# EVALUATION OF GLASS DOMES USING PYFORMEX

**Jan BELIS**
Assistant Professor
Ghent University
Ghent, BELGIUM

**Benedict VERHEGGHE**
Professor
Ghent University
Ghent, BELGIUM

**Matthieu DE BEULE**
Assistant
Ghent University
Ghent, BELGIUM

**Rudy VAN IMPE**
Professor
Ghent University
Ghent, BELGIUM

**Summary**

In contemporary architecture, the use of glass in buildings is not restricted to cladding and infill applications anymore. A strong tendency exists to attribute to glass a primary structural role, as is the case in glass domes. For this type of structures, several conceptual, experimental or built examples exist.

Obviously, the geometry of these different glass domes (diameter, shallowness, meshing pattern, glass panel curvature,…) is of major importance for their mechanical behaviour (transfer of forces, overall and local stability,…). For this reason, it was of interest to dispose of a practical tool, suitable for parametric analyses, for an "a posteriori" evaluation of the geometry of one of the domes mentioned above.

A powerful and convenient tool to perform the required geometrical manipulation tasks is currently under development at Ghent University by Verhegghe. The free software tool is called "pyFormex", a Python implementation of Formex algebra. pyFormex was developed for the automated design of spatial structures by means of sequences of mathematical transformations: using a surprisingly short script, pyFormex is able to generate a visual three-dimensional model or a spatial definition of nodes and elements that can be used as input for finite elements (FE) models.

The objectives of this contribution are two-fold. First, the power of pyFormex scripting and its suitability for the geometrical designing of structures like glass domes are demonstrated. Secondly, some of the pyFormex possibilities regarding to further structural analysis are retrospectively illustrated on an example of a recently realised glass dome.

**Keywords**: glass, glass domes, pyFormex, Formex, python, facetted shells, hotel Hesperia, Abaqus.

## 1. Introduction

Since more than a decade, the application of glass in buildings has been gradually expanding from cladding and infill applications towards primary load-bearing building components. Examples of such relatively transparent components are i) glass beams that support a glass roof or floor, ii) glass fins that act as stiffeners against wind loads on a façade, iii) glass columns, iv) folded glass plate structures [15] and v) glass domes.

In this last category, some examples exist in recent glass-building history, e.g. the self-supporting glass-and-cable dome presented by Glasbau Seele at the 1998 glasstec fair in Düsseldorf [17]; the built prototype for a garden pavilion with flat, replaceable trapezoidal monolithic glass panels at Delft University of Technology [5], [16], [17]; the conceptual study for a ring grid dome with incorporated sun shading textiles in Aachen [17], the structurally bonded spherical dome with double-curved laminated glass panels, created by Blandini and Sobek at Stuttgart University (ILEK) [1], [4]; and recently the glazed dome of the EVO restaurant on top of the Hesperia Tower hotel in Barcelona [14]. A brief overview of selected data about these projects is given in Table 1 and Fig. 1:

*Table 1 Examples of glass dome projects (FLT=annealed float glass, HS=heat-strengthened glass, T=toughened glass, CS=chemically strengthened glass, IGU=insulating glass unit, *=not built)*

| No. | Year | Design & concept | Location | Glass (mm) | Geometry | Concept |
|---|---|---|---|---|---|---|
| 1 | 1998 | Stuttgart University, Seele GMBH & CO.KG | Düsseldorf (D) | 2x10 HS | Equilateral triangles projected on sphere: 282 flat triangular facets; 27 different formats | Flat spherical dome on circumferential steel ring; prestressed by cables |
| 2 | 2002 | Delft University of Technology | Delft (NL) | 8 T | 4 Rings, 16 medians: 64 flat trapezoids | Ring grid dome with open apex; panels connected by bonded and clamped stainless steel profiles |
| 3* | 2003 | RWTH Aachen | Aachen (D) | 2x8 FLT | 5 Rings, 20 meridians: 100 flat trapezoids | Ring grid dome with open apex; meridian joints by silicone gaskets; ring joints by bolted clamping plates at edges; shading fabric provided |
| 4 | 2003 | ILEK, Stuttgart University | Stuttgart (D) | 8 FLT + 2 CS | 44 Double-curved laminated glass panels | Spherical shell on circumferential titanium ring; panels solely connected by adhesive joints |
| 5 | 2006 | Richard Rogers Partnership; Alonso, Balaguar y Arq. Asociados; Bellapart S.A.U. | Barcelona (E) | 8 T + 2x4 FLT | Icosahedron subdivided in triangles and projected on sphere: 219 triangular IGUs | Supported by steel wireframe; glass cladding fixed with structural silicone sealant and additional clamps along vertices; 5m wide and 3m high entrance opening provided |

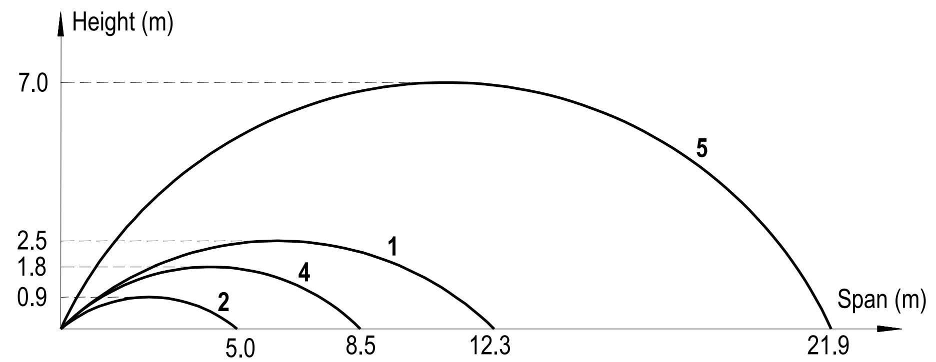


*Fig. 1 Comparison of dimensions of built glass domes (the numbers relate to Table 1)*

Table 1 shows that most of these examples are composed of a considerable number of glass elements, which often show an important variety of shapes and sizes. Therefore, finding the exact mesh in terms of coordinates of nodes, lines and planes determining the dome's geometry may become a very complex task. Moreover, even for relatively simple geometries this can quickly become a time-consuming activity, particularly when parametric studies are required – which will often be the case during a design process.

Consequently, a convenient tool to perform the required geometrical manipulation tasks could be very helpful. For this reason, a powerful software tool is presented, which is very well suited for the tasks described above. The free tool is called "pyFormex", a Python implementation of Formex algebra. Using a surprisingly short script, pyFormex is able to generate a visual three-dimensional model or a spatial definition of nodes and elements that can be used as input for FE models.

In this contribution, the general philosophy of pyFormex is explained and applied to two glass dome case studies.

## 2. pyFormex

pyFormex was developed by Verhegghe at Ghent University for the automated design of spatial structures and the generation of complex three-dimensional geometries by means of sequences of mathematical transformations [9]. Furthermore, pyFormex can be useful for structural analysis purposes, for operations on surface models or simply for generating illustrations.

pyFormex is a Python implementation of Formex algebra, pioneered by Nooshin [12]. It implements most features of the Formian language [9], but has far wider aspirations. Being an open source project and programmed in Python, pyFormex is fully open to quickly create extensions, including analysis modules, interfaces with other programs, customizations of the GUI. Python also allows for a much neater scripting language, where logically named operators perform different action depending on the operands. Because of all this, the field of applications of pyFormex has quickly broadened way beyond that of spatial structures.

Originally developed in a Linux environment, pyFormex can be used, however, in a Windows environment as well. The software is based on a script, from which a whole construction can be generated using only a limited number of command lines [8]. Typical Formex operations used in scripts are related to copying, translating, rotating, scaling, etcetera. The major advantage of this modus operandi is that parameters (e.g. dimensions, material properties, ...) can easily be replaced and consequently a multitude of different configurations of the same geometry can be generated in a limited period of time.

Also because of the scripting, there is virtually no limitation to the geometries that pyFormex can generate other than one's own imagination (see online examples, [7]).
The development of pyFormex is an ongoing process, presented as an open source project: the program can be used, studied, modified and distributed under the conditions of the GNU Genral Public License (GPL) [6]. For the time being, the program is already in use in both a research and educational context. Future developments of this tool include interactive tools, surface and volume meshing, post-processing, distribution and installation.

## 3. Case studies

### 3.1 Hotel Hesperia's glazed dome

#### *3.1.1 Generation of the geometry using pyFormex*

Apart from its unique location at 95 m atop a hotel tower, the relatively large size (see above, Fig. 1) and high geometric complexity (see above, Table 1) of the Hotel Hesperia's glazed dome make it a very interesting example of a recent construction on which some of the possibilities of pyFormex can be illustrated a posteriori (Fig. 4, Fig. 5, Fig. 6). The interested reader can find a detailed description of the project in literature [14]. The major pyFormex input commands for the construction of the dome geometry are given and commented in Table 2. Fig. 2 visualises the according step-by-step generation of the structure.

*Table 2 Selection of commented pyFormex input commands for the generation of the Hotel Hesperia glazed dome*

| Step | Comments | pyFormex input |
|---|---|---|
| 1 | Construct a triangle of an icosahedron oriented with a vertex in the y-direction; divide edges into *n* parts | n = 6 |
| 2 | Start with an equilateral triangle in the x-y-plane | F = simple.triangle() |
| 3 | Create a mirror for extension | G = F.select([0,2]).reflect(0,F[0,1,0]) |
| 4 | Determine its edge length | a = length(F[2,0] - F[0,0]) |
| 5 | Replicate *n* times | b = a*sqrt(3.)/2.<br>F = F.replic2(n,n,a,b,1,0,bias=a/2,taper=-1) |
| 6 | Create the extension | G = G.replic(n,a,1) |
| 7 | Remove the outermost members | keep = range(G.nelems())<br>del keep[1]<br>del keep[-2]<br>G = G.select(keep) |
| 8 | Get the last vertex and make it the origin | P = F[-1,-1]<br>F.setProp(3)<br>G.setProp(2)<br>F += G<br>F = F.translate(-P) |
| 9 | Rotate around the y axis over an angle so that the projection on the x-y plane is an isosceles triangle with top angle = 360/5 = 72 degrees. The base angles thus are (180-72)/2 = 54 degrees.<br>Ratio of the height of the isosceles triangle over the icosaeder edge length. | c = 0.5*tand(54.)<br>angle = arccos(tand(54.)/sqrt(3.))<br>F = F.rotate(-angle/rad,1) |
| 10 | Project it on the circumscribing sphere with radius *ru* | golden_ratio = 0.5 * (1. + sqrt(5.))<br>ru = 0.5 * a * sqrt(golden_ratio * sqrt(5.))<br>ru *= n<br>C = [0.,0.,-ru]<br>F = F.projectOnSphere(ru,center=C)<br>x,y,z = F[1][1]<br>h = -z<br>rb = sqrt(x*x+y*y) |
| 11 | Give the base points a z-coordinate 0 | F = F.translate([0.,0.,h]) |
| 12 | Create the base circle | H = simple.circle().scale(rb) |
| 13 | Take out the bottom elements and determine intersection with base plane | G = F.withProp(1)<br>G = cutAtPlane(G,[0.,0.,0.],[0.,0.,1.]) |
| | Add back | F = F.withProp(3) + G |
| 14 | Finally, create a rosette to make the circle complete | F = F.rosette(5,72.) |
| 15 | Cut out the access: remove all members having a point less than edge-length "a" away from the base point | d = distanceFromPoint(F.f,[-rb,0.,0.])<br>t = (d < a).any(axis=-1)<br>F.p[t] = 0<br>F = F.select(F.p != 0) |
| 16 | Get supported bottom nodes | B = Formex(F.f[isClose(F.f[:,:,2],<br>0.0)]).unique(rtol=1.e-4,atol=1.e-5) |

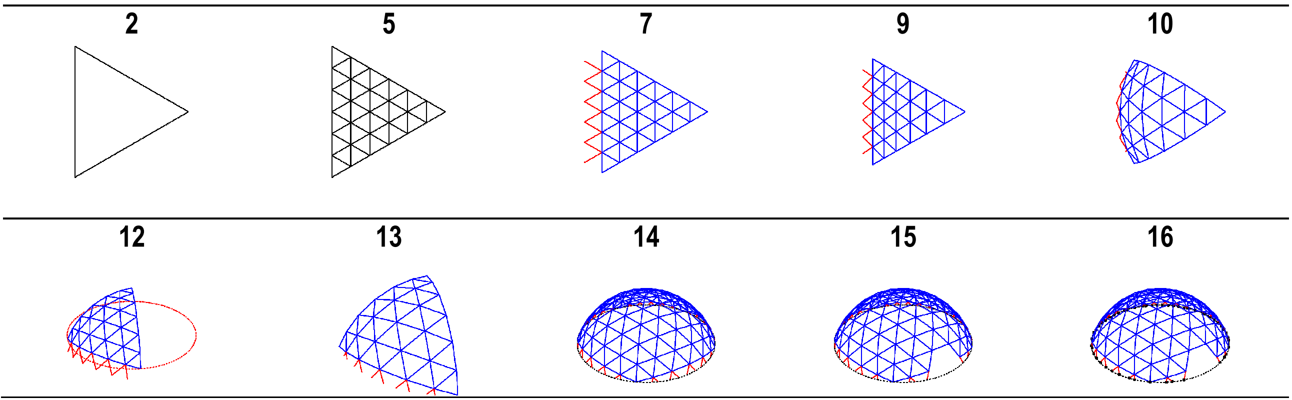


*Fig. 2 Resulting geometry according to selected steps from Table 2 for the generation of the Hotel Hesperia glazed dome using pyFormex*

### *3.1.2 Structural evaluation*

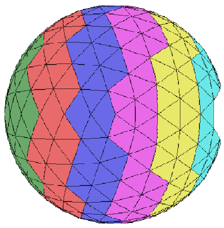

*Fig. 3 Overview of load distribution zones due to non-uniform snow load on the Hesperia glazed dome*

Given the elevated location of the glazed dome (see Fig. 4, Fig. 5 and Fig. 6), wind loads obviously played a major role in the overall design of the construction. For each glass panel, a specific wind load was obtained from a three-dimensional fluid flow analysis of the complete building, taking into account 12 wind directions [14]. In addition, the dome was originally designed according to Eurocode 1 snow loads as well: a qualitative overview of the corresponding non-uniform load distribution is shown in Fig. 3.

Recently, an Abaqus® input file generation process was incorporated directly in the pyFormex graphical user interface (GUI) to facilitate further finite elements analysis of pyFormex models. Using these pyFormex pre-processing facilities, the model was retrospectively loaded and subsequently analysed with the finite elements package Abaqus® [1].

*Fig. 4 Hotel Hesperia in Barcelona, upwards view (Photo: Bellapart S.A.U.)*

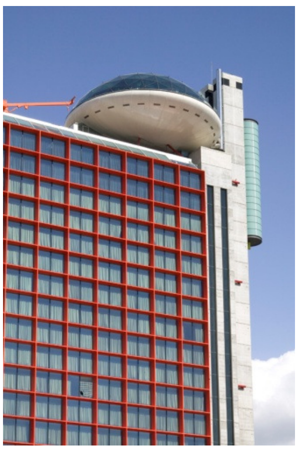

*Fig. 5 EVO restaurant on top of the hotel (Photo: Bellapart S.A.U.)*

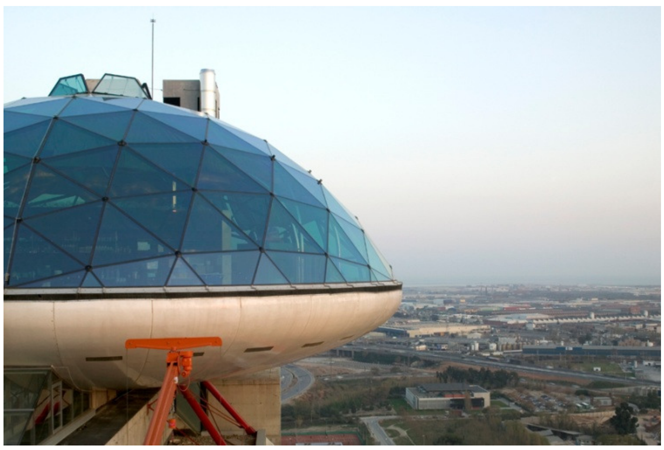

*Fig. 6 Glazed dome of the restaurant on top of Hotel Hesperia (Photo: Bellapart S.A.U.)*

The coupling of pyFormex and Abaqus® is illustrated below for the total nodal displacements and the normal forces in the

< >

steel members due to the own weight of the structure (steel and glass cladding; Fig. 7 and Fig. 8) and due to a non-uniform snow load (Fig. 9 and Fig. 10). The figures are generated with Abaqus®; colour codes vary from blue (low values) to red (high values), with maximum total nodal displacements of 1.77 mm and 1.36 mm for both separate loading cases respectively. Since the FE models have only a demonstrative purpose, simplified boundary conditions have been used (all nodes of the base ring have been fixed). Obviously, the resulting values mentioned above are influenced by this choice.

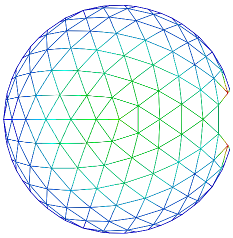

*Fig. 7 Overall displacements of the supporting steel structure of the hotel Hesperia glazed dome solely due to its own weight (steel + glass)*

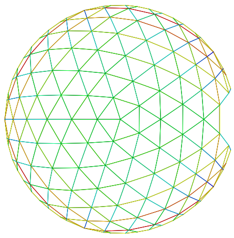

*Fig. 8 Normal forces in the supporting steel structure of the hotel Hesperia glazed dome solely due to its own weight (steel + glass)*

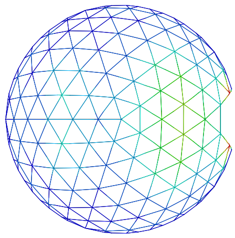

*Fig. 9 Overall displacements of the supporting steel structure of the hotel Hesperia glazed dome solely due to non-uniform snow load*

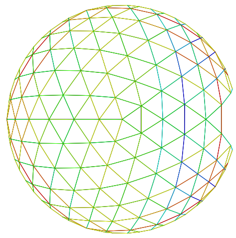

*Fig. 10 Normal forces in the supporting steel structure of the hotel Hesperia glazed dome solely due to non-uniform snow load*

### 3.2 Facetted shell structure of glass

Bagger et al. already pointed out that facetted shell structures are feasible to create highly transparent glass constructions without additional supporting structure: loads should basically be taken by in-plane stresses in the glass facets and transferred from one facet to another by distributed forces along the edges [1]. Even though many practical issues currently could still be questioned (e.g. high costs for manufacturing polygonal glass panes with high geometrical precision, development of suitable edge connection methods, erection methods,…), the idea seems very valuable from a theoretical point of view. Obviously, in such a concept all facets of a glass construction should by perfectly planar.

One might be tempted to use the same simple approach as for the Hesperia example: subdividing the facets of the icosahedron in a hexagonal base pattern (Fig. 11) and then projecting it on a sphere. While this leads to a nice hexagonal dome (Fig. 12), the hexagons in it are not fully planar.

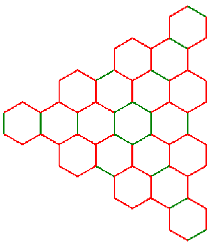

*Fig. 11 Hexagonal base pattern based on one triangular plane of an icosahedron*

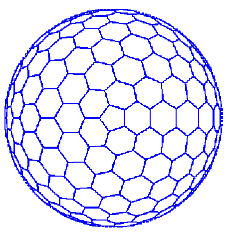

*Fig. 12 Hexagonal dome based on a spherical projection of a hexagonal base pattern of an icosahedron*

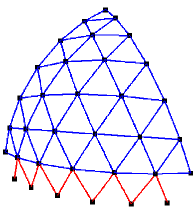

*Fig. 13 Triangular pattern based on a spherical projection of a triangular base pattern of an icosahedron*

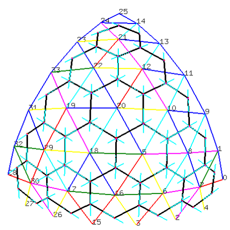

*Fig. 14 A section (1/5) of a facetted spherical dome with hexagonal planar facets generated with pyFormex*

Therefore a more complex procedure was used. Starting from the nodes of the Hesperia dome (Fig. 13), we constructed the tangent planes and then determined the intersection lines of those planes with pyFormex (Fig. 14).

## 4. Discussion

Once a pyFormex script has been created, further possibilities for visualisation, parametric variation and study, preprocessing,… etcetera are virtually unlimited. As an example, some variations of the Hesperia glazed dome are depicted below, based on a variation of the parameter *n* in the pyFormex script (see step 1 in Table 2). As stated before, however, possibilities of pyFormex are not at all limited to glass domes: several other examples are available online [7].

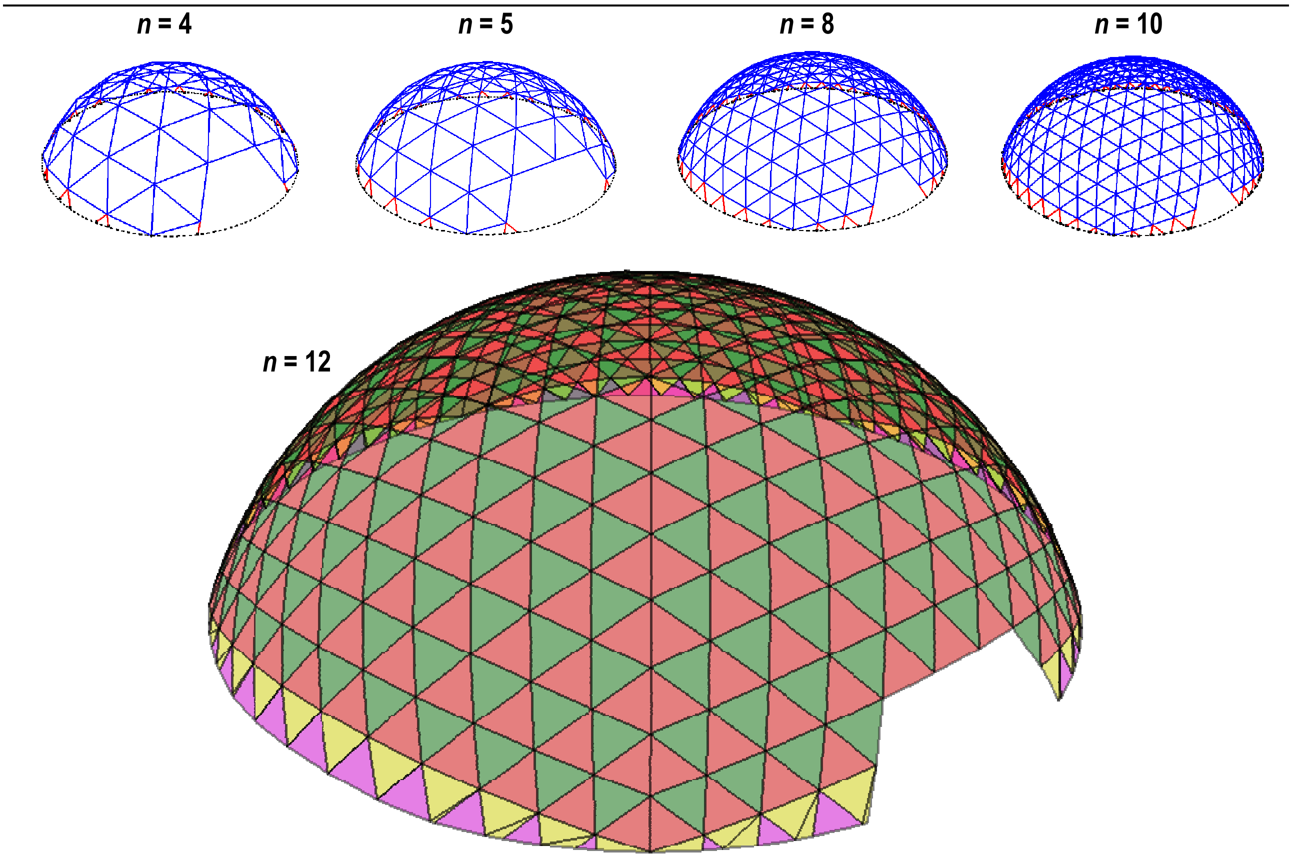


*Fig. 15 Geometric variations of the Hesperia glazed dome generated by varying the parameter n in the pyFormex script*

## 5. Conclusions

A review of existing glass dome designs has revealed that these structures often have to deal with a large variety and a large number of glass elements, which require a high-precision designing method. To ease the design process, the in-house developed software tool "pyFormex" was presented and applied to different glass dome examples. The main conclusions are:

- Using a very condensed scripting method, pyFormex is able to generate relatively complex geometries using only very few input command lines. The example of the hotel Hesperia glazed dome was generated in only 16 steps, including entrance opening and supporting base ring.
- The scripting method makes pyFormex very well suited for preparing parametric structural analyses: geometric variations can be studied in an automated way changing only one or more simple parameters (e.g. the span or height of a dome, the amount of secondary triangles, etc.).
- Automated generation of Abaqus® FE input files can be incorporated in pyFormex, as was shown for the Hesperia glazed dome.
- pyFormex is open source software and is continuously subjected to further development. Users are warmly encouraged to study, use and further develop its possibilities under the conditions of the GNU GPL [6].

### 5.1 Acknowledgements

The authors thankfully acknowledge Mr. Carles Teixidor (Bellapart S.A.U.) who was very helpful in providing useful supplementary data on the Hotel Hesperia's glazed dome (www.bellapart.com).

### 5.2 References

[1] ABAQUS, INC., *User Manuals Abaqus Version 6.6*, 2006.

[2] BAGGER, A., JÖNSSON, J., ALMEGAARD, H. and WESTER, T., "Facetted Shell Structures of Glass", *Proceedings of Glass Processing Days*, 15-18 June 2007, Tampere, pp. 111-114.

[3] BLANDINI, L., "The Glass Dome", *Proceedings of Glass Processing Days*, 17-20 June 2005, Tampere, pp. 416-419.

[4] BLANDINI, L., *Structural Use of Adhesives in Glass Shells*, dissertation, ILEK, Universität Stuttgart, Verlag Grauer, Beuren and Stuttgart, 2005, 88 pp.

[5] HOBBELMAN G.J, TIMM B., VEER F.A. and VAN SWIETEN P.M.J., "The Design of a Full Glass Dome, Using a New Glass/polymer Composite Material", *Lightweight Structures in Civil Engineering*, ed. Jan B. Obrebski, Proceedings of the International IASS Symposium on Lightweight Structures in Civil Engineering, 24-28 June, Warsaw, 2002, pp. 828-831.

[6] http://www.gnu.org/licenses/gpl.html.

[7] http://pyformex.berlios.de/examples.html.

[8] http://pyformex.berlios.de/manual/.

[9] http://pyformex.berlios.de.

[10] http://www.python.org.

[11] NEELS T. and VERHEGGHE B, *pyFormex manual*, Release 0.4.2, Ghent, 2007, 59 pp.

[12] NOOSHIN H., "Algebraic Representation and Processing of Structural Configurations", *International Journal of Computers and Structures*, Vol. 5, Issues 2-3, June 1975, pp. 119-130.

[13] NOOSHIN H. and DISNEY P., *Formian 2,* Multi-Science Publishing Co. Ltd., 1997

[14] TEIXIDOR C., "Hotel Hesperia's Glazed Dome, Barcelona, Spain", *Structural Engineering International*, Vol. 17, No 1, February 2007, pp. 53-55.

[15] TROMETER S. and KRUPNA M., "Development and Design of Glass Folded Plate Structures", *Journal of the International Association for Shell and Spatial Structures*, Vol. 47 No. 3, December 2006, pp. 253-260.

[16] VEER, F.A, WURM J. and HOBBELMAN G.J., "The Design, Construction and Validation of a Structural Glass Dome", *Proceedings of Glass Processing Days*, 15-18 June, Tampere, 2003, pp. 156-158.

[17] WURM J., *Glass Structures, Design and Construction of Self-supporting Skins*, Birkhäuser, Basel-Boston-Berlin, 2007, 255.