Understanding the Limitations of Structural Engineering Software

By Clifford Schwinger, P.E., SECB and Eric J. Heller, E.I.T.

The use of engineering software is an important tool for efficiently analyzing and designing building structures; however, in order for engineers to effectively use computer software, they must understand the limitations of their software and know how to quickly validate the results with manual calculations.

This article discusses numerous examples of potential pitfalls associated with the use of computer software that the authors have encountered, and provides suggestions for managing software within an office.

Mid to large firms should designate in-house experts for each program used. These individuals are responsible for understanding how their programs work, keeping abreast of updates, training the rest of the office and answering questions.

Understanding software assumptions and limitations is crucial to avoiding problems. While computers are good at bulk analysis and design, they only are capable of accomplishing the specific tasks for which they were programmed. Knowing the limit states not considered by the software is crucial to ensuring safe and complete structural design. Note that this article focuses solely on understanding software limitations, and does not address human error in data input.

Software issues typically fall into one of four categories:

- Incorrect or misunderstood default settings
- Conditions not considered by the software
- Constructability issues
- Programming errors and idiosyncrasies

Incorrect or Misunderstood Default Settings

Misunderstanding software defaults can lead to mistakes varying from minor to catastrophic. A single set of default settings should be used office-wide, and these defaults should not be modified without consent of the engineer in charge. When software is upgraded to a newer version, default settings must be reviewed to ensure that they have been copied properly from the previous version, and that no new defaults have been added.

Conditions Not Considered By the Software

Conditions not considered by the software include limit states or load path issues that the program does not check. Software user's manuals do not dwell on software shortcomings, and lists of things not analyzed or designed are usually not provided.

Some typical examples of items not checked by most analysis and design software are:

- Column bracing requirements: Do members bracing columns have sufficient strength and stiffness?
- Slab on metal deck capacity: Does the slab on metal deck have sufficient strength to span between beams?
- Floor diaphragm strength and stiffness: Do floor diaphragms have sufficient strength and stiffness to transfer loads to the lateral load resisting system, and are connections between diaphragms and the lateral load resisting elements sufficient? Is the distribution of lateral loads to the lateral load resisting system resulting from a "rigid diaphragm" assumption a realistic one? *Figure 1* illustrates how a computer analysis distributed lateral loads to the shear walls and moment frames in a precast concrete parking structure. A rigid diaphragm default setting was used in the computer analysis. While the structure did have substantial torsional stiffness, engineering judgment dictated that the computer analysis resulted in too little load going into the moment frame. The design was revised to require the moment frame to carry substantially more lateral load than the computer analysis required.
- Drag struts: Are drag struts required to transfer loads from floor diaphragms to lateral load resisting elements?
- Wind girt design: Were wind girts designed to resist lateral wind loads?
- Connection design: Are connections designable without requiring expensive details such as web reinforcing plates, stiffeners, etc.?
- Connection workpoints: Are connection workpoints assumed by the software the same as the workpoints assumed by the engineer and indicated in the details on the Contract Documents?
- Concrete column load transfer through floor slabs: Where column concrete strength is higher than the floor slab compressive strength, will the slabs have sufficient strength to transfer the column loads through the floors?

Constructability Issues

All engineers should review their designs for constructability. Computer software will generally not consider constructability issues unless those issues are addressed indirectly in the default settings.

Some typical constructability issues include:

- Reinforcing steel in concrete columns: For economy and ease of construction, try to limit the percentage of steel in columns to 2 percent.
- Top reinforcing steel in concrete slabs perpendicular to slab edges: Select bars such that hooked bars at slab edges can be easily installed in thin slabs. Hooks on larger bars will hinder installation of thin slabs.

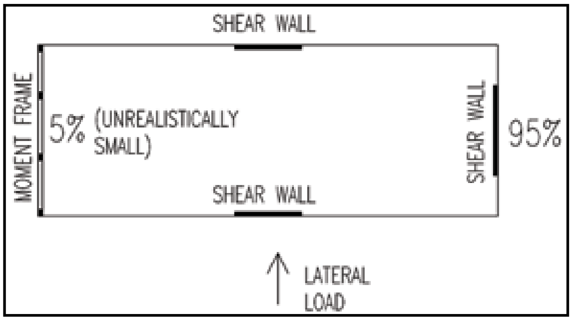

Figure 1: Illustration of unrealistic computer-generated lateral load distribution to shear walls and moment frame.

- Use of commonly available reinforcing steel: Use Grade 60 reinforcing steel, unless availability of grade 75 is confirmed.
- Review constructability of connections for steel and cast-in-place concrete construction: Constructability of connections is a whole topic in and of itself. Suffice to say, computers are capable of designing any imaginable configuration of framing; however, a review must be performed to understand whether connections can be accomplished in an efficient and economical manner.
- Standardization of reinforcing steel configurations: The optimal least-weight reinforcing steel arrangement generated by a computer analysis may not always be the least-cost configuration.

Programming Errors and Idiosyncrasies

Computer programs occasionally have flaws. Engineers need to be familiar with these flaws and understand how to work around them. Finding software flaws can be challenging, and when they are discovered, all engineers using the software must be alerted to them.

An example of a flaw of this type is one in which a program incorrectly computes deflections at the ends of cantilevered beams when the cantilevers are supported by transverse girders. Some programs do not consider effects of the girder deflections when computing the deflection at the tip of the cantilever and, accordingly, can substantially underestimate the cantilever deflection (*Figure 2*).

Conclusion

No structural engineering analysis and design software is perfect. Understanding the methodology and assumptions used by the software and the default settings available is crucial to efficiently and effectively using the program to design building structures. That said, manual checks of computer results are essential to verify the accuracy of the analysis. The next QA Corner article will discuss quick and easy methods of validating the results of computer analysis and design.▪

*Clifford Schwinger, P.E., SECB is a Vice President at The Harman Group's King of Prussia, PA office where he is the Quality Assurance Manager. He may be reached at **cschwinger@harmangroup.com**.*

*Eric Heller, E.I.T. is a Design Engineer at The Harman Group. He may be reached at **eheller@harmangroup.com**.*

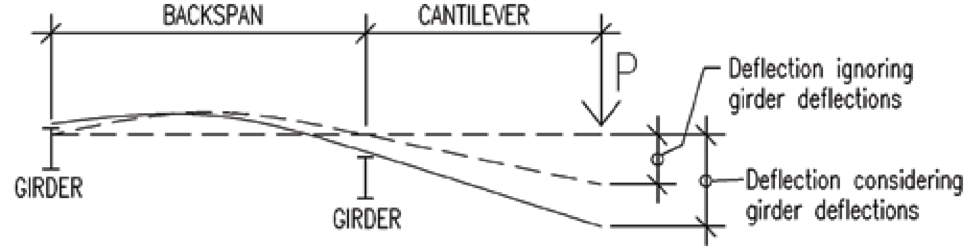

Figure 2: Illustration showing influence of girder deflections on deflection at end of cantilevered beam.