Challenging Glass Conference Proceedings – Volume 9 – June 2024 – Louter, Bos & Belis (Eds.)
International Conference on the Architectural and Structural Application of Glass
Challenging Glass Conference 9 – 19 & 20 June 2024 – TU Delft – The Netherlands

W-Glass: A Cloud-Based User-Friendly Software Platform for Simplified EN 16612 Compliance

M. Burak Yilmaz [a], **Murat Yilmaz** [b]

a Macrostatic, Türkiye, mehmetburakyilmaz@macrostatic.com
b Istanbul Technical University, Department of Civil Engineering, Türkiye, yilmazmura@itu.edu.tr

Abstract

Published in 2019, EN 16612 represents a significant advancement in window glass standards, surpassing its predecessor, prEN13474. It introduces new methodologies for calculating triple-glazed insulating glass units, a detailed approach to determining the shear transfer coefficient, and refined formulations for design strength and cavity pressure variations. These enhancements, however, add complexity to the procedural application of EN 16612. In particular, the variable shear transfer coefficient, depending on loading conditions, influences multiple calculation steps, creating a complex implementation dependency chain. Additionally, the absence of explicit load combination specifications has led to variations in practical applications. This study aims to clarify the complexities and variations involved in implementing EN 16612, with a focus on procedural nuances from the end-user's perspective. It examines the architectural design of a cloud platform, W-Glass, which is specifically designed to streamline the application of this standard. The platform facilitates end-users in performing relevant calculations with ease. Among its various components, the study highlights the calculation service as a particularly crucial element. Employing state-of-the-art programming techniques, this service efficiently manages the structures required by the standard, ensuring a codebase that is not only clean and flexible but also maintainable for potential future enhancements of the standard.

Keywords

Glass, EN 16612, Cloud platform, Flexible code base

Article Information

- Digital Object Identifier (DOI): 10.47982/cgc.9.518
- Published by Challenging Glass, on behalf of the author(s), at Stichting OpenAccess.
- Published as part of the peer-reviewed Challenging Glass Conference Proceedings, Volume 9, June 2024, 10.47982/cgc.9
- Editors: Christian Louter, Freek Bos & Jan Belis

1. Introduction and Literature Summary

Window glass standards play a crucial role in ensuring the safety and performance of glazing systems in buildings. Over the years, these standards have evolved to address advancements in materials, technology, and construction practices. One of the significant milestones in this evolution is the publication of EN 16612 in 2019, which represents a significant advancement over its predecessors, prEN 13474-1:1999, prEN 13474-2:2000, prEN 13474-3:2009, prEN 16612:2013, prEN 16612:2017.

EN 16612 introduces new methodologies for calculating the load resistance of triple-insulating glass units, provides a detailed approach to determining the shear transfer coefficient and offers refined formulations for design strength and cavity pressure variations. While these enhancements improve the accuracy and reliability of glass design, they also introduce complexities in the procedural application of the standard.

Several studies have examined the implications of EN 16612 for architects, engineers, and other stakeholders in the construction industry. For instance, a comparative analysis by Siebert (2018) highlights the differences between DIN 18008, prEN 13474, and EN 16612, emphasizing the advancements made by the latter. Similarly, Morse and Norville (2016) present a series of case studies to identify common implementation challenges of EN 16612 and propose solutions to address them. Further studies are performed by Halilovič et al. (2023) to implement the triple and quadruple pane window glass.

In addition to practical challenges, there is also a growing interest in leveraging technology to streamline the application of EN 16612. Cloud-based solutions, in particular, have gained traction due to their ability to perform complex calculations efficiently. The authors discuss the benefits of using cloud computing technology for ensuring EN 16612 compliance and present a case study of a cloud-based platform developed for this purpose.

This study aims to build upon existing research by providing a comprehensive analysis of the complexities and variations involved in implementing EN 16612. It examines the architectural design of a cloud platform, Macrostatic W-Glass, which is specifically tailored to streamline the application of this standard. The platform facilitates end-users in performing relevant calculations with ease, addressing the challenges identified in previous studies.

2. A quick overview of EN 16612

This chapter explores the framework of EN 16612, highlighting the sequential dependence of each calculation step. It decomposes the standard into its essential components, detailing how various sections interrelate and impact the overall calculation process.

The methodology for determining glass thickness involves a trial-and-error approach, beginning with an initial estimation of thickness (M. Haldimann et. al 2008). Required inputs for calculations are summarized in Figure 1. After thickness selection, the section's stresses are assessed, and the design process may extend across multiple iterations. For conducting strength evaluations of the glass, identifying the glass geometry is a preliminary step. Analytical formulas catering to a wide array of geometries, ranging from rectangular, isosceles and, right-angled triangles, circular, other triangular shapes, trapezoidal, and arched edge rectangles are employed. (Feldmann 2014). Furthermore, the inclination angle of the glass, influenced by gravitational acceleration, plays a critical role in these calculations.

User Inputs

shape and dimensions of the pane
installation angle
support conditions
consequences class (CC0 to CC3)
h;i nominal thickness of any ply
s;i nominal cavity width of IGU
h_{int}, thickness of the interlayer
interlayer type (as per EN16613)
glass material per production
glass prestressing condition
glass edge finish type
deflection limits

wind load pressure
wind load suction
snow load
maintenanace load
barrier load
internal action (isochore pressures)
H, final installation altitute
H_p, altitute of production of IGU
$T_{c;i}$, cavity temperature
T_p, temp. of production of IGU
p_a, meteorological air pressure
p_p, met. air pressure at prod. of IGU

Database

$f_{g;k}$, $f_{b;k}$, characteristic bending strentgh
$\gamma_{M;A}$, $\gamma_{M;V}$, material partial factors
k_{sp}, glass surface profile factors
k_v, strengthening factors for manufacturing
k_e, edge strength factors
duration of loads
ω, coefficient of shear transfer of an interlayer
γG, γQ, ψ, factors to derive load combinations

Outputs

glass strength checks
glass deflection checks

Fig. 1: Basic summary of inputs, database, and outputs structure of the lateral load resistance calculation as per EN 16612.

Internal actions resulting from temperature changes and other variations in internal pressure do not create any support reaction since they balance each other out in the inner and outer glass layers (McMahon et al 2018). When calculating wind loads on an IGU supported from two opposite edges, analytical formulas are used for the supported-from-two-edges scenario (Galuppi and Carfagni 2020). However, for internal actions, different analytical formulas, four edge support, are required because the glasses transfer loads to each other through sealing silicone located on all four sides in the case of rectangular glass (Kozlowski 2023). The superposition of the result obtained from here with other loading conditions further complicates the solution (Lori et. al 2022).

Window panels that can be considered as infill walls correspond to Consequences class, CC0 (Badalassi et. Al. 2014). The partial load factor and combination factor values compatible with CC0 are provided in EN 16612.

For laminated glasses used in IGU glass units, the mechanical properties of the interlayer material that connects these plies have a significant effect on behaviour (Serafinavicius et. al 2013). The viscoelastic nature of this interlayer material indicates that its mechanical properties change with both loading duration and ambient temperature (Xavier et. al 2021). This effect varies depending on the glass transition temperature of the interlayer material. Therefore, EN 16613 classifies different interlayer materials into separate families for different load conditions.

The complex behaviour of the interlayer, coupled with the fact that the strength of the glass also depends on the loading duration, affects parameters such as effective thickness and glass strength (Lopez and Pelayo 2014). For example, under conditions of 40°C ambient temperature and long-term loads such as self-weight, the same glass may have a lower effective thickness, whereas under conditions of 0°C and short-term loads such as wind, the same glass may have a much bigger effective thickness. Each loading scenario is considered separately, and one consequence of this is the variation in load distribution between the inner and outer panes for different effective thickness values. Consequently, the interlayer material is an important input in window glass calculations.

Factors such as glass type, prestressing method, and edge finish type, which affect the strength of the selected glass, are also significant inputs in glass calculations (Vandebroek et. al. 2012). In a glass supported from all four sides, the strength of the glass in the middle, away from the edges, is crucial, whereas, in glass supported from two sides, the edge strength of the glass (which is lower than the strength in the middle) becomes important. All these specific details related to production are critical in window glass calculations.

Loading duration is another factor that changes all equations for several reasons. Firstly, when the duration of the load decreases, we consider a higher strength value for the glass. Secondly, short-term loading increases the interlayer rigidity, and It also leads to higher climatic loading, due to the rigidity of the system.Furthermore, the altered effective thickness changes the load distribution between the inner and outer glass layers for laminated glass. Therefore, it is necessary to carefully apply these interactions in all checks, tracking the nature of each load separately and considering these effects for each load combination to complete window glass controls.

3. Cloud-Based Micro-Service Solution

W-Glass utilizes a composition of modular services to support a distributed operational framework, as depicted in Figure 2. It features a User Interface (UI) that is supported by an intermediary Backend for Frontend (BFF), aimed at simplifying client-server interactions. The system integrates with a third-party authentication service, ensuring secure access. A stateless reporting service is included to accommodate various reporting needs with ease. The core application logic resides within backend services, which are spread out through web services and linked by worker processes.

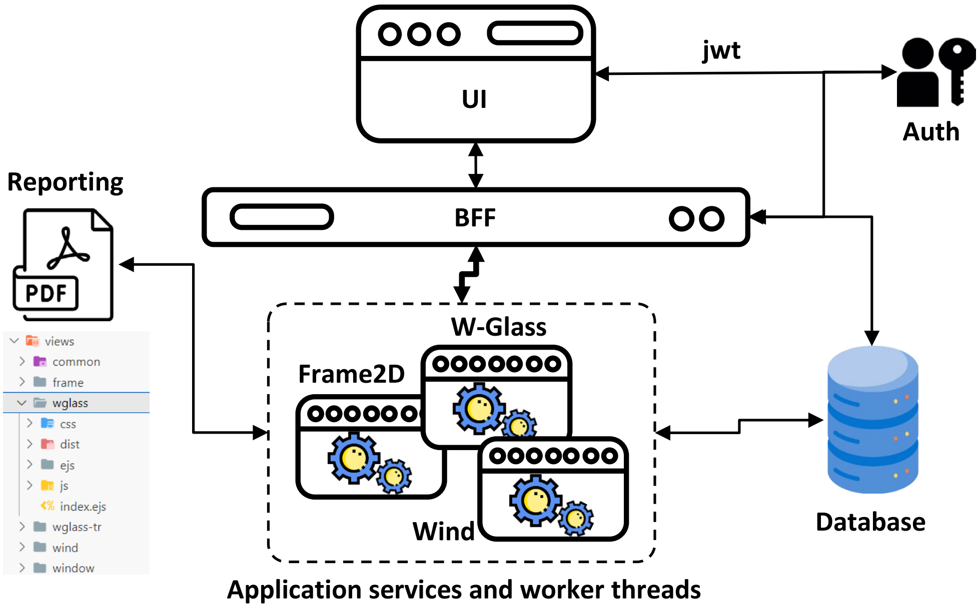

Fig. 2: Schematic representation of the components of the W-Glass cloud architecture (jwt: JSON Web Token).

For the UI, a modern single-page application (SPA) framework is preferred, namely *Vue.js*. This framework embraces model-view architecture while proving a declarative and component-based programming model that helps the developer efficiently code complex user interfaces.

All backend services are programmed with *Express.js*, a web application framework that provides a robust set of features for web and mobile applications. Following is a short discussion of the services that provides a flexible structure to the framework as all are designed with flexibility and extensibility in mind.

The BFF service is kept responsible for all data communication between the front-end and the remaining backend services. It handles all the authentication and validation-related issues, before passing the payload received from the outside world, serving as a gateway for the whole network.

The PDF reporting service is designed as a custom template renderer which converts the custom HTML designs into PDF files using the *wkhtmltopdf* technology. It is designed to automate the mapping of the custom payloads into the HTML rendering engine. The engine then renders the page views that are defined in their respected folders using this payload.

Finally, the application services are designed with a service-worker strategy, in which different web services are employed for each application with their respective worker threads waiting to receive respected payloads, given a job the workers execute the job and return the results back to the server in a stateless fashion, all communicating with their respected services over a well-defined interface.

4. Software Architecture (W-Glass Worker)

The main component of the W-Glass framework, which contains the EN 16612 design calculations, is the W-Glass worker framework. It has been developed using the Python programming language and employs an object-oriented programming (OOP) architecture (Lee and Arora 1991). Figure 3 presents the UML diagram of the classes that constitute a glass unit assembly.

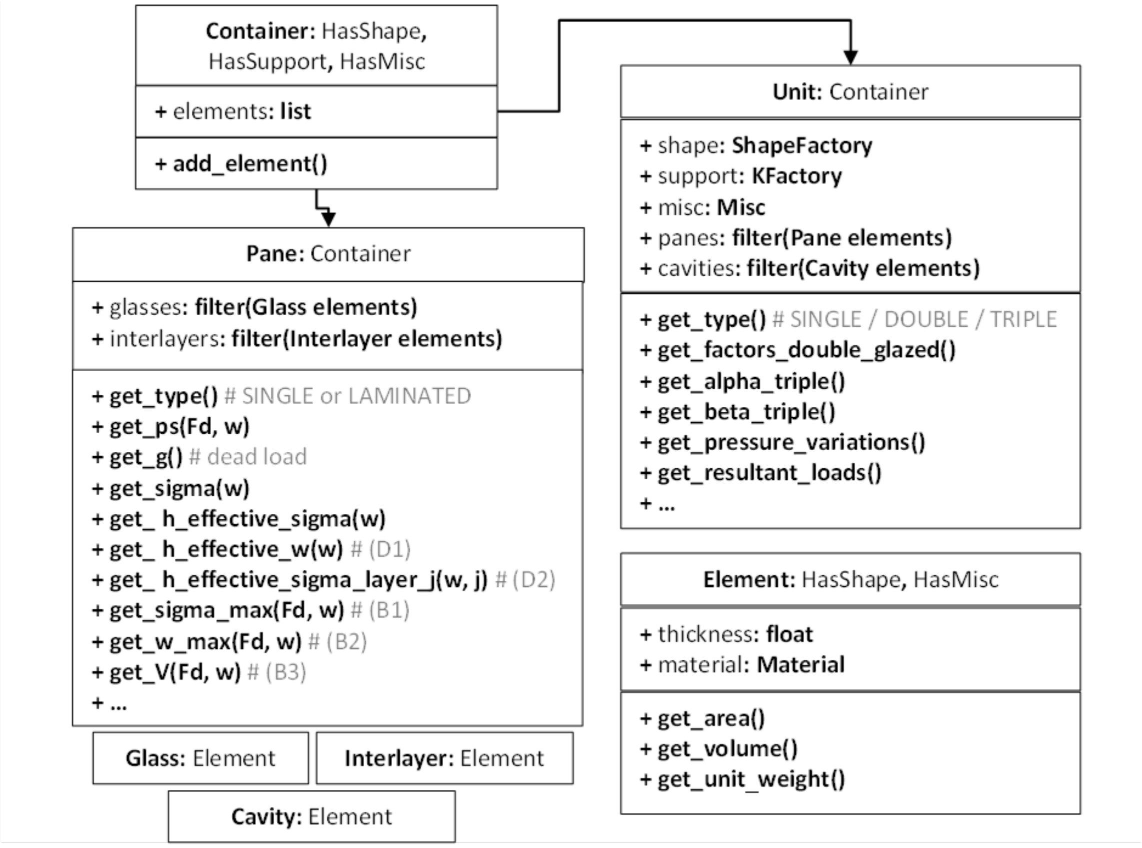

Fig. 3: Design of the classes that constitute a glass unit.

At the heart of this architecture is the Unit class, which acts as the central container for the entire assembly. This class possesses attributes that determine the shape and support mechanisms of the glass unit, along with miscellaneous properties such as slope and other potential attributes that, while not specified in detail, are essential for the unit's comprehensive characterization. The Container base class functions as a generic object to encompass different types of elements. Upon incorporating multiple elements into this generic class, each object is tasked with self-identifying its type by implementing the abstract get type() method. For instance, a Unit classifies its glazing type as single, double, or triple, whereas a Pane specifies whether it is a single glass pane or a laminated one.

There are three types of elements that define the cross-section of a glass unit: Glass, Interlayer, and Cavity. When added to a Pane object, Glasses and Interlayers together form a pane. Subsequently, panes and cavities are incorporated into a Unit object to create a Glass Unit. An Element is characterized by its thickness and material. Once integrated into either a Pane or a Unit, elements can access their physical shape information from their parent objects (either a Pane or a Unit), enabling them to calculate their respective areas, volumes, and unit weights accordingly. Both Unit and Pane classes are equipped with a series of methods to perform the required calculations of EN 16612. These methods include assessing maximum stress and displacements considering geometric non-linearity by utilizing effective thickness parameters.

The glass unit can be characterized by various geometries and support conditions. As illustrated in Figure 4, two factory classes, KFactory and ShapeFactory, are utilized to instantiate the respective objects that specify the support conditions and dimensions of the glass unit.

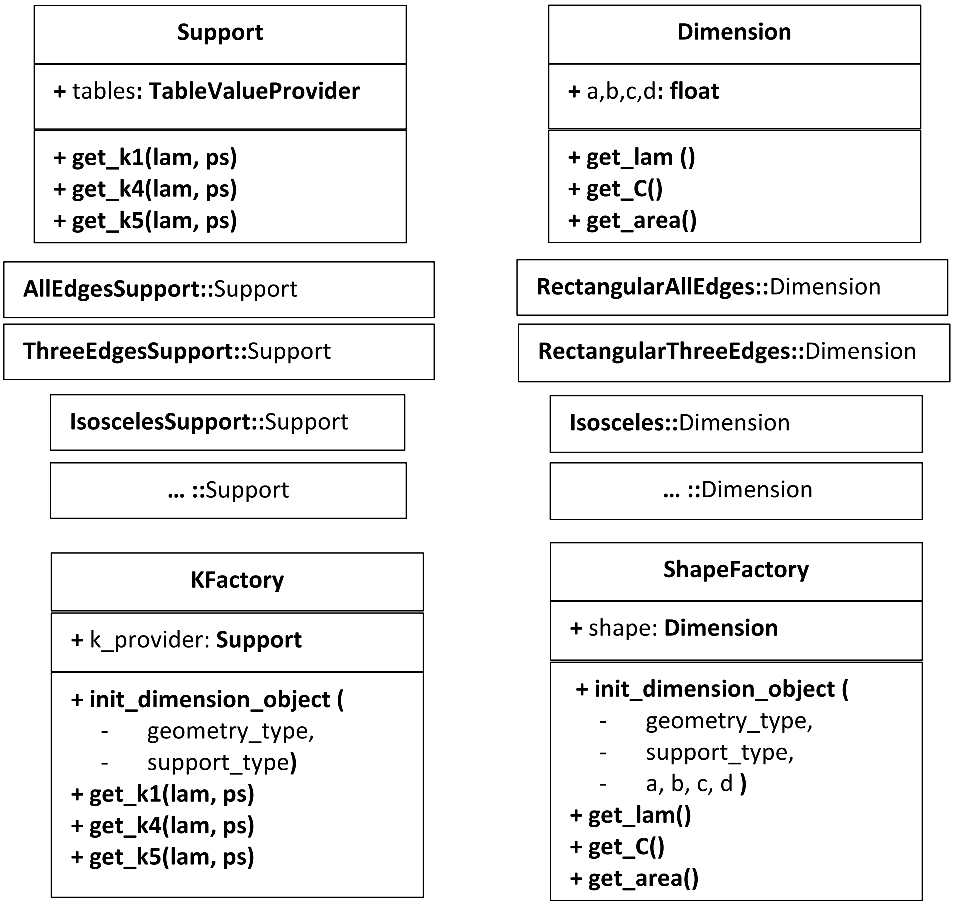

Fig. 4: Design of the classes that defines geometry and support conditions of a glass unit.

Within the W-Glass framework, numerous additional classes perform a variety of functions mandated by the design code. Given the limited space in this paper, the authors find it advantageous to present an example code for the detailed definition and analysis of a triple glass unit in the following section. This self-explanatory code serves as an illustration of the design of classes not covered in this study.

5. Example

The example code presented in Listing 1 illustrates the process of creating a glass unit within the W-Glass framework. It involves specifying the geometry, support conditions, and components of the unit, such as panes and cavities, together with their material properties. Additionally, it demonstrates the calculation of maximum stresses for a triple-glazed glass unit, a solution already provided in EN 16612, Section C.2.

Listing 1: W-Glass implementation of the example triple glazed glass given in EN 16612 Section C.2.

```
sp = ShapeProvider(geometry_type=GeometryType.RECTANGULAR,
                   support_type=SupportType.ALL,
                   a=660, b=2200, L=0, d=0)

kp = KProvider(geometry_type=GeometryType.RECTANGULAR,
               support_type=SupportType.ALL)

gp1, gp2, gp3 = [Pane(), Pane(), Pane()] # Define three empty pane objects

glass_mat = GlassMaterial(id=GlassMaterialType.FLOAT,
                          press=Prestressing.ANNEALED,
                          sprofile=SurfaceProfile.AS_PRODUCED,
                          estrength=EdgeStrength.AS_CUT,
                          safety_class=SafetyClass.CC2)

interlayer_mat = InterlayerMaterial(family=InterlayerFamilies.F0,
                                    wind_location=WindLocationType.MEDITERRANEAN,
                                    snow_location=SnowLocationType.HEATED)

gp1.add_element(Glass(t=6, material=glass_mat))
gp2.add_element(Glass(t=4, material=glass_mat))
gp3.add_element(Glass(t=4, material=glass_mat))
gp3.add_element(Interlayer(t=0.8, material=interlayer_mat))
gp3.add_element(Glass(t=4, material=glass_mat))

cavity_mat = CavityMaterial(id=CavityMaterialType.AIR)
cv1 = Cavity(t=12, material=cavity_mat)
cv2 = Cavity(t=12, material=cavity_mat)

gu = Unit(shape=sp, support=kp, misc= Misc(slope_deg=45)) # Create the Glass Unit

gu.add_element(gp1)
gu.add_element(cv1)
gu.add_element(gp2)
gu.add_element(cv2)
gu.add_element(gp3)

print(gu.get_type()) # UnitType.TRIPLE (Automatically inferred)

lam = gu.pane[0].shape.get_lam()
k5 = gu.pane[0].support.get_k5(lam, 0)

print("lam:", lam)  # lam: 0.3
print("A:", gu.A)   # A: 1.4520000000000002
print("k5:", k5)    # k5: 0.06756645066525173
print("\nPane Volume Changes")
print("vp1:", gu.pane[0].get_vp_k(w=0)) # vp1: 1.2311803430432644e-06
print("vp1:", gu.pane[1].get_vp_k(w=0)) # vp1: 4.155233657771017e-06
print("vp1:", gu.pane[2].get_vp_k(w=0)) # vp1: 2.0776168288855063e-06

print("\nRelative Pane Volume Changes")
alpha_1, alpha_p1, alpha_2, alpha_p2 = gu.alfa_1_1p_2_2p()
print("alpha_1:", alpha_1)   # alpha_1: 7.066002887071076
print("alpha_p1:", alpha_p1) # alpha_p1: 23.847759743864877
print("alpha_2:", alpha_2)   # alpha_2: 23.847759743864877
print("alpha_p2:", alpha_p2) # alpha_p2: 11.923879871932426

print("\nInsulating unit factors")
fi1, fi2 =  gu.fi_1_2()       # fi1: 0.03133444374968933, fi2: 0.027194871113943864
```

```
print("\nFactor BETA")
beta = gu.beta()              # beta: 0.51537683476228

print("\nPane Dead Loads")
print("g_pane1:", gu.pane[0].get_g())   # g_pane1: 0.10606601717798213
print("g_pane2:", gu.pane[1].get_g())   # g_pane2: 0.07071067811865477
print("g_pane3:", gu.pane[2].get_g())   # g_pane3: 0.14142135623730953

print("\nPane External Loads")
p_ex_wind = 0.8                         # (kN/m^2)
p_ex_snow = 0.6 * gu.misc.slope_cos**2  # (kN/m^2)
p0_1 = -4.3                             # (kN/m^2) Isochore pressure
p0_2 = -5.8                             # (kN/m^2) Isochore pressure
w = 0                                   # shear transfer coefficient (0 for the example)

# Definition of the external actions (Dead Load, Wind and Snow loads)
DL1 = Action(id=ActionType.DEAD, factor=365*24).set_duration(50).set_value(gu.pane[0].get_g())
WGP1 = actions[ActionType.WIND_GUST_PRESSURE]
SNOW1 = actions[ActionType.SNOW]

print("\nDL+ 0.6*SL + WL Pressure Variations")
dp11, dp12, dp13, dp14, dp15, dp21, \
dp22, dp23, dp24, dp25 = gu.pressure_variations(p0_1, p0_2, 0.6*p_ex_snow + p_ex_wind, 0, w=w)
print("d11:", dp11)    # d11: -0.261436096920834
print("d12:", dp12)    # d12: -0.22869658613505878
print("d13:", dp13)    # d13:  0.46658108749916416
print("d14:", dp14)    # d14: -0.03603376176536905
print("d15:", dp15)    # d15: -0.06649121639658934
print("d21:", dp21)    # d21: -0.16955091730702634
print("d22:", dp22)    # d22: -0.3060483937614857
print("d23:", dp23)    # d23:  0.3025949833069471
print("d24:", dp24)    # d24:  0.022489254721619393
print("d25:", dp25)    # d25: -0.08898047111820871

print("\nResultant Loads for DL+ 0.6*SL + WL")
[[cp_1, Fd_1],
 [cp_2, Fd_2],
 [cp_3, Fd_3]] = gu.resultant_loads(p0_1, p0_2, Fex1=0.6*p_ex_snow + p_ex_wind, Fex3=0, w=w)

print("Cavity Pressure 1:", cp_1, "Fd_1:", Fd_1) #  0.4901326830558928,  0.7220099078407765
print("Cavity Pressure 2:", cp_2, "Fd_2:", Fd_2) # -0.01453337198738075, 0.19866302054550278
print("Cavity Pressure 3:", cp_3, "Fd_3:", Fd_3) # -0.47559931106851205, 0.3775251231476673

pane_1_max_stress = gu.pane[0].get_sigma_max(Fd=Fd_1, w=w) / 1000  # 6.321606699947406
pane_2_max_stress = gu.pane[0].get_sigma_max(Fd=Fd_2, w=w) / 1000  # 1.7394075464147996
pane_3_max_stress = gu.pane[0].get_sigma_max(Fd=Fd_3, w=w) / 1000  # 3.3054468128849446
```

The provided code outlines the process of defining and analysing a triple-glazed glass unit using the W-Glass framework. It starts by setting up dimensional parameters and support types for the glass unit, followed by the initialization of pane objects and the assignment of materials to these panes, including glass and interlayer materials. Cavities between the panes are also defined, emphasizing the comprehensive setup of the glass unit's structure. The unit is then assembled by adding panes and cavities, showcasing the flexibility of the framework in handling complex glass unit configurations. The code demonstrates calculations for various physical properties, such as volume changes and pressure variations, underscoring the framework's capability to conduct detailed analyses based on predefined conditions and materials. This example succinctly illustrates the framework's utility in facilitating the design and evaluation of glass units in compliance with the EN 16612 standard, highlighting its potential for simplifying complex engineering tasks.

6. Conclusions

Macrostatic W-Glass has been developed to facilitate the application of the EN 16612 standard more easily. By offering various services and a carefully designed object-oriented worker architecture for analysing and designing glass units, the platform aims to overcome the complexities of these standards with flexibility and extensibility in mind. This effort represents a step towards making the design of glass units more accessible, with the potential for further simplification and enhancements in future research.

Acknowledgements

The authors would like to express our gratitude to The Scientific and Technological Research Council of Türkiye (TÜBİTAK) for their support of the W-Glass development through project code 2130086.

References

Badalassi, m., Biolzi, L., Royer-Carfagni, G., Salvatore, W.: Safety factors for the structural design of glass. Construction and Building Materials. Volume 55, Pages 114-127 (2014)

Centelles X., Pelayo F., Lamela-Rey M.J., Fernández A.I., Salgado-Pizarro R., Castro J.R., Cabeza L.F.: Viscoelastic Characterization of Seven Laminated Glass Interlayer Materials from Static Tests. Constr. Build. Mater. (2021)

DIN 18008: Glass in building – Design and construction rules – Part 1: Terms and general bases, draft May 2018 Part 2: Linearly supported glazing, draft May (2018)

EN 16612:2019: Glass in building. Laminated glass and laminated safety glass. Determination of interlayer viscoelastic properties (2019)

EN 16613:2019: Glass in building. In: Determination of the Lateral Load Resistance of Glass Panes by Calculation (2019)

Express.js: Fast, unopinionated, minimalist web framework for Node.js. https://expressjs.com. Accessed (2024)

Feldmann, M., Kasper, R. et al.: Guidance for European Structural Design of Glass Components. EUR 26439. Luxembourg (Luxembourg): Publications Office of the European Union; (2014)

Haldimann, M., Luible, A., and Overend, M.: Structural Use of Glass, Zürich,Switzerland: IABSE-AIPC-IVBH, (2008)

Kozlowski, M., et. al: Experimental and Numerical Simulations of Climatic Loads in Insulating Glass Units by Controlled Change of Pressure in the Gap. *Appl. Sci.*, *13*, 1269 (2023)

Lee H, Arora JS: Object-oriented programming for engineering applications. Eng Comput. 7(4), 225–35 (1991)

López-Aenlle, M., & Pelayo, F.: Dynamic effective thickness in laminated-glass beams and plates, Compos. B 67, 332–347, (2014)

Lori, G., and Carfagni, G., R.: Betti's analytical method for the load sharing in double glazed units, Composite Structures, vol. 235 111765, (2020)

Lori, G., et. al. A simple numerical tool for superimposition of climatic load and wind load in Closed Cavity Facades: importance of a coupled PE-thermal-mechanical model. Glass Structures and Engineering 7(2):1-20 (2022)

Macrostatic W-Glass: https://www.macrostatic.com/

McMahon, S., Norville, S. H., and Morse, S. M., Experimental investigation of load sharing in insulating glass units, Journal of Architectural Engineering, vol. 24 04017038, (2018)

Halilovič, M., Maček, A., Mole, N., Koc, P., Plešnik, F., Rus, P., Žnidaršič, M., Kralj, A.: Accurate determination of the static equilibrium in insulating glass units under climatic loading, Journal of Building Engineering, 10.1016/j.jobe.2023.107955, 80, (107955), (2023).

Morse, S. M., & Norville, H. S.: Comparison of methods to determine load sharing of insulating glass units for environmental loads. Glass Structures & Engineering, 1(1), 315–329, (2016)

prEN 13474-3:2009: Glass in building - Determination of the strength of glass panes - Part 3: General method of calculation and determination of strength of glass by testing, CEN, (2009) prEN 16612:2013: Glass in Building - Determination of the load resistance of glass panes by calculation and testing, CEN, (2013)Serafinavicius, T. et. al.: Long-term laminated glass four point bending test with PVB, EVA and SG interlayers at different temperatures, Procedia Eng. 57, 996–1004 (2013)

Siebert, G.: Comparison of IGU-design according to German and European Standards. In: Ce/Papers 2 pp. 81–92 (2018),

Vandebroek, M., et. Al.: Experimental validation of edge strength model for glass with polished and cut edge finishing, Engineering Fracture Mechanics 96, pp. 480-489 (2012)

Vue.js: The Progressive JavaScript Framework. https://vuejs.org Accessed 15 June 2024.

wkhtmltopdf: Open-source command line tools to render HTML into PDF. https://wkhtmltopdf.org. Accessed 15 June 2024.

Challenging Glass Conference Proceedings – Volume 9 – June 2024 – Louter, Bos & Belis (Eds.)
International Conference on the Architectural and Structural Application of Glass
Challenging Glass Conference 9 – 19 & 20 June 2024 – TU Delft – The Netherlands

CGC9

Platinum Sponsor

SAINT-GOBAIN

Gold Sponsors

EASTMAN

kuraray

sedak

Silver Sponsors

DOW®

octatube

PERMASTEELISA GROUP

Organising Partners

TU Delft Delft University of Technology

Technical University of Munich TUM

GHENT UNIVERSITY