mechanics", "Construction Building Technology", "Physics" and "Architecture", we will obtain 3111 results. A thorough review of all these papers focused

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

on selecting just those that deal with structural analysis and design software reduces this amount to 817, being the oldest document from 1984.

Figures 1 and 2 depict the number of papers published between 1985 and 2015 about structural analysis and design software. The size of each circle representing each lustrum is proportional to the amount of papers on this topic published during that period. Likewise, each circle has been divided in twelve main categories for classifying the selected papers after their study:

- Software production and its impact
- Finite element calculation
- Unconventional building typologies
- Design criteria
- Properties of materials
- Lateral loading effects
- Internal forces analysis
- Non-linear analysis
- Deformation, buckling and vibrations
- Analysis and design of joints
- Sustainability, assessment and reinforcement
- Failure and collapse

Each portion hosts a number which indicates the amount of articles published that can be classified under this category in the corresponding lustrum. By observing the sequence we can see that the production of software and its impact, which is the category that comprehends papers on this type of software reliability, was the only topic in the scarce documents published in the late 1980s. That amount grew the decade after and in the 2000s, always keeping a significant presence in percentage. Anyhow, it progressively gave prominence to papers on lateral loads such as wind, earthquakes and blast. However and surprisingly, the last five years have seen a surprising contraction of the amount of papers devoted to software production and its impact. That circumstance is rather surprising in a context where the amount of papers published every year is on the increase.

Finding bibliography which explicitly deals with the evaluation of reliability of structural analysis and design software or with comparisons between different software is an arduous task. There are few examples of papers focused on comparing results of different software applications when performing the same case, but we can list the seismic study of reinforced concrete frames developed by V. Pereira, R. Barros and M. César (2010); the seismic study of high-rise buildings with oblique column wrote by K. Hu, Y. Yang and S. Mu among other (2012); or the evaluation of the effects of a blast on light roof steel structures made by J. Geringer, C. Tuan and P. Lindsey (2013). None of them mentions any precedent neither regarding similar studies nor regarding the use of a specific methodology about how to raise comparisons.

As previously mentioned, literature about studies of reliability of structural analysis and design software is rather scarce. But still, from the 1970s until now, several science people have met the challenge of verifying the correctness and accuracy of results of several applications in different contexts. In all the cases of reliability studies and comparisons performed, the surprising absence of precedents and bibliography about how to design the process is quite remarkable. Some of them occasionally allude to previous similar studies. But no one refers to theoretical approaches, although borrowed from neighboring fields, that might guarantee the quality and impartiality of the

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

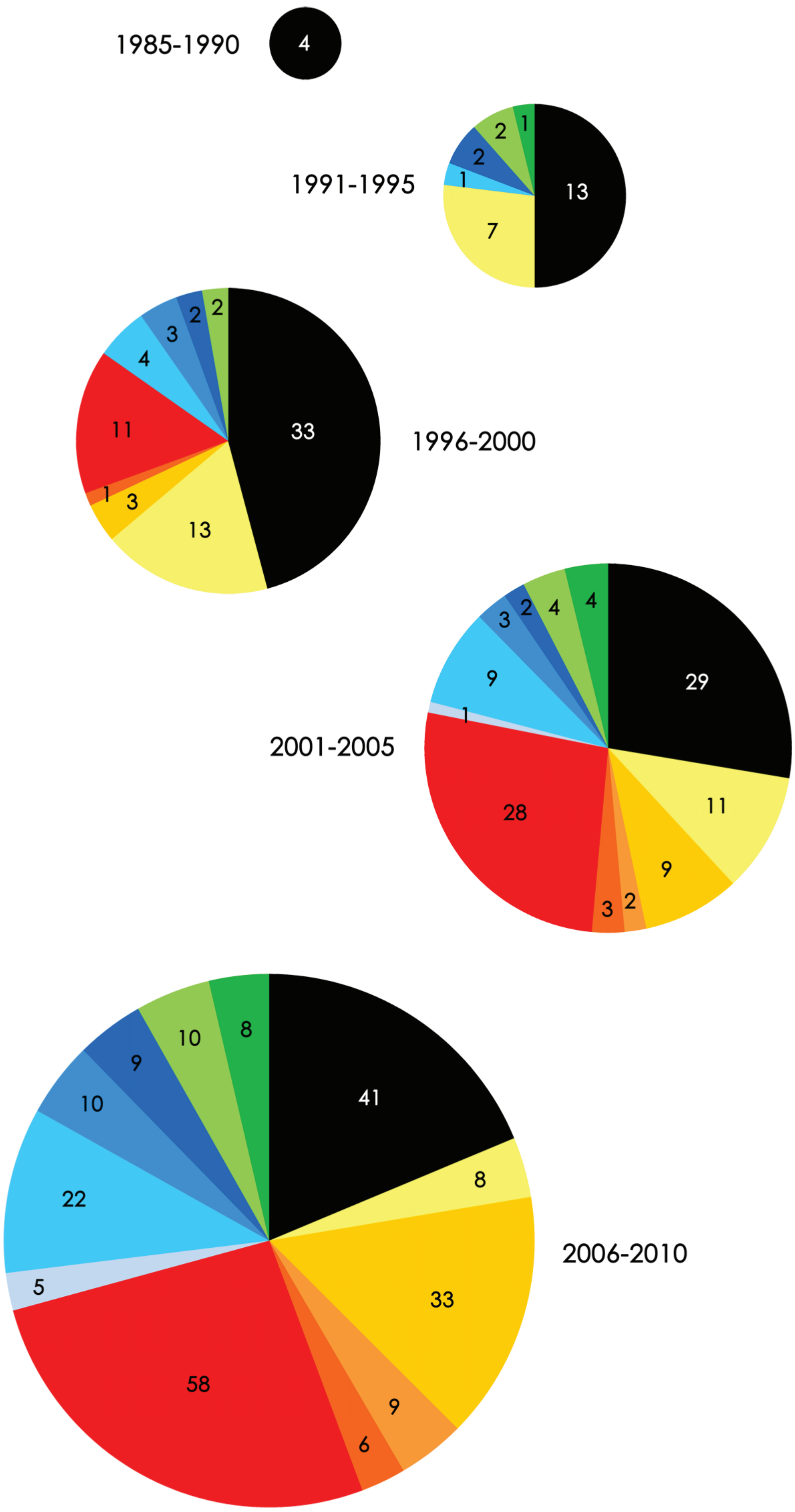

Figure 7.
Number of papers on building structures analysis and design software published between 1985 and 2010 according to the Web of Science.

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

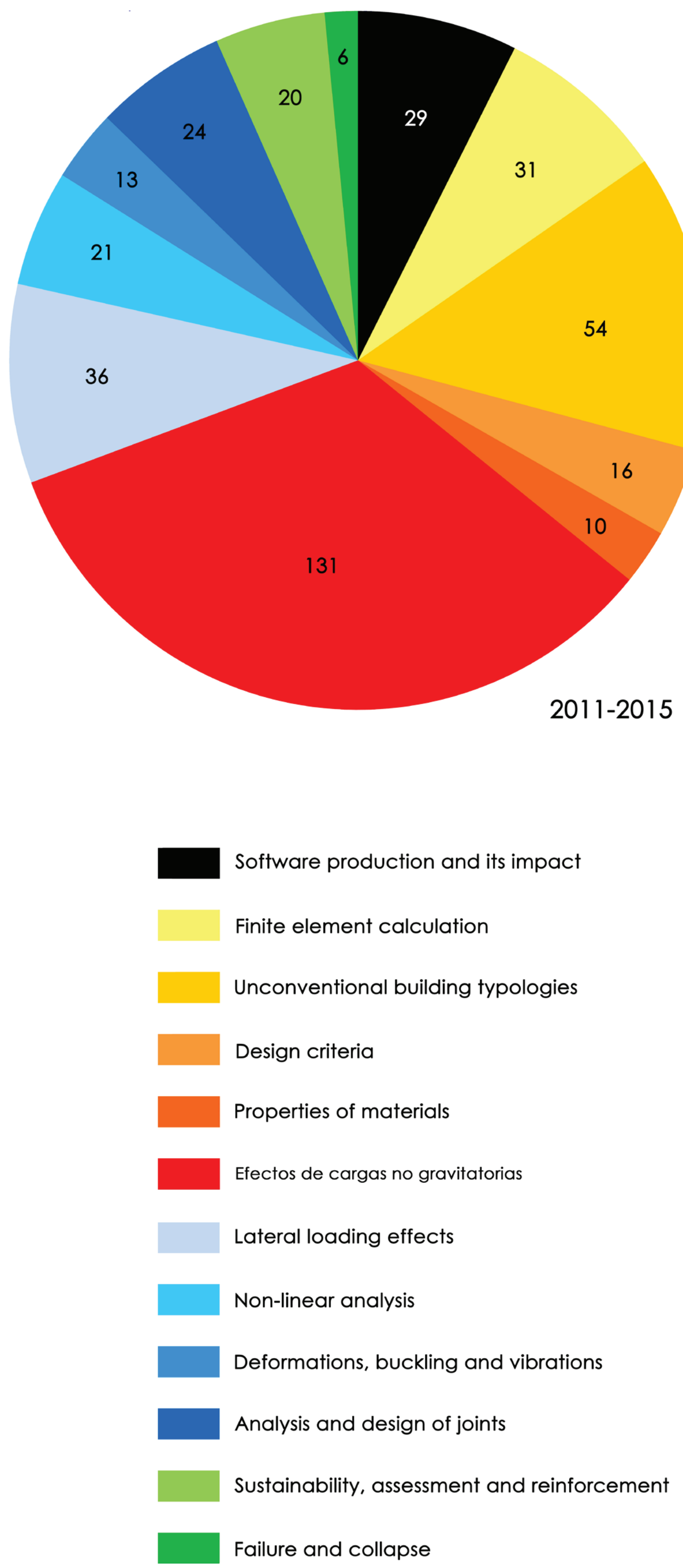

Figure 8.
Number of papers on building structures analysis and design software published between 2010 and 2015 according to the Web of Science

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

assessment process and its conclusions. At the very most, some criteria considered adequate by the authors are included but never justifying their origin apart from common sense. This lack may reside in the scarcity of this type of works, not allowing the formation of a discipline which seriously assumed the compilation of its own theoretical approaches. However, it is also true that in other neighboring fields, other authors have developed guidelines which are easily translatable to the quality evaluation of structural analysis and design software.

3.6. The importance of comparing results:

Most of the current codes about building structures analysis and design refer explicitly the possibility of running the structural analysis of design with IT resources. In the specific case of the Spanish "Código Técnico de la Edificación. Documento Básico sobre Seguridad Estructural" (Ministerio de la Vivienda 2009), practitioners are requested to write a report detailing their acknowledgement the software employed is adequate for the structure that has been analyzed and designed. This regulation also demands a detailed writing about how the model was input and how the results have been interpreted. Surprisingly, it never questions the software proper function and the validity of the solutions provided. However, there are other national codes which do not trust software that much in the case of dangerous buildings. So for example, the Chinese code for seismic-resistant buildings demands that the analysis of internal forces and deformations of complex structures under the effects of frequent earthquakes must be performed at least with two different mechanical models and results for both cases must be compared (National Standards of the People's Republic of China 2010).

As previously referred, nothing might look more alarming for a practitioner than the same structural model for a building structure producing different results depending on the software used for its analysis and design. Some studies reveal that dispersions can go further than 30%. That means that, at least one of the software applications employed and under comparison lacks rigor if both have been properly used (Gensichen & Lumpe 2008). Occasionally, that dispersion which immediately means suspicion is not detected in studies that compare the same case on different IT resources but during continued use of the same tool employed for solving many different cases. Dispersion of results obtained with different software applications reveals unequivocally that at least one of them was not conveniently validated. In order to reduce this uncertainty it is indispensable to assess the performance of software by means of solving cases whose results are previously known and verified, even empirically. That will allow us for certainly determining which tool has a poor functioning. When informed about the programming error that are producing this wrong results, producers will be able to remedy the internal functioning mechanisms of the tool and consequently to enlarge the scope of validity of their product (Geringer et al. 2013). Determining reliability by means of comparisons of controlled cases leads us undoubtedly to the field of test beds.

4. Test beds:

4.1. Definition:

A test bed is a system that makes possible to observe and measure the performance of a product under controlled and parametrical use conditions. This term is used in multiple fields to describe a protected experimentation environment which avoids the risk of an actual use of the tested

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

products which are normally in a development phase. Test beds are a rigorous, transparent and reproducible testing procedure not only for industrial products, but also for new technologies, computational elements and even scientific theories. They range from continuous improvement of new prototypes, to machines or complex systems in manufacturing industries such as the automotive or aeronautic industry (Márquez & Rojas 2011), and even to the intellectual characteristics refinement in fields such as the software products development.

Testbeds are widely used in industry and represent a significant part of the budget intended for the products design and development. Tests basically verify the correct performance of products not only in intermediate stages of their development, but also when their design is definitive. In big firms, test beds are part of the experimental general strategy, usually performed by the company itself, despite the fact that it can also be totally or partially commissioned to specialized laboratories.

Characteristics and components of a test bed depend on the format of the product to be tested. A typical one includes apparels, applications and mechanisms for networking. When a specific test demands specific characteristics or even specific models, the set composed by all these components is called test environment. However, it can also be a completely virtual environment (Lam 2007).
Testbeds are equipped with an instrumentation which tests the product in operating conditions, measuring its performance by means of a series of devices. Semiautomatic test beds are those which require the participation of an operator to perform the measures. Automatic test beds are those which can control directly the measuring tools and can run the tests autonomously.

The existence and employment of test beds benefits not only developers but also costumers, since they permit not only verifying the safety of the product, but also guaranteeing that the developed products offer optimal results or, at least, those previewed by the developer (Márquez & Rojas 2011).

4.2. Design principles and criteria:

When producing a test bed it is important to have not only a good design but also precise guidelines for future users (Lindvall 2005). All the tests must be easy to perform following the recommendations about how to develop it, how to collect the information and how to write the final report (Lam 2007). That enables novel users with reduced knowledge about scientific research to take their first steps with test beds. Homogeneity when presenting the results will make easier to compare results obtained by other researchers with the consequent enrichment of the debate (Brat et al. 2004). This sort of user guide of the test bed should help potential users to become familiar with its management by means of clear guidelines or a section with frequent answered questions (Basili et al. 2001). The lack of good documentation increases the amount of necessary work which is required to run the test bed and might become one of the reasons for its failure (Tichelaar et al. 1997). Another good principle is to conceive it as an open access product, not only allowing access to it but also designing it in such a way that its employment doesn't mean a significant expense (Stone 2003). If the experimentation cost of a test bed is too high, users will be reluctant to use it and the designer's work will have little impact and will lose the chance of receiving comments and appreciations from others (Brat et al. 2004). Finally, a test bed designer should include guide lines about how to assess numerically results since one of the main objectives of users having tested different products is to know which one is the best.

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

Conclusions:

Proficiency in the use of a structural analysis and design software application is important for architects in countries such as Spain, where they have professional competences in building structures. Building typologies and their inherent structures have evolved along with society, increasing their complexity and hyperstaticity. This process has been enabled by the breakthrough in different subjects such as Mathematics, Physics, Materials Science, and fundamentally in recent times by the development of computers and software applications.

Dating back to the late 1960s, computer-assisted structural analysis and design has always kept a sequence similar to the traditional manual methods and composed by: preprocessing or modelling, processing or analysis, postprocessing or design and deliverables management. Commercial software is usually marketed according to these four stages or working modules. When applications are publicized most of the emphasis is put on the variety of typologies that can be calculated and on the possibilities of the deliverables management, but few or no comments about the reliability of the results provided are made.

Professors or practitioners looking for a structural analysis and design software are not only interested on functionality but also on reliability. Information sources about this last aspect are scarce and controversial. So is scientific bibliography which in this specific field evinces a decreasing interest in this matter, making researchers step back to papers published more than twenty years ago and feel the need of exploring neighboring fields.

When talking about reliability of any software application for structural analysis and design three different aspects have to be taken into account: reliability of the structure, reliability of the computerized analysis and design process, and reliability of the software application. Being the first one controlled by the stochastic methods compiled in the national codes which set the ground for the software programming, and being the second one controllable by means of a good theoretical knowledge and a careful usage of the application, just the third one is a complete mystery difficult to unravel. Current national codes assume that building structures analysis and design software operates perfectly and no guidance to test them is officially provided.

Nevertheless, other neighboring industrial sectors and scientific disciplines have met the challenge of assessing even numerically the performance and reliability of their products, tools, applications and theories. A quick look to their methods brings us to the world of test beds whose principles and criteria might guarantee quality tools to assess the reliability of building structures analysis and design software and hence minimize the unavoidable risk implicit in designing a structure, analyzing it, building it, and finally putting it to service.

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

References

Basili, V.R., Tesoriero, R., Costa, P. et al. (2001). "Building an experience base for software engineering: a report on the first CeBASE eWorkshop." *3rd International Conference on Product Focused Software Process Improvement*, Springer-Verlag Berlin, Heidelberg, 110-125.

Bell, G. (1997). "Misuse of structural analysis software." *Journal of Computing in Civil Engineering*, 11(4), 215.

Biedermann, J. (1996). "Addressing current issues in structural design software." *Journal of Computing in Civil Engineering*, 10(4), 286-294.

Brat, G., Giannakopoulou, D., Goldberg, A., et al. (2004). "Experimental evaluation of verification and validation tools on Martian rover software." Formal Methods in Systems Design Journal, 25(2-3), 167-198.

Emkin, L. (1988). "Computers in structural engineering practice: the issue of quality." *Computers & Structures*, 30(3), 439-446.

Eskenasi, H. (1989). "Evaluation of software product quality by means of classification methods." *Journal of Systems and Software*, 10(3), 213-216.

Holick , M. y Vrouwenvelder, T. (2004). "Conceptos básicos de fiabilidad estructural." *Instituto de Ciencias de la Construcción Eduardo Torroja*, < http://alcala.ietcc.csic.es/fileadmin/Ficheros_IETcc/Web/Investigacion/IngenieriaEstructural/Leonardo/Cap_1-II__version1_.doc> (August 4, 2015)

Gendron, G. (1997). "A review of four PC packages for FE structural analysis." *Finite Elements in Analysis and Design*, 28(2), 105-114.

Gensichen, V., Lumpe, G. (2008). "Zur Leistungsfähigkeit, korrekten Anwendung und Kontrolle von EDV-Programmen für die Berechnung räumlicher Stabwerke im Stahlbau (Teil 1)." *Stahlbau*, 77(6), 447-453.

Geringer, J., Tuan, C., Lindsey, P. (2013). "Assessment of software for blast loading and structural response analysis using a lightweight steel-joist roof as a test case." *Journal of Performance of Constructed Facilities*, 27(2), 144-154.

Gracia, J. y Bayo, E. (2013). "Integrated 3D web application for structural analysis software as a service." *Journal of Computing in Civil Engineering*, 27(2), 159-166.

Hu, K., Yang, Y., Mu, S. et al. (2012). "Study on high-rise structure with oblique columns by ETABS, SAP2000, MIDAS/GEN and SATWE." *International Conference on Advances in Computational Modeling and Simulation (ACMS)*, Elsevier Science Bv, Amsterdam, 474-480.

ISO. (1998). *ISO 2394:1998 General Principles on Reliability for Structures*, International Organization for Standardization, Geneva.

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

Isreb, M. (1984). "Software and synthesis-oriented-structural analysis education." *Computers & Structures*, 18(4), 641-646.

Kao, C. y Yeh, I. (2014). "Optimal design of reinforced concrete plane frames using artificial neural networks." *Computers and Concrete*, 14(4), 445-462.

Lam, A.K. (2007). "Architecture and application of an autonomous robotic software engineering technology testbed (SETT)." Faculty of the Graduate School, University of Southern California, Los Angeles.

Langlais, S.J. (1985). "ENIAC, revisiting the legend." *American History Illustrated*, 20(6), 48-49.

Lindvall, M., Rus, I., Shull, F. et al. (2005). "An evolutionary testbed for software technology evaluation." *Innovations in Systems and Software Engineering – A NASA Journal*, 1(1), 3-11.

Márquez, F., Rojas, M. (2011). "Diseño y construcción de un banco de pruebas para motores monocilíndricos de cuatro tiempos a gasolina." *Repositorio Institucional de la Universidad Pontificia Bolivariana*, < http://repository.upbbga.edu.co:8080/jspui/bitstream/123456789/1679/1/digital_21165.pdf> (2 de agosto de 2015).

Melosh, L. y Utku, S. (1988). "Verification Tests for Computer Aided Structural Analysis." *Microcomputers in Civil Engineering*, 3(4), 289-297.

Ministerio de Vivienda. (2009). *Código Técnico de la Edificación (CTE). Seguridad Estructural. Libro 1*, Boletín Oficial del Estado, Madrid.

National Standards of the People's Republic of China. (2010). *Code for seismic design of buildings (GB 50011-2010) (English version)*, China Architecture & Building Press, Pequín.

Pellissetti, M. & Schueller, G. (2006). "On general purpose software in structural reliability – An overview." *Structural Safety*, 28 (1-2), 3-16.

Pereira, V.G., Barros, R.C., César, M.B. (2010). "Pushover analysis of a R/C frame by distinct software." *2nd International Symposium and Computational Mechanics & 12th International Conference on the Enhancement and Promotion of Computational Methods in Engineering and Science*, American Institute of Physics, Melville, 1618-1623.

Priest, J. (1988). *Engineering design for producibility and reliability*, Marcel Decker, Inc., Nueva York.

Ramírez de Dampierre, R. (1981). "La informática en la empresa constructora." *Informes de la Construcción*, 32(329), 5-16.

Redwine, S. y Riddle, W. (1985). "Software Technology Maturation." *8th International Conference on Software Engineering*, IEEE Computer Society Press, Los Alamitos, 189-200.

ISSN 2309-0103
www.enhsa.net/archidoct
Vol. 5 (2) / February 2018

Rojas, R. & Hashagen, U. (2000). *The First Computers – History and Architectures*, The MIT Press, Cambridge.

Rojiani, K., White, M., y Hemler, S. (1994). "Accuracy and reliability of structural analysis and steel design software." *2nd International Conference on Computational Structures Technology*, Civil-Comp Press, Edimburgo, 117-126.

Schueller, G. (2000). "Recent software developments for structural reliability assessment." *5th International Conference on Probabilistic Safety Assessment and Management*, Universal Academy Press, Inc., Tokio, 1229-1234.

Stone, P. (2003). "Multiagent competition and research: Lessons from RoboCup and TAC." *6th Robot World Cup Soccer and Rescue Competitions and Conference (RoboCup 2002)*, Springer-Verlag Berlin, Heidelberg, 224-237.

Tichelaar, S., Ducasse, S., and Meijler, T. (1997). "Architectural extraction in reverse engineering by prototyping: an experiment." *ESEC/FSE Workshop on Object-Oriented Reengineering*, Technical University of Vienna, Viena, 1-4.

Vitruvius, M. (25 BC). *Los diez libros de arquitectura*, Trad. Oliver, J.L. (1995), Alianza Editorial, S.A., Madrid.

Vora, V. (1986). "Selection of software." *ASCE Conference on Computing in Civil Engineering*, ASCE, Chicago, 870-874.

Wallace, D. y Fujii, R. (1989). "Verification and validation: Techniques to assure reliability." *IEEE Software*, 6(3), 8-9.

Wallace, D. y Fujii, R. (1989). "Verification and validation: An overview." *IEEE Software*, 6(3), 10-17.

Zhao, J. y Zhen, Z. (2013) "PKPM and SAP2000 software on a layer of engineering aseismic structure performance analysis based on structure mechanics." *International Conference on Material Engineering, Chemistry and Environment (MECE 2013)*, Trans Tech Publications Ltd., Zurich, 498-501.