Proceedings of the
Annual Stability Conference
Structural Stability Research Council
Denver, Colorado, March 22-25 2022

Verification and validation of a nonlinear analysis model for ponding

Mark D. Denavit[1], Minjie Zhu[2], Michael H. Scott[3]

Abstract

Structural demands from rain loads can control the design of roof systems, especially roofs with large, flat bays where the accumulation of water due to deflections, known as ponding, amplifies the load. Under ponding conditions, insufficient strength or stiffness of the roof system can lead to instability and collapse. While procedures for elastic analysis of ponding are available, few models consider material or geometric nonlinearity in addition to the ponding nonlinearity. A recently developed analysis model that uses the OpenSees finite element framework to capture material, geometric, and ponding nonlinearity is verified and validated through comparisons to previously published analytical and experimental results, including large-scale experimental tests on open web steel joist roofs. These comparisons demonstrate that the analysis model is efficient and accurate in capturing the behavior of roofs subjected to ponding conditions up to incipient collapse.

1. Introduction

The 2022 edition of ASCE/SEI 7 *Minimum Design Loads and Associated Criteria for Buildings and Other Structures* requires explicit assessment of ponding head, the additional depth of water in the design condition due to roof deflections, for the determination of rain load (ASCE 2022). In contrast, previous editions of ASCE/SEI 7 did not include the ponding head in the rain load and ponding stability was ensured through a separate investigation. The new requirements will drive demand for structural analyses capable of capturing ponding effects. Engineers will need to perform elastic ponding analyses to assess rain loads and researchers will need to perform more advanced ponding analyses that account for material or geometric nonlinearity to build knowledge on the behavior of roof systems subject to ponding loads.

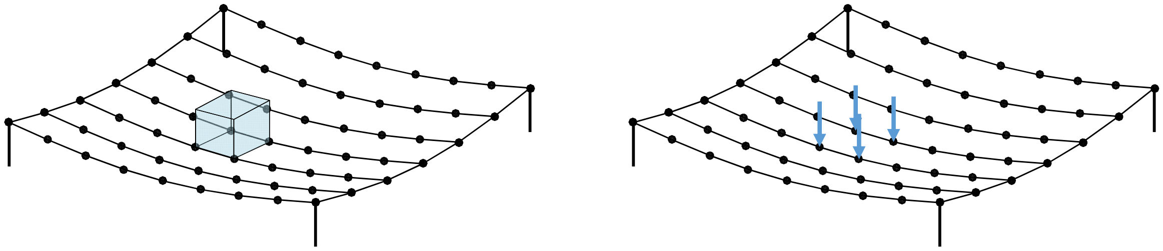
Several elastic ponding analysis methods have been proposed (Marino 1966; Colombi 2006; Baber and Rigsbee 2010; Silver 2010; Denavit 2019). However, few analysis methods consider material or geometric nonlinearity in addition to the ponding nonlinearity. The authors developed an analysis model, built upon the OpenSees finite element framework, that captures material,

[1] Assistant Professor, University of Tennessee, Knoxville, <mdenavit@utk.edu >
[2] Research Associate, Oregon State University, <zhum@oregonstate.edu>
[3] Professor, Oregon State University, <michael.scott@oregonstate.edu>

geometric, and ponding nonlinearity. The objective of this work is to verify and validate this model against existing analysis methods and experimental results.

2. OpenSeesPy and PyPonding

The ponding analysis model is implemented in the OpenSees framework and thus can leverage the wide range of constitutive relations, element formulations, and solution algorithms within the framework. The key aspect of the model is the computation of loads based on the water level and current deformed state of the structure. The roof is divided into "ponding load cells" connected at finite element nodes. Implemented within the load cell are procedures to calculate the volume of water above the cell and convert its weight to loads on the structure (Figure 1). The procedures are based on the work of Colombi (2006) with extensions to allow for quadrilateral (not only rectangular) load cells and automatic evaluation of the overlap of water and snow load as described in SJI Technical Digest 3 (Fisher and Denavit 2018).



(a) Physical representation of water on one ponding load cell. The volume of water is computed based on the deformed shape of the roof.

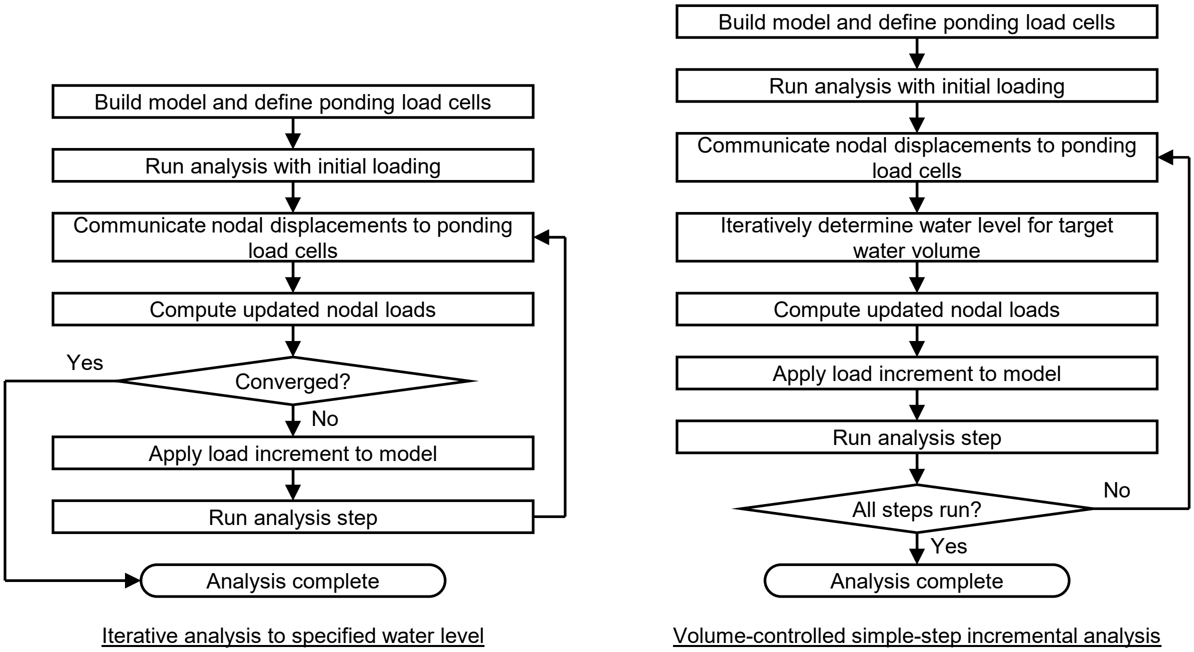
(b) Nodal point loads on the roof structure computed by the ponding load cell to represent the physical volume of water.

Figure 1. Schematic showing the conversion of a physical volume of water to point loads for one ponding load cell on a small roof structure. Multiple ponding load cells cover the roof.

The ponding load cells are implemented in the PyPonding Python package and thus can interoperate with the OpenSeesPy Python interpreter for the OpenSees framework (Zhu et al. 2018). Two-dimensional and three-dimensional versions of the ponding load cells are implemented. PyPonding also includes ponding load manager classes that aggregate several ponding load cells and automate common functions between finite element analysis and ponding, such as communicating deformations and assembling the load vector.

The ponding load cells allow for a variety of analysis types. Figure 2 describes the procedure for two types of analysis. The first type of analysis described in Figure 2 achieves equilibrium for a given water level iteratively. Such an analysis may be appropriate for determining rain loads per ASCE/SEI 7 (ASCE 2022). The second type of analysis described in Figure 2 incrementally ramps up the volume of water without iteration in individual load steps. This type of analysis may be appropriate for cases where the water level reaches a peak since controlling the volume allows for analysis past this peak. The model and this analysis type have been used to evaluate the strength and reliability of steel roofs under rain loads (Denavit and Scott 2021). Other analysis types such as iterative analysis to a specified water volume, which may be appropriate for determining rain load per Canadian standards (NRCC 2015), are also possible.

Note that volume-controlled analyses include a step where the water level is determined iteratively using a standard Newton algorithm to achieve a target volume for the current deformed shape of the structure. The ponding load cell classes include procedures to return total water volume and water surface area (i.e., the ratio of change in water volume to change in water level or *dV*/*dz*), to accommodate this iterative procedure.


Build model and define ponding load cells
Run analysis with initial loading
Communicate nodal displacements to ponding load cells
Compute updated nodal loads
Converged?
Yes
No
Apply load increment to model
Run analysis step
Analysis complete
Iterative analysis to specified water level
Build model and define ponding load cells
Run analysis with initial loading
Communicate nodal displacements to ponding load cells
Iteratively determine water level for target water volume
Compute updated nodal loads
Apply load increment to model
Run analysis step
All steps run?
No
Yes
Analysis complete
Volume-controlled simple-step incremental analysis

(a) (b)

Figure 2. Flowchart for two types of ponding analysis.

3. Verification Against Closed-Form Solutions

Researchers have developed closed-form solutions of displacements and internal forces, including ponding effects, for idealized cases. These solutions, especially those by Marino (1966), have a long history of use in design. Verification against these solutions confirms that PyPonding and OpenSeesPy accurately represent ponding effects as described by these solutions.

3.1 Two-dimensional solutions by Silver (2010)

Silver (2010) derived closed-form solutions for deflection and bending moments for several cases. The basic case of a flat simply supported beam with initial uniform loading (i.e., the loading would be uniform without ponding effects) is investigated here.

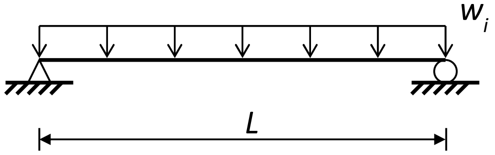


Figure 3. Flat simply supported beam with initial uniform loading as investigated by Silver (2010).

Silver identified that the loading on the beam is the sum of the initial uniform loading and the weight of the water, which is proportional to the deflection of the beam.

$$w(x) = w_i + \gamma\, S\, y(x) \tag{1}$$

where $w(x)$ is the total load on the beam, w_i is the initial uniform load (force per length), γ is the specific gravity of water (62.4 lb/ft^3 = 3.61×10^{-5} k/in.^{3}), S is the beam spacing (tributary width), and $y(x)$ is the total deflection.

Assuming small displacements and elastic behavior, the governing differential equation is formed noting that loading is proportional to the fourth derivative of the deflection.

$$EI\, y''''(x) = w_i + \gamma\, S\, y(x) \tag{2}$$

Silver derived the solution to this differential equation as:

$$y(x) = \frac{w_i}{2\gamma S}\left[\left(\frac{\cos\left(\sqrt[4]{\frac{\gamma S}{EI}}\left(\frac{L}{2}-x\right)\right)}{\cos\left(\sqrt[4]{\frac{\gamma S}{EI}}\left(\frac{L}{2}\right)\right)}-1\right)-\left(1-\frac{\cosh\left(\sqrt[4]{\frac{\gamma S}{EI}}\left(\frac{L}{2}-x\right)\right)}{\cosh\left(\sqrt[4]{\frac{\gamma S}{EI}}\left(\frac{L}{2}\right)\right)}\right)\right] \tag{3}$$

Silver further derived the bending moment, noting that it is proportional to the second derivative of deflection. The flexibility coefficient, C, is a key parameter for these solutions.

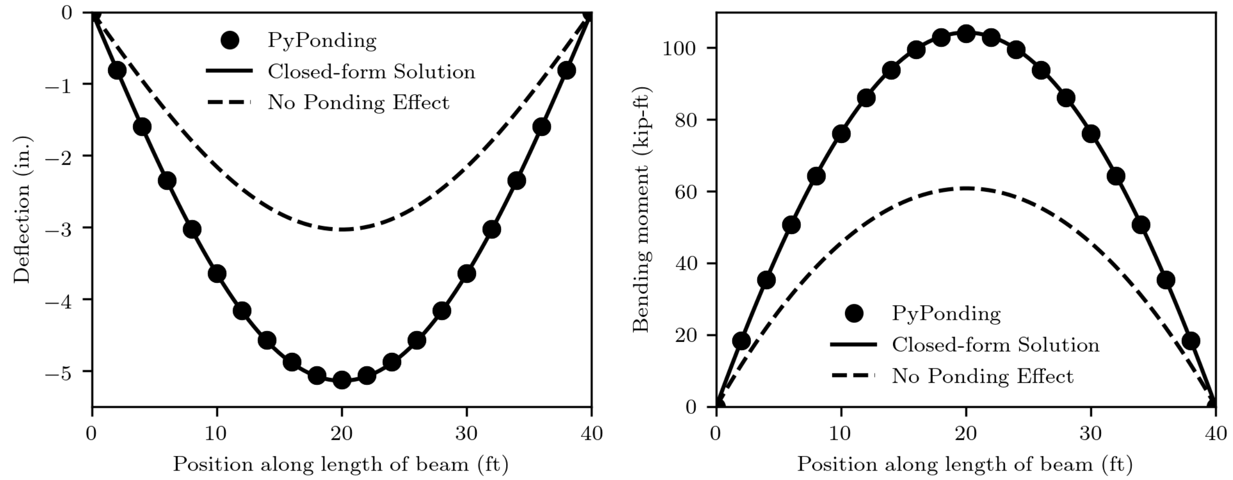
$$C = \frac{\gamma S L^4}{\pi^4 EI} \tag{4}$$

Note that as C approaches unity, the deflections and bending moments diverge to infinity.

Analysis results for the case of a W14×22 beam (I = 199 in^4) spanning 40 ft and with a spacing of 10 ft are shown in Figure 4. The loading for this analysis consists of 20 psf dead load and water at a 2 in. level above the undeflected surface of the roof. Three sets of results are shown. The first set of results is from Silver's closed-form solutions (e.g., Eq. 3 for beam deflection). The second set of results is from an analysis using PyPonding and OpenSeesPy. For this analysis, 20 elastic beam elements were used along the length of the beam and linear geometric transformations were employed. The PyPonding results match those from the closed-form solution. Both results exhibit significantly increased deflection and bending moment in comparison to the third set of results which does not include ponding effects and were obtained from standard beam equations, specifically Case 1 of Table 3-23 of the AISC *Manual* (AISC 2017).

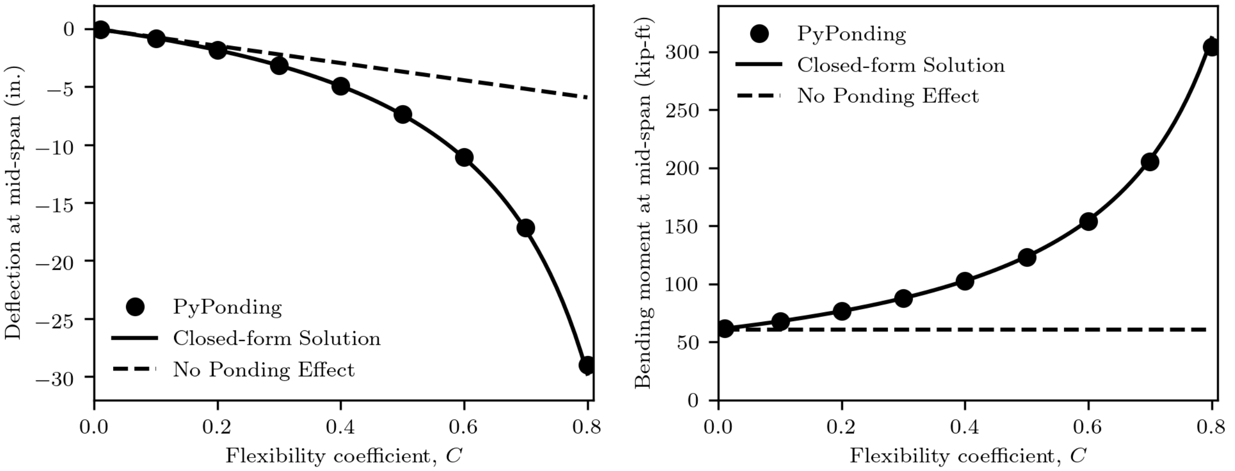
The flexibility coefficient, C, equals 0.424 for the case presented in Figure 4. Additional comparisons over a range of flexibility coefficients are shown in Figure 5. The beam length, beam spacing, modulus of elasticity, and loading were the same as for Figure 4. Only the moment of inertia of the beam varied. Again, the results agree, indicating that PyPonding with

OpenSeesPy is capable of accurately capturing the effects of ponding even for large flexibility coefficients that lead to large ponding loads in comparison to the initial uniform loading.



(a) (b)

Figure 4. Analysis results for a X 25ð33 beam (C = 0.424).



(a) (b)

Figure 5. Analysis results for cases with various flexibility coefficients.

3.2 Three-dimensional solutions by Marino (1966)

Marino (1966) derived a closed-form solution for deflections and bending moments of an idealized rectangular bay consisting of primary members supporting secondary members. These solutions formed the basis for the methods of design for ponding that appear in Appendix 2 of the AISC *Specification* (AISC 2016). The bay is perfectly flat with no slope or camber. The primary members have equal stiffness. The secondary members have equal stiffness, are equally spaced, and are spaced closely enough that the primary members can be considered uniformly loaded. All members are simply supported, and adjacent bays are identical.

The main results of Marino's derivation are the maximum deflection of the primary members and the secondary members due to the ponding loads. Using these results, the maximum moment in the primary member, $M_{p,max}$, and maximum moment in the secondary members, $M_{s,max}$, are:

$$M_{p,max} = \left(\frac{\left(q_i L_s\right) L_p^2}{8}\right)\left(1 + \frac{\alpha_p \left(1 + \frac{\pi}{4}\alpha_s + \frac{\pi}{4}\rho\left(1+\alpha_s\right)\right)}{1 - \frac{\pi}{4}\alpha_p \alpha_s}\right) \tag{5}$$

$$M_{s,max} = \left(\frac{\left(q_i S\right) L_s^2}{8}\right)\left(1 + \frac{\alpha_s \left(1 + \frac{\pi^3}{32}\alpha_p + \frac{\pi^2}{8\rho}\left(1+\alpha_p\right) + 0.185\alpha_s \alpha_p\right)}{1 - \frac{\pi}{4}\alpha_p \alpha_s}\right) \tag{6}$$

where q_i is the initial uniform load (force per area), L_p is the length of the primary members, L_s is the length of the secondary members, S is the spacing of the secondary members, $\alpha_p = C_p/(1–C_p)$, $\alpha_s = C_s/(1–C_s)$, and $\rho = C_s/C_p$. The flexibility coefficients for primary and secondary members are

$$C_p = \frac{\gamma L_s L_p^4}{\pi^4 E I_p} \tag{7}$$

$$C_s = \frac{\gamma S L_s^4}{\pi^4 E I_s} \tag{8}$$

where I_p is the moment of inertia of the primary members and I_s is the moment of inertia of the secondary members.

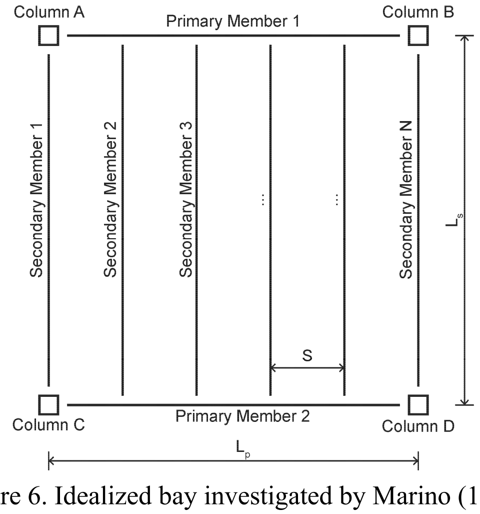
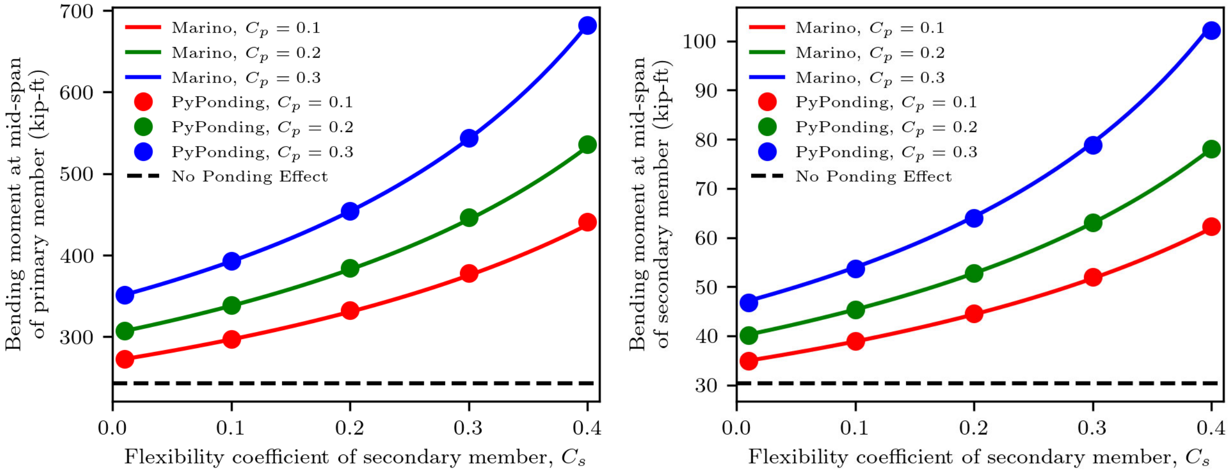


Figure 6. Idealized bay investigated by Marino (1966).

The maximum moment in the primary and secondary members for a variety of flexibility coefficients are shown in Figure 7 for a case with $L_p = L_s = 40$ ft, $S = 5$ ft (8 secondary member spaces), $E = 29000$ ksi, 20 psf dead load and 2 in. of water above the undeflected surface of the roof. As before, three sets of results are shown. The first set is from Marino's closed-form solutions (i.e., Eqs. 5 and 6). The second set of results is from analyses using PyPonding and OpenSeesPy. The third set of results is based on standard beam theory and does not include the ponding effect and thus shows no variation with the flexibility coefficients.

To accommodate the symmetry at each edge of the bay, individual members are analyzed separately with PyPonding and OpenSeesPy. Once the water load is determined, analyses are performed for each of the secondary members with pin and roller boundary conditions. The loads are doubled on the secondary members at the edges of the bays to account for the loads applied from the adjacent bays. The reactions from the analyses of the secondary members are recorded, doubled (to account for the adjacent bays), then applied as loads in the analyses of the primary members. Deformations from analyses of primary and secondary members are combined to determine the total deformations that the ponding load cells use to update the water load. For these analyses, eight elements were used along the length of the primary members (matching the number of secondary member spaces) and 20 elements were used along the length of the secondary members. Elastic beam elements with linear geometric transformations were employed.



(a) (b)

Figure 7. Analysis results for cases with various flexibility coefficients.

The PyPonding results match those from the closed-form solution. Both results exhibit significantly increased bending moment compared to the third set of results which do not include ponding effects. As before, these results indicate that PyPonding with OpenSeesPy is capable of accurately capturing the effects of ponding even for highly flexible framed roof systems.

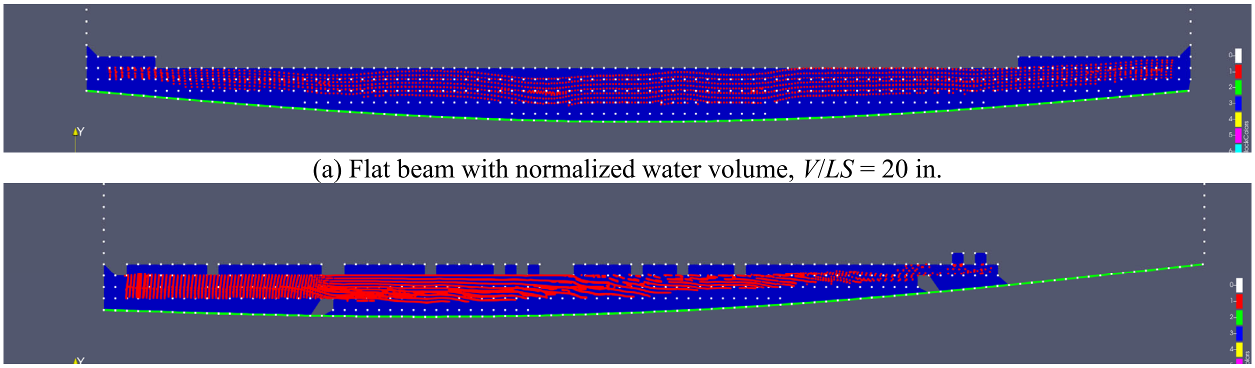
4. Validation Against the Particle Finite Element Method

Ponding models based on static, mechanical loading of structural members neglect the effects of fluid flow and pressure in computing the structural response. The particle finite element method (PFEM) is an approach to fluid-structure interaction that forms and solves the governing equations in a monolithic system rather than using a staggered solution along the fluid-structure boundary. Due to its Lagrangian formulation, the PFEM integrates with the structural finite elements implemented within OpenSees (Zhu and Scott 2014). In addition to the physical experiments of the next section, PFEM simulations can validate the ponding analysis method since they provide an accurate representation of the physics of ponding that is conceptually distinct from the representation upon which PyPonding and the closed-form solutions are based.

In the first validation, the ponding analysis method is compared to an OpenSees PFEM analysis of a flat beam like shown in Figure 3. As before, the beam is a X 25ð33 (I = 199 in^4) spanning 40 ft and with a spacing of 10 ft. The loading for this analysis consists of 20 psf dead load and a given volume of water. Three PFEM analyses were performed with normalized water volumes (i.e., V/LS) of 20 in., 30 in., and 40 in., respectively. This amount of water would cause the beam to yield and subsequently collapse, however, both the PyPonding and PFEM analyses assumed linear elastic beam behavior thus collapse did not occur. The large volumes of water were selected to allow for a coarser mesh in the PFEM analyses and reduced run times. Unlike PyPonding, the PFEM analyses are dynamic. Each analysis was performed for 10 seconds to reach steady state. A snapshot at the end of one of the analyses is shown in Figure 8(a). The red points are fluid particles, the blue mesh is the background finite element mesh, and the green line is the beam.

Detailed results are compared in Figure 9. Water level is not an explicit output from the PFEM analyses, and the water particles were not in a strict horizontal line at the end of the analyses as assumed in PyPonding. Water level from the PFEM analyses was taken as the nearest mesh line. Deflections, bending moments, and shears along the length of the beam match; however, the axial load in the beam does not match. In PyPonding, water loads are always applied vertically, therefore with a flat beam and linear geometric transformations, the axial load in the beam will always be zero. Physically, water pressure acts perpendicular to the beam. Thus, as the beam deflects, a small horizontal component of the water load arises, resulting in tension in the beam. The PFEM analyses capture this behavior through interaction between the fluid and structure even when the underlying structure is modeled with linear geometric transformations.

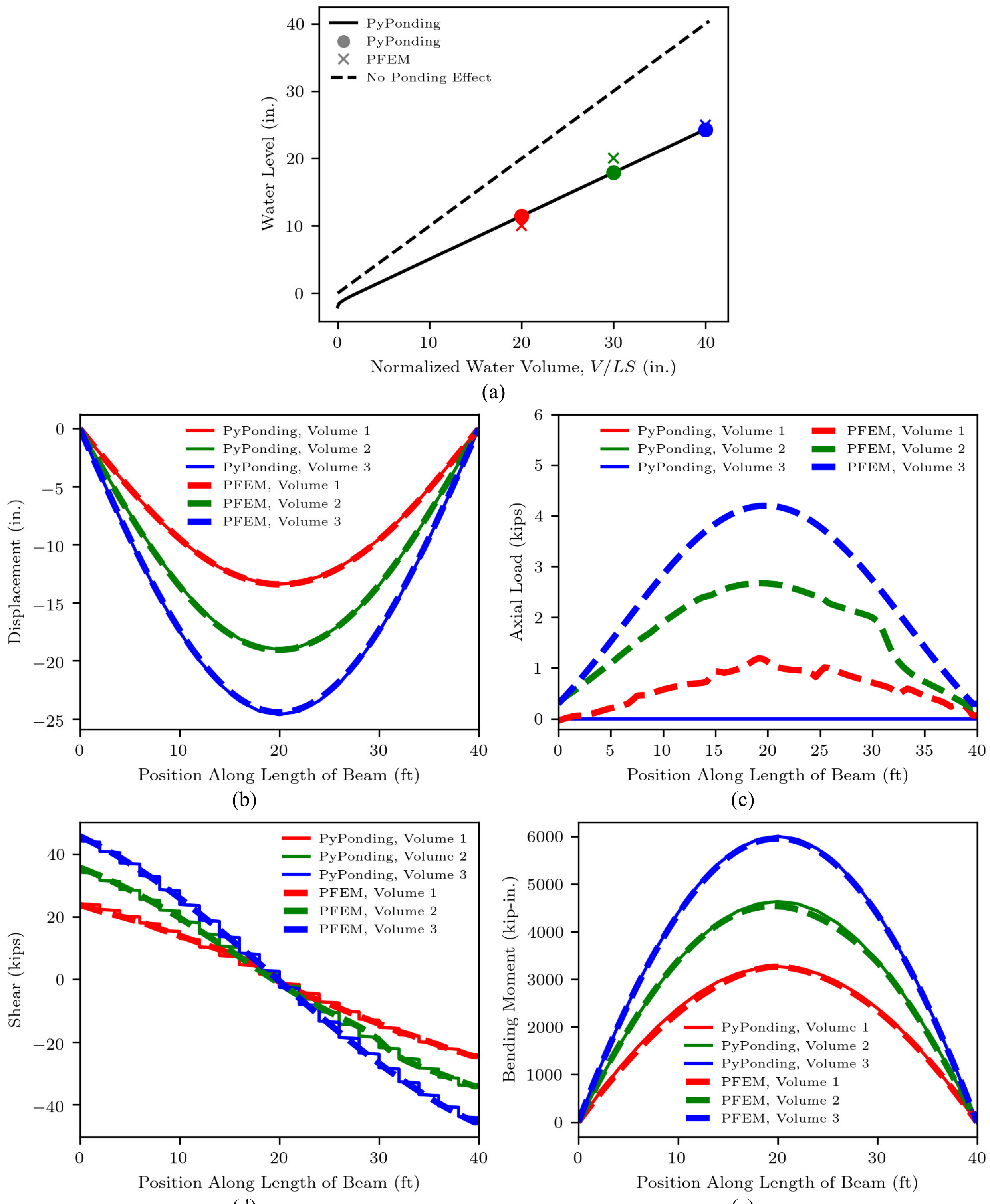
The second validation is for a sloped beam, which has the same properties as the flat beam except the right support is 20 in. higher than the left support (i.e., 1/2 in. per foot slope). A single PFEM analysis was run for the sloped beam with normalized water volume, V/LS = 15 in. The analysis was performed for 30 seconds to reach steady state. A snapshot at the end of the analysis is shown in Figure 8(b). Note that the right end of the beam is above the water level. Detailed results are compared in Figure 10. Again, the results match except for axial force due to the realistic simulation of fluid pressures normal to the deflected shape of the beam.



(a) Flat beam with normalized water volume, V/LS = 20 in.

(b) Sloped beam with normalized water volume, V/LS = 15 in.

Figure 8. Snapshots of PFEM ponding analysis results at end of analysis.



(a)

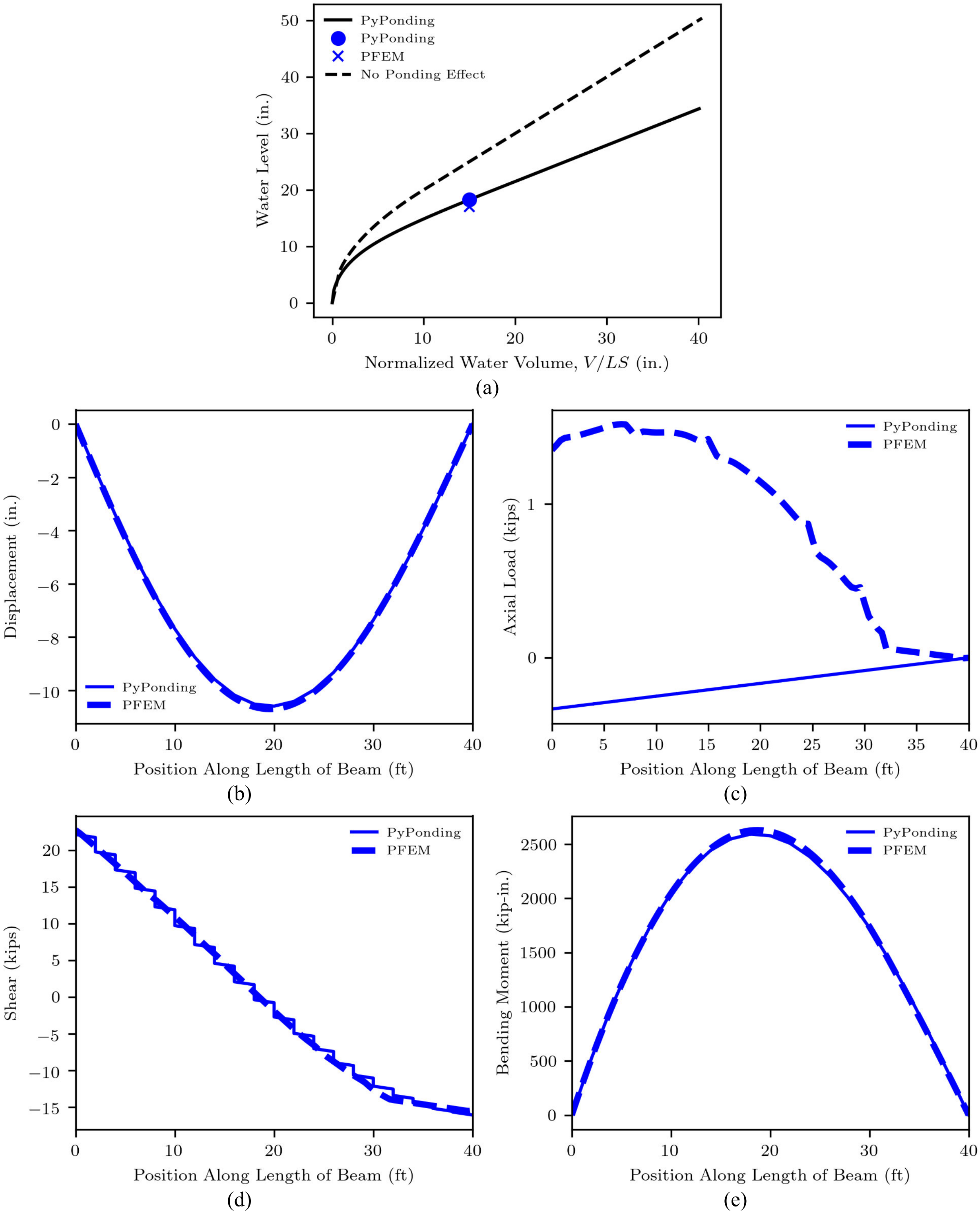
(b)

(c)

(d)

(e)

Figure 9. Comparison of PyPonding and PFEM results for a flat beam.



(a)

(b)

(c)

(d)

(e)

Figure 10. Comparison of PyPonding and PFEM results for a sloped beam.

5. Validation Against Physical Experiments

Due to fluid loading conditions, physical testing of roof ponding is challenging. The authors are aware of only one set of physical tests which have been performed on full-scale roof structures subjected to ponding loads (Stark 2009; Stark et al. 2010). In this series of tests, two open-web steel joist roofs were loaded with water until failure. The roofs were constructed with three joists each, a 24K9 center joist and two edge joists. The span of the joists was 48 feet, and the width of the roof was 12 feet. The two edge joists were designed not to fail and to be half as stiff as the center joist to approximate a one-way bending condition. One of the roofs was constructed without slope while the other "pitched" roof was constructed with a 1/4 in. per foot slope. All joists were constructed with approximately 1 in. of camber.

Both specimens were analyzed using PyPonding and OpenSeesPy. The analyses were performed in three-dimensions. The center joist was modeled using mixed beam finite elements for the top chord and truss elements for the bottom chord and web members. Two beam elements were used in each unbraced length of the top chord for a total of 48 elements along the length of the joist. Roof slope and camber were modeled explicitly through the definition of nodal coordinates. Out-of-plane deformations of the joist were restrained in the model as they were in the physical specimens by bridging. Fiber cross sections were used for the double angle top chord with 23 fibers along the height of the section and a bilinear stress strain relationship. Measured yield strength was used for the chord and web members of the center joist with an assumed $E/200$ hardening modulus. The edge joists were modeled with elastic beam elements with the reported moment of inertia of the joist since details of the chord and web members of the edge joists were not reported. The magnitude of dead load was also not reported in the experiments however it was noted that the specimens were initially nearly planar. The dead load was assumed to be 5 psf as deflections due to this level of dead load offset the camber.

Results of the comparisons are shown in Figure 11. Two sets of analysis results are presented one in which corotational (i.e., nonlinear) geometric transformations were used for the beam and truss elements, and one where linear geometric transformations were used. Little to no post-peak response is observed in the analyses using the nonlinear geometric transformations. While the ponding analysis overall is volume-controlled, each OpenSeesPy analysis step was performed in load control. The model is likely reaching a limit point at the maximum water level and thus is unable to return convergent results. The development of more sophisticated solution algorithms for ponding analysis is recommended for future research. Given the configuration and relatively low slenderness of the top chord ($L/r_x \approx 39$), no difference is seen between the analyses results using nonlinear and linear geometric transformations up to the maximum water level, however the lack of negative geometric stiffness allows the analysis with linear geometric transformation to converge for a few volume steps past the maximum water level.

The analysis model can predict with high accuracy the experimentally observed response of the roof up to the point of collapse, indicating that the ponding effects are well-captured by the model. For both the physical experiment and the analysis, the failure mode was in-plane bending of the top chord. However, collapse occurs earlier in the analysis than it does in the experiment (i.e., for both specimens the analysis fails at a total water load of approximately 39 kips whereas the specimens failed at a total water load of 46 kips in the physical experiment). This discrepancy may be due to differences between actual and assumed values of parameters which were not reported by the experimentalists.

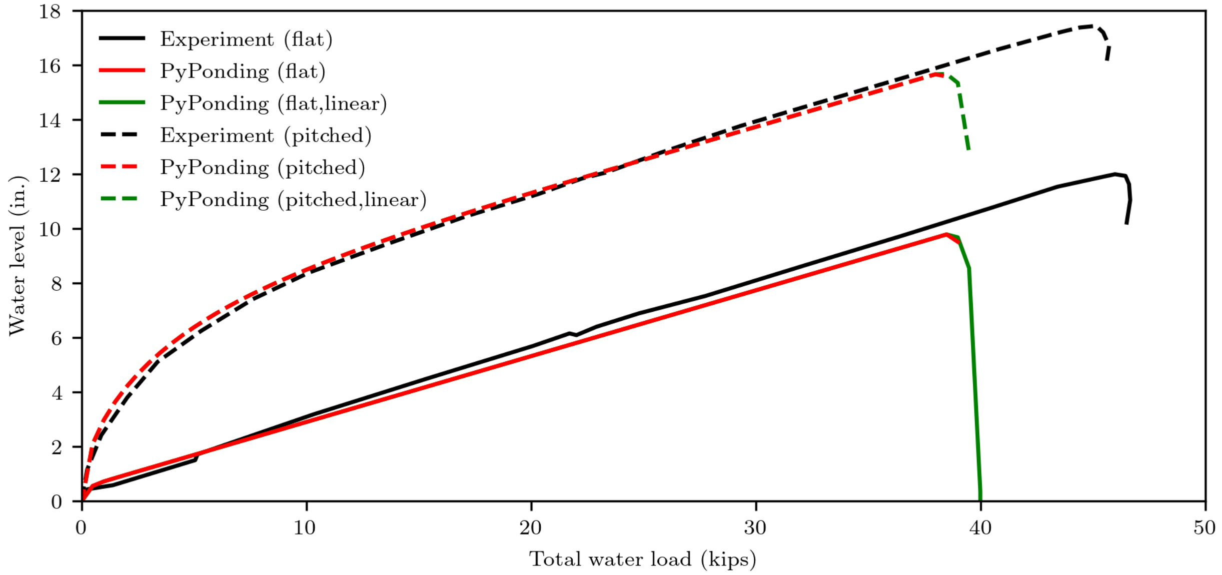


Figure 11. Model validation against large scale physical experiments of ponding on open-web steel joist roofs

6. Conclusions

Methods of structural analysis that include the nonlinear effects of ponding are needed to satisfy new design requirements and to further expand knowledge on the behavior of flexible roof systems. PyPonding is a newly developed Python package for ponding analysis. When coupled with a finite element framework, such as OpenSeesPy, PyPonding enables efficient ponding analyses that also consider other types of nonlinearities. Verification and validations studies were presented in this work, comparing PyPonding analysis results to those from closed-form solutions and physical experiments. The agreement of results for these challenging problems demonstrates the accuracy and capabilities of the PyPonding modeling approach.

7. Supplementary Material

The PyPonding package and numerical examples are available for download from GitHub at https://github.com/denavit/PyPonding. The OpenSeesPy package is also available from GitHub at https://github.com/zhuminjie/OpenSeesPy and can be installed directly using PyPi at https://pypi.org/project/openseespy/.

References

AISC. (2016). *Specification for Structural Steel Buildings*. American Institute of Steel Construction, Chicago, Illinois.

AISC. (2017). *Steel Construction Manual, 15th Edition*. American Institute of Steel Construction, Chicago, Illinois.

ASCE. (2022). *Minimum Design Loads and Associated Criteria for Buildings and Other Structures*. American Society of Civil Engineers, Reston, Virginia.

Baber, T. T., and Rigsbee, E. D. (2010). “Noniterative Finite Element Analysis of Ponding.” *Structures Congress 2010*, American Society of Civil Engineers, 1150–1159.

Colombi, P. (2006). “The Ponding Problem on Flat Steel Roof Grids.” *Journal of Constructional Steel Research*, 62(7), 647–655.

Denavit, M. D. (2019). “Approximate ponding analysis by amplified first-order analysis.” *Engineering Structures*, 197, 109428.

Denavit, M. D., and Scott, M. H. (2021). “Strength and Reliability of Structural Steel Roofs Subjected to Ponding Loads.” *Journal of Structural Engineering*, American Society of Civil Engineers, 147(2), 04020318.

Fisher, J. M., and Denavit, M. D. (2018). *Structural Design of Steel Joist Roofs to Resist Ponding Loads*. Technical Digest 3, Steel Joist Institute, Florence, South Carolina.

Marino, F. J. (1966). “Ponding of Two-Way Roof Systems.” *Engineering Journal*, AISC, 3(3), 93–100.

NRCC. (2015). *National Building Code of Canada 2015*. National Research Council Canada, Ottawa, Canada.

Silver, E. (2010). “A Strength Design Approach to Ponding.” *Engineering Journal*, AISC, 47(3), 175–187.

Stark, D. (2009). “Experimental and Analytical Investigation of Ponding Load Effects on a Steel Joist Roof System.” M.S. Thesis, Oregon State University, Corvallis, Oregon.

Stark, D., Higgins, C., and Green, P. S. (2010). “Stability of Open Web Steel Joists Subjected to Ponding Loads.” *Proceedings of the 2010 SSRC Annual Stability Conference*, SSRC, Orlando, Florida.

Zhu, M., McKenna, F., and Scott, M. H. (2018). “OpenSeesPy: Python library for the OpenSees finite element framework.” *SoftwareX*, 7, 6–11.

Zhu, M., and Scott, M. H. (2014). “Modeling fluid–structure interaction by the particle finite element method in OpenSees.” *Computers & Structures*, 132, 12–21.