

## Stabil: An educational Matlab toolbox for static and dynamic structural analysis

Peer-reviewed author version

Francois, Stijn; Schevenels, Mattias; DOOMS, David; Jansen, Miche; Wambacq, Jef; Lombaert, Geert; Degrande, Geert & De Roeck, Guido (2021) Stabil: An educational Matlab toolbox for static and dynamic structural analysis. In: COMPUTER APPLICATIONS IN ENGINEERING EDUCATION, 29(5), p.1372-1389.

DOI: 10.1002/cae.22391

Handle: <http://hdl.handle.net/1942/33871>

# Stabil: an educational Matlab toolbox for static and dynamic structural analysis <sup>\*,\*\*</sup>

Stijn François<sup>1</sup> | Mattias Schevenels<sup>2</sup> | David  
Dooms<sup>1‡§</sup> | Miche Jansen<sup>1¶</sup> | Jef Wambacq<sup>1</sup> |  
Geert Lombaert<sup>1</sup> | Geert Degrande<sup>1</sup> | Guido De  
Roeck<sup>1</sup>

<sup>1</sup>Structural Mechanics Section, Department of Civil Engineering, KU Leuven, Kasteelpark Arenberg 40, B-3001 Leuven, Belgium

<sup>2</sup>Architectural Engineering Section, Department of Architecture, KU Leuven, Kasteelpark Arenberg 1, B-3001 Leuven, Belgium

## Correspondence

Corresponding Author: Stijn François, Department of Civil Engineering, KU Leuven, Kasteelpark Arenberg 40, B-3001 Leuven, Belgium.  
Email: [stijn.francois@kuleuven.be](mailto:stijn.francois@kuleuven.be)

## Present address

<sup>‡</sup>Mouton, Voskenslaan 171, B-9000 Ghent.

<sup>§</sup>Faculty of Architecture and Arts, Hasselt University, Agoralaan gebouw E, B-3590 Diepenbeek.

<sup>¶</sup>Cenaero, Rue des Frères Wright 29, B-6041 Charleroi

## Funding information

KU Leuven, Educational Policy Unit, OOI Project 2006/20 "An interactive and adaptive application for the static and dynamic analysis of structures".

## Postprint version.

<sup>\*\*</sup>**Published version:** S. François, M. Schevenels, D. Dooms, M. Jansen, and G. Lombaert, G. Degrande, and G. De Roeck. Stabil: an educational Matlab toolbox for static and dynamic structural analysis. *Computer Applications in Engineering Education*, 2021:1-18. DOI: <https://doi.org/10.1002/cae.22391>

The capability to analyze structures under static and dynamic loads is an essential skill for structural engineers. Structural analysis therefore is a key component in civil and architectural engineering education, where analytical methods are traditionally complemented by the use of (commercial) software packages. The latter are often closed source, which may obscure the links to the underlying matrix structural analysis or finite element formulation. In order to stimulate active, cooperative, and solution-oriented learning, we developed Stabil, an electronic learning environment implemented as a Matlab toolbox. Stabil has an open structure to elucidate the links between theory and implementation and is presently used throughout the curricula of Civil and Architectural Engineering at KU Leuven. This paper explains the main principles of Stabil and discusses teaching experiences over the past 15 years. Stabil can be downloaded from <http://bwk.kuleuven.be/bwm/stabil>.

## KEYWORDS

Structural Analysis, Matrix Analysis of Structures, Finite Elements, Technology Enhanced Active Learning, Blended Learning

## 1 | INTRODUCTION

A fundamental understanding of structural behavior is essential for civil and architectural engineers and forms the basis of well-considered designs. Apart from a solid theoretical background in strength of materials and static and dynamic structural analysis, this also requires hands-on experience in using modern software packages. The user-friendliness of such tools, however, should not make students believe that structural analysis can be performed without expert knowledge. Structural analysis and interpretation of results requires a thorough knowledge of the underlying theories as well as insight in structural behavior [1]. Teaching digital literacy to civil and architectural engineers, linked to structural insight and critical thinking therefore is a key aspect of modern engineering education.

A traditional approach to teach structural analysis is centered around analytical methods and hand calculations [2, 3]. This approach is commonly supported by hands-on activities [4, 5], such as graphic statics [6], classroom experiments [7], and project-based cooperative learning methods [8, 9, 10]. In addition, several computer tools have been developed to provide structural insight to bachelor students with little knowledge of (computational) structural analysis [11, 12, 13]. EasyStatics [14], for example, is a teaching and learning tool for bachelor students in architecture and civil engineering. A similar approach is used in Schnittkraftmeister [15] to convey an intuitive reasoning on member forces in isostatic structures. While such tools do not guarantee an improved learning outcome [16], they may at least trigger motivation and interest with students [12] and be useful in promoting Science, Technology, Engineering, and Mathematics (STEM) education [17].

Computer resources have become readily available and computational techniques nowadays allow to perform complex numerical simulations in real time. This offers the possibility to bring digital intuitive learning to a next level. In the StaticAR project, for example, augmented reality is employed to visualize member forces in wooden scale models [18]. Model order reduction techniques are used to speed up complex calculations and to provide results in real-time for a wide range of engineering applications [19]. These tools are of great interest for conveying intuitive structural insight at bachelor level, as they are conceived as black-box software tools that allow users to perform basic analysis with little prior knowledge.

At master level, students have acquired basic structural insight and structural engineering education now focuses on computational methods such as matrix structural analysis and the finite element method. In terms of educational objectives and in line with Bloom's taxonomy [20], master students should extend their basic structural insight in order to apply structural analysis methods to actual engineering designs and to analyze and evaluate the results obtained. A critical attitude towards computational methods remains crucial. In a traditional teaching approach, master students study the basic principles of matrix structural analysis and the finite element method, employing the virtual work principle to derive the stiffness matrix of simple structures consisting of a few elements. They also use commercial finite element software packages to analyze more complex structural problems. A drawback of using these packages is the limited availability of intermediate results from the different stages of the analysis procedure, so that the links between theory and results may remain unclear to many students.

In order to elucidate the link between a computer analysis and the underlying theory, dedicated programming languages have been developed. Notable examples include the Computer-Assisted Learning (CAL) language [21, 22, 23] and the Language for the Analysis of Structures (LAS) [24]. These languages are conceived as integral learning environments, where students program the entire structural analysis, defining nodes, elements, and degrees of freedom, assembling the stiffness matrix and load vector, solving the resulting system of equations, and post-processing the results. CAL was developed at UC Berkeley in the 1970s and 1980s [21, 22, 23] and distributed as freeware to encourage further development. A modified version CALm was developed at KU Leuven in the late 1980s [25] and used for more than a decade in the Civil and Architectural Engineering curricula. The CALm learning environment was based

on creating a text-based input file that was processed using DOS commands to generate text-based output files and vector graphics. Despite its many advantages, the use of text-based input and output files soon became outdated. This was our main motivation to develop a successor, Stabil, taking into consideration the following prerequisites:

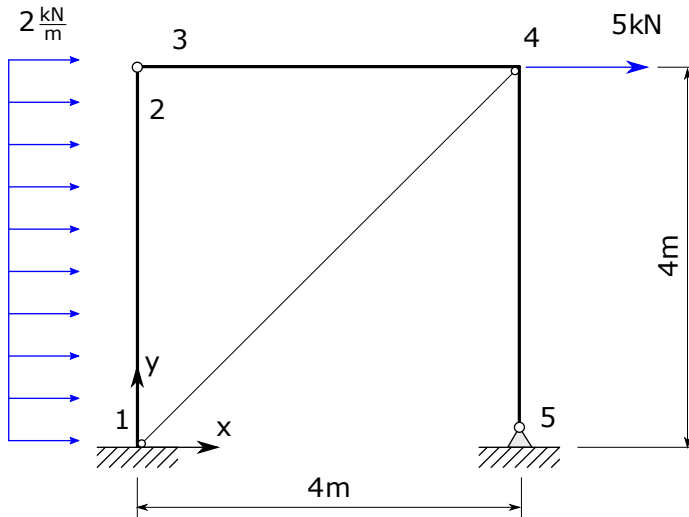
1. The different steps of structural analysis are as transparent as possible to the users.
2. The software is provided as open-source. All subroutines are available to students in order to effectively link theory to computer implementation. This also allows users to add custom functions.
3. Post-processing tools provide accurate plots of the deformed structure and member forces.
4. Stabil is developed in Matlab [26], a high-level programming language centered around matrix computations, allowing for the efficient solution of (sparse) linear systems of equations. KU Leuven's engineering students are familiarized with the Matlab environment early in their curriculum.

The objective of this paper is to explain the main concepts of the Stabil toolbox and to demonstrate how the toolbox has been integrated in several of the Civil and Architectural Engineering curricula at KU Leuven. The paper is structured as follows. Section 2 explains the basic concepts of the Stabil toolbox using an example problem and discusses the technical features and possibilities. Section 3 gives an overview of the use of the toolbox throughout the Civil and Architectural Engineering curricula. Conclusions are presented in section 4.

## 2 | THE STABIL TOOLBOX

Stabil is a Matlab toolbox for structural analysis, based on the finite element method. The element library includes truss, beam, plate, and continuum elements and the toolbox can be used to solve a wide variety of static and dynamic structural problems. Stabil contains a number of post-processing functions that are specifically designed for structural analysis, including the plotting of displacements and member forces. As users need to program the sequence of operations, they are more conscious of the different steps in a finite element analysis. The user can use Stabil at a low- or high-level of abstraction, which is further explained in the following. Due to this multi-level approach, the toolbox is suited for educational purposes as well as for use in a research environment: the high-level functions allow for an easy and efficient implementation of common structural problems, while the low-level functions facilitate the understanding of structural analysis methods in a teaching context, as well as implementation of novel finite element techniques in a research environment. While Stabil was developed for the Matlab environment, it is also compatible with the open source GNU Octave Scientific Programming Language [27].

The basic concepts of the Stabil toolbox are introduced by means of a two-dimensional (2D) reinforced concrete frame structure, shown in figure 1. The frame has a height and width of 4 m and is clamped at the left and pin-connected at the right support. There is an internal hinge at the top left corner of the frame, while a diagonal steel brace connects the lower left with the upper right corner. The concrete beams and columns have a rectangular cross section with a width of 0.2 m and a height of 0.4 m. The concrete has a Young's modulus of 30 GPa and a Poisson's ratio of 0.2. The steel brace has a circular cross section with a diameter of 8 mm. The steel has a Young's modulus of 210 GPa and a Poisson's ratio of 0.3. At the top right corner, a horizontal point load  $F = 5$  kN is applied, while the left column is loaded by a distributed lateral load  $p = 2$  kN/m.



**FIGURE 1** Reinforced concrete frame structure under lateral loading.

## Preprocessing

The Stabil code to compute the displacements and member forces in the frame is listed in figure 2. After the definition of some model parameters, the nodes are defined by a matrix that contains the number and  $x$ -,  $y$ - and  $z$ -coordinates of each node, arranged in four columns. As the nodes of the frame are located in the  $(x, y)$ -plane, all  $z$ -coordinates are zero. Next, the element types are defined by a cell array, `Types`, containing element type numbers followed by a string (either `beam` or `truss`). The `Sections` matrix defines the section properties (cross section, moments of inertia,...). In a similar way, the `Materials` matrix defines material properties (Young's modulus and Poisson's ratio). The model definition is completed by providing an element connectivity table, which refers to the previously defined element type, material, section and node numbers. In order to uniquely define a local coordinate system for the beam elements, use is made of a reference node (node 6). The local  $x$ -axis is directed from the first node of the element to the second node, while the local  $y$ -axis is perpendicular to the local  $x$ -axis with its origin corresponding to the first node, so that the reference node has a positive local  $y$ -coordinate.

Next, the degrees of freedom (DOFs) are specified. The DOFs are referred to by a decimal number `node.index`, where the digits of `node` refer to the node number, while the digits of `index` refer to the degree of freedom. By convention, the degrees of freedom 01, 02, and 03 correspond to the translations  $u_x$ ,  $u_y$ , and  $u_z$  in the global frame of reference, whereas the degrees of freedom 04, 05, and 06 correspond to the rotations  $\varphi_x$ ,  $\varphi_y$ , and  $\varphi_z$  around the global coordinate axes. For example, DOF 3.02 represents the translation  $u_y$  of node 3. This format enforces students to reflect on the kinematics of a structure. The explicit definition of the vector of degrees of freedom is considered as a low-level functionality of the toolbox. Once students have established a basic understanding of degrees of freedom, high-level functions (`getdof`, `selectdof`, `removedof`) are provided to generate and process the DOF vector so that more complex structural models can be efficiently analyzed.

```

b=0.20;           % Beam width [m]
h=0.40;           % Beam height [m]
r=0.004;          % Tendon radius [m]

%      NodeID  x  y  z
Nodes= [1      0  0  0
        2      0  4  0
        3      0  4  0
        4      4  4  0
        5      4  0  0
        6      2  2  0]; % Reference node

%      TypeID  Type
Types={1      'beam'
       2      'truss'};

%      SecID  A      ky      kz      Ixx  Iyy  Izz
Sections= [1      b*h      Inf      Inf      0      0      b*h^3/12
          2      pi*r^2 NaN      NaN      NaN      NaN      NaN ];

%      MatID  E      nu
Materials= [1      30e6  0.2      % concrete
           2      210e6 0.3;      % steel

%      EltID  TypID  SecID  MatID  n1  n2  n3
Elements= [1      1      1      1      1  2  6
          2      1      1      1      3  4  6
          3      1      1      1      5  4  6
          4      2      2      2      1  4  NaN];

dof=[2.01 2.02 2.06 3.01 3.02 3.06 4.01 4.02 4.06 5.06]; % Degrees of freedom

K=asmm(Nodes,Elements,Types,Sections,Materials,dof); % Stiffness matrix K

%      EltID  n1globalX  n1globalY  n1globalZ ...
DLoads= [1      2  0  0  2  0  0];

P=nodalvalues(dof,4.01,5)+elemloads(DLoads,Nodes,Elements,Types,dof);

%      Constant = Coef1 * dof1 + Coef2 * dof2 ...
constraints= [0      1      2.01  -1      3.01;
             0      1      2.02  -1      3.02];

[K,P]=addconstr(constraints,dof,K,P);

u=K\P;

figure
plotdisp(Nodes,Elements,Types,dof,u,DLoads,Sections,Materials)

Forces=elemforces(Nodes,Elements,Types,Sections,Materials,dof,u,DLoads);

figure
plotforc('norm',Nodes,Elements,Types,Forces,DLoads)

figure
plotforc('sheary',Nodes,Elements,Types,Forces,DLoads)

figure
plotforc('momz',Nodes,Elements,Types,Forces,DLoads)

```

**FIGURE 2** Stabil code for the calculation of a reinforced concrete frame structure under lateral loading.

```

function [KeLCS,MeLCS] = kelcs_truss(L,A,E,rho,Options)
%KE LCS TRUSS Truss element stiffness and mass matrix in local coordinate system.
%
% [KeLCS,MeLCS] = kelcs_truss(L,A,E,rho,Options)
% KeLCS = kelcs_truss(L,A,E)
% returns the element stiffness and mass matrix in the local coordinate system
% for a two node truss element (isotropic material)
%
% L      Truss length
% A      Truss cross section
% E      Young's modulus
% rho    Mass density
% Options Options for the mass matrix: {'lumped'}
% KeLCS  Element stiffness matrix (6 * 6)
% MeLCS  Element mass matrix (6 * 6)
%
% See also KE_TRUSS, KELCS_BEAM.

% STIFFNESS MATRIX
KeLCS=[E*A/L 0 0 -E*A/L 0 0
        0 0 0 0 0 0
        0 0 0 0 0 0
        -E*A/L 0 0 -E*A/L 0 0
        0 0 0 0 0 0];

% MASS MATRIX
if nargin>1
    if (nargin<5), Options={}; end
    if ~any(strcmpi(Options,'lumped')) % Consistent mass matrix
        MeLCS=[rho*A*L/3 0 0 rho*A*L/6 0 0
                0 rho*A*L/3 0 0 rho*A*L/6 0
                0 0 rho*A*L/3 0 0 rho*A*L/6
                rho*A*L/6 0 0 rho*A*L/3 0 0
                0 rho*A*L/6 0 0 rho*A*L/3 0
                0 0 rho*A*L/6 0 0 rho*A*L/3];
    else % Lumped mass matrix
        MeLCS=diag(rho*A*L/2*[1 1 1 1 1 1]);
    end
end
end

```

**FIGURE 3** Stabil code for the calculation of the element stiffness matrix of a truss element in the local coordinate system.

## Calculation

The finite element stiffness matrix is assembled using the `asmkm` function. This function takes the `Nodes`, `Elements`, `Types`, `Sections`, and `Materials` variables that define the finite element model and assembles the sparse global stiffness matrix  $K$ , in correspondence with the `doF` vector. The `asmkm` function is a high-level function that loops over the various elements of the mesh and calls low-level functions that generate element stiffness matrices.

The source code of both low- and high-level Stabil functions is available to students and some functions are discussed in detail during the courses. As an example, the low-level functions `kelcs_truss`, listed in figure 3, computes the element stiffness matrix of a truss element in the local element coordinate system. This function is called by the function `ke_truss`, listed in figure 4, where the element stiffness matrix is rotated to yield the element stiffness matrix in the global coordinate system. Students can run these functions to study their functionality and possibly modify them to fit their needs, e.g. within the framework of Master's theses.

Functions are available to either generate nodal loads on DOFs (`nodalvalues` function) or distributed loads on elements (`elemloads` function). The load vector  $P$  is available in the Matlab environment, so that students can check its dimensions and the defined loads in relation to the `doF` vector.

In order to account for the internal hinge at the upper left corner of the frame, two nodes (nodes 2 and 3) with 3 degrees of freedom are defined and joined by means of constraint equations, that couple the horizontal and vertical displacements of both nodes, but leave the rotations independent. These constraint equations are imposed on the system of equations by means of the function `addconstr` that modifies the stiffness matrix  $K$  and load vector  $P$ .

```

function [Ke,Me] = ke_truss(Node,Section,Material,Options)
%KE_TRUSS Truss element stiffness and mass matrix in global coordinate system.
%
% [Ke,Me] = ke_truss(Node,Section,Material,Options) returns the element
% stiffness and mass matrix in the global coordinate system
% for a two node truss element (isotropic material)
%
% Node      Node definitions      [x1 y1 z1; x2 y2 z2] (2 * 3)
% Section   Section definition    [A]
% Material  Material definition    [E nu rho]
% Options   Element options       {Option1 Option2 ...}
% Ke        Element stiffness matrix (6 * 6)
% Me        Element mass matrix     (6 * 6)
%
% See also KELCS_TRUSS, TRANS_TRUSS, ASMKM, KE_BEAM.

% Input argument processing
if (nargin<4), Options={}; end
if ~all(isfinite(Node(1:2,1:3))), error('Not all nodes exist.') end

% Element length
L=norm(Node(2,:)-Node(1,:));

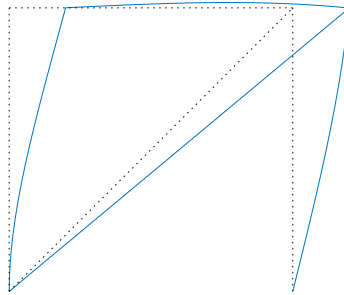
% Material properties
E=Material(1);
A=Section(1);

% Transformation matrix
t=trans_truss(Node);
T=blkdiag(t,t);
if nargin>1 % stiffness and mass
    rho=Material(3);
    [KeLCS,MeLCS]=kelcs_truss(L,A,E,rho,Options);
    Ke=T.'*KeLCS*T;
else % only stiffness
    KeLCS=kelcs_truss(L,A,E);
    Ke=T.'*KeLCS*T;
end

```

**FIGURE 4** Stabil code for the calculation of the element stiffness matrix of a truss element in the global coordinate system.





**FIGURE 5** Deformed reinforced concrete frame structure under lateral loading.

The finite element system of equations ( $Ku = P$ ) is solved using the Matlab backslash (left divide, `mldivide`) command. Since the global stiffness matrix  $K$  is sparse, a sparse solver is used by Matlab by default. The resulting displacement vector  $u$  has the same size as the `dof` vector and the load vector  $P$  and is available in the Matlab workspace for further processing.

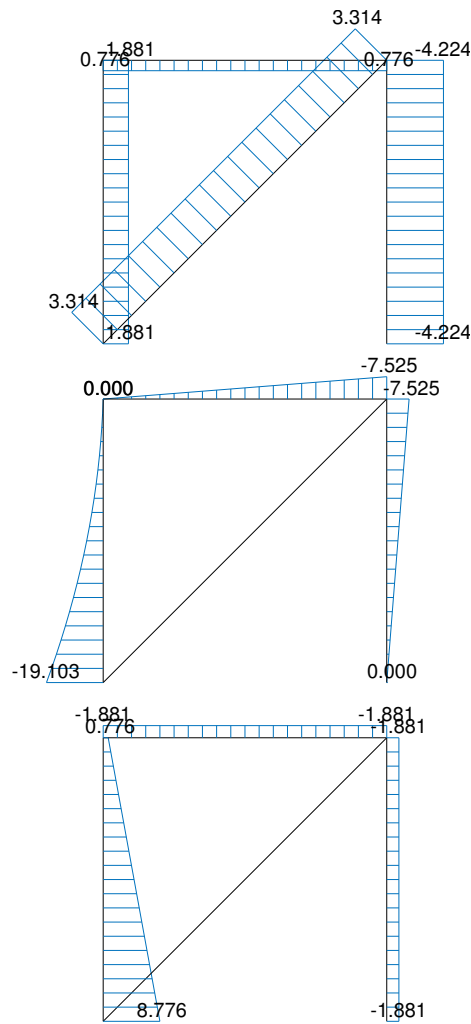
## | Post-processing

Post-processing functions are available to plot the deformed structure and the member forces. Figure 5 shows the deformed shape of the frame as obtained with the `plotdisp` command. This command uses the model definition (`Nodes`, `Elements`, `Types`), the displacement vector and the corresponding `dof` vector. A key feature of the Stabil toolbox is that the deformed shape of beams is plotted exactly, accounting for the cubic shape functions and the deformation of the elements due to distributed loads. In the latter case, the `plotdisp` command also takes the distributed loads and the section and material definitions (`Sections` and `Materials`) as optional input. Figure 6 shows the normal forces, bending moments, and shear forces in the frame structure as plotted with the `plotforc` command. This command also accounts for the effect of distributed loads, as is apparent in the quadratic variation of the bending moment in the left column (figure 6b).

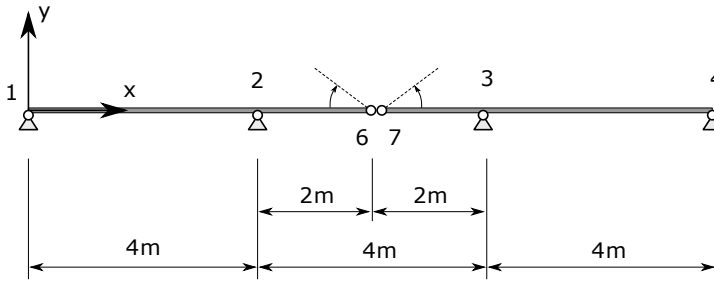
The above example shows how students are able to perform a basic structural analysis with little effort. The limited number of lines of code makes it easy to clearly explain the order of operations of the computation.

## 3 | THE USE OF STABIL THROUGHOUT THE CURRICULUM

The courses Strength of Materials and Structural Analysis form a key component of the Bachelor programs in Civil and Architectural Engineering at KU Leuven. These courses aim at acquiring structural insight at an expert level, including the analysis of member forces in iso-static and hyperstatic structures and the evaluation of stresses and strains over the cross section. The Master programs in Civil and Architectural Engineering build further upon this knowledge, with a focus on numerical techniques for structural analysis, dynamics of structures, and project work. This section explains how the Stabil toolbox is embedded in the curricula to support structural engineering courses. This is illustrated with several examples.



**FIGURE 6** (a) Normal forces [kN], (b) bending moments [kNm], and (c) shear forces [kN] in the reinforced concrete frame structure under lateral loading.



**FIGURE 7** Multiply supported beam with three spans. In order to compute the influence line, two nodes 6 and 7 are defined at midspan, and a unit relative rotation is applied at these nodes while their horizontal and vertical displacements are kept equal.

### 3.1 | Matrix Analysis of Structures

Civil and architectural engineering students are familiarized with Stabil in a course on Matrix Analysis of Structures. Students derive the stiffness matrix of 2D and 3D truss and beam elements using the virtual work theorem and Euler-Bernoulli and Timoshenko beam theory. Besides studying the underlying theory of matrix analysis of structures, as detailed in textbooks such as Weaver and Gere [28] Przemieniecki [29], Kassimali [30], and Doyle [31], the general structure and practical implementation of a finite element formulation for truss and beam elements is systematically taught by means of the Stabil toolbox, using examples such as the frame structure discussed in section 2. Modeling of rigid offset connections by means of constraint equations is also considered.

The following example illustrates the calculation of the influence line for the bending moment at midspan of a three-span beam for a vertical point load (figure 7). The Stabil input file is listed in figure 8. Following the Müller-Breslau principle, which is based on Maxwell-Betti's reciprocity theorem, two nodes 6 and 7 are defined at midspan, and a unit relative rotation is applied at these nodes while their horizontal and vertical displacements are kept equal. These kinematic constraints are added to the finite element system of equations using the `addconst` function. The resulting system of equations is solved, and the vertical displacement along the beam (figure 9) corresponds to the influence line, representing the bending moment at midspan caused by a unit load at the corresponding position along the beam.

The course of Matrix Analysis of Structures is organized through ex cathedra lectures and classroom exercise sessions. During these exercise sessions, practical problems are solved by means of analytical methods as well as the numerical toolbox Stabil. Students are encouraged to interact with teaching assistants and fellow students, aiming at gaining a deeper insight in structural analysis. During the semester, students work in smaller groups on an assignment where a structural engineering problem, inspired by existing truss and frame structures, is solved with Stabil. An oral exam at the end of the semester finally assesses each individual student's background and knowledge in structural analysis.

### 3.2 | Dynamics of Structures

In the course on Dynamics of Structures, students learn to analyze the response of structures under dynamic excitation using the finite element method. The course is supported by textbooks on the subject, including Chopra [32] and Clough and Penzien [33]. Several solution methods are discussed, such as response history analysis by means of modal superposition in the time and frequency domain or direct time integration, as well as response spectrum analysis.

```

b=0.10; % beam width
h=0.25; % beam height

% Nodes=[ModID X Y Z]
Nodes= [1 0 0 0;
        2 4 0 0;
        3 8 0 0;
        4 12 0 0;
        5 0 -1 0; % reference node
        6 6 0 0;
        7 6 0 0];

% {EltTypeID EltName}
Types= {1 'beam'};

% [SecID A ky kz Ixx Iyy Izz]
Sections= [1 b*h Inf Inf 0 0 b*h^3/12];

% [MatID E nu];
Materials= [1 30e6 0.2]; % concrete

% [EltID TypID SecID MatID n1 n2 n3]
Elements= [1 1 1 1 1 2 5;
           2 1 1 1 2 6 5;
           3 1 1 1 7 3 5;
           4 1 1 1 3 4 5];

dof=getdof(Elements,Types);
seldof=[0.03;0.04;0.05;1.01;1.02;2.02;3.02;4.02];
dof=removedof(dof,seldof);

% Assembly of stiffness matrix K
K=asmm(Nodes,Elements,Types,Sections,Materials,dof);

% Assembly of the (empty) load vector
P=zeros(size(dof));

% Constraint equations: Constant=Coef1*DOF1+Coef2*DOF2+...
% [Constant Coef1 DOF1 Coef2 DOF2 ...]
Constr= [0 1 6.01 -1 7.01;
         0 1 6.02 -1 7.02;
         -1 1 6.06 -1 7.06];

[K,P]=addconstr(Constr,dof,K,P);

% Solve K * U = P
U=K\P;

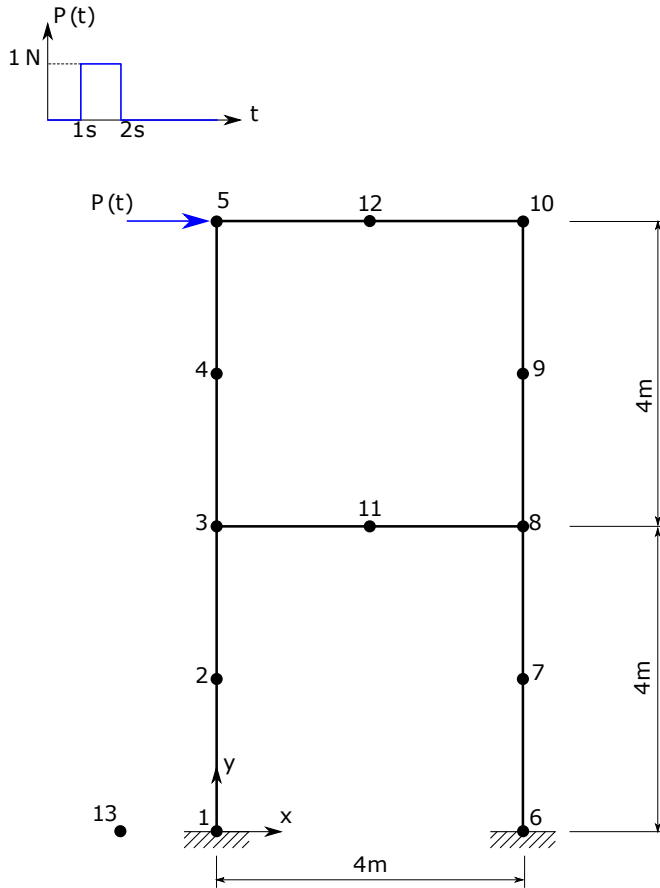
% Plot displacements
figure
plotdisp(Nodes,Elements,Types,dof,U)

```

**FIGURE 8** Stabil code for the calculation of an influence line for the bending moment at midspan of a three-span beam.



**FIGURE 9** Influence line for the bending moment at midspan of a three-span beam.



**FIGURE 10** Two-story steel portal frame structure loaded by a lateral unit rectangular pulse load.

During the ex cathedra lectures and exercise sessions, complexity is gradually increased by first considering single degree of freedom systems, followed by multiple degree of freedom systems. The latter are inspired by real-life case histories, such as buildings subjected to wind, seismic, or traffic loading, pedestrian bridges and grandstands subjected to moving or jumping people, and vibration isolation of sensitive equipment, machine foundations and railway tracks. During the exercise sessions, students apply different solution techniques to the dynamic analysis of a two-story portal frame subject to lateral and seismic loading. The assessment of the course consists of an assignment, where small groups of students solve a dynamic problem using Stabil, applying different solution techniques. This is followed by a final presentation in a mini-symposium attended by all students and teaching assistants.

As an example, figure 10 shows a two-story steel portal frame structure with a width  $L = 4$  m and story height  $H = 4$  m, subject by a lateral point load at the top left corner with a time history corresponding to a unit rectangular pulse of duration 1 s. Both columns are clamped at the base. The steel has a Young's modulus of 210 GPa, a Poisson's ratio of 0.3, and a density of  $7850 \text{ kg/m}^3$ . The columns have a cross section of  $0.012 \text{ m}^2$  and a moment of inertia of  $1 \times 10^5 \text{ m}^4$ , while the beams have a cross section of  $0.014 \text{ m}^2$  and a moment of inertia of  $2 \times 10^5 \text{ m}^4$ . Proportional damping with a (relatively high) constant modal damping ratio of 0.08 is assumed.

The Stabil code to compute the response of the structure by means of modal superposition in the frequency domain is listed in figure 11. Similar as in previous examples, the analysis starts by defining nodes, element types, materials, elements, and degrees of freedom. The stiffness and mass matrices of the structure are obtained from the `asmkm` function. The time and frequency axis are uniformly sampled in view of subsequent Fast Fourier Transforms. The eigenmodes and eigenfrequencies are computed using the `eigfem` function. The eigenmodes are available as columns of the matrix `phi` and can be plotted and animated with the `plotdisp` and `animdisp` functions (figure 12).

The response of the two-storey steel portal frame due to the lateral unit rectangular pulse load is computed by modal superposition in the frequency domain. The analysis starts by assembling the time history of the load in the vector `q`. The frequency content of the load is computed by means of the Fast Fourier Transform algorithm and stored in a vector `Q`. After computation of the modal participation factors `Gamma`, a modal superposition analysis in the frequency domain is performed using the Stabil function `msupf`. An inverse Fast Fourier Transform of the modal coordinates `X` results in the time history of the modal coordinates `x`. The time history of the nodal displacements `u` is finally computed as the product of the modes `phi` and the time history of the modal coordinates `x`. The Stabil function `msupf` also computes the modal transfer functions `H` in the frequency domain.

It is very instructive for students to have access to all intermediate results of the modal superposition procedure. Figure 13 illustrates how the time history of the load `q` is transformed to the frequency domain `Q` and subsequently multiplied by the modal transfer function `H` to yield the modal coordinates `X` in the frequency domain, which are finally transformed back to the time domain to obtain the modal coordinates `x`. As students adjust the section properties of the columns and beams or the duration of the rectangular pulse load (in comparison with the highest eigenperiods of the structure), they can explore the influence of these parameters on the structural response.

### 3.3 | The Finite Element Method

The course on the Finite Element Method focuses on the formulation and application of finite elements for 2D and 3D linear elastic continuum problems, as well as Kirchhoff and Mindlin-Reissner plate theory. The theory of the finite element method is extensively discussed in various textbooks, such as Bathe [34], Cook [35, 36], or Zienkiewicz et al. [37]. The students are familiarized with the formulation of common finite element types and learn how to apply them, with a focus on checking and interpreting the results obtained.

Considering the implementation of finite elements, the course focuses on shape functions, isoparametric transformations, numerical (Gauss-Legendre) integration, and element technology. Essential functionalities are discussed in detail, as illustrated in figure 14 for the numerical integration of a 4-node quadrilateral plane element in the `ke_plane4` function. The integration of the element stiffness matrix is based on the evaluation of the Jacobian and its determinant at all Gaussian integration points, summing up products  $B^T * D * B * H_g(1, i_{Gauss}) * \det J$  of the derivatives `B` of the shape functions, the constitutive matrix `D`, the Gaussian weights `Hg` and the determinant of the Jacobian `detJ`.

A typical example problem considered in the course is the computation of the stress concentration around a circular hole in a square plate subjected to uniaxial loading along the  $x$ -axis (figure 15). Students are already familiar with the classical linear elastic solution by Kirsch [38] for a stress concentration around a circular hole in a plate with infinite dimensions, and hence will understand the need for mesh refinement around the edge of the hole where the factor-of-three stress concentration is expected. The plate is made of Steel with a Young's modulus of 210 GPa and a Poisson's ratio of 0.2 and has finite dimensions of  $8 \times 8$  cm. The hole has a radius of 2 cm.

The finite element mesh consists of three-node linear finite elements, also known as constant strain triangles, as the resulting strains and stresses are constant throughout the element. Since the amount of code is rather limited, this introductory problem allows students to learn the basic structure of a finite element calculation. Figure 16 shows

```

Types={1 'beam'};
Sections=[1 0.012 Inf Inf 0 0 1E-5;
          2 0.014 Inf Inf 0 0 2E-5];
Materials=[1 210E9 0.3 7850];
Nodes=[ 1 0 0 0
        2 0 2 0
        3 0 4 0
        4 0 6 0
        5 0 8 0
        6 4 0 0
        7 4 2 0
        8 4 4 0
        9 4 6 0
        10 4 8 0
        11 2 4 0
        12 2 8 0
        13 -1 0 0];
Elements=[1 1 1 1 1 1 2 13
          2 1 1 1 2 3 13
          3 1 1 1 3 4 13
          4 1 1 1 4 5 13
          5 1 1 1 6 7 13
          6 1 1 1 7 8 13
          7 1 1 1 8 9 13
          8 1 1 1 9 10 13
          9 1 2 1 3 11 13
          10 1 2 1 11 8 13
          11 1 2 1 5 12 13
          12 1 2 1 12 10 13];

% DEGREES OF FREEDOM
DOF=getdof(Elements,Types);
seldof=[0.03; 0.04; 0.05; 1.00; 6.00];
DOF=removedof(DOF,seldof);

% STIFFNESS AND MASS MATRICES
[K,M]=asmm(Nodes,Elements,Types,Sections,Materials,DOF);

% TIME AND FREQUENCY SAMPLING
N=8192; % Number of samples
dt=0.0025; % Time step
T=N*dt; % Period
F=N/T; % Sampling frequency
df=1/T; % Frequency resolution
t=[0:N-1]*dt; % Time axis
f=[0:N/2-1]*df; % Positive frequencies corresponding to FFT [Hz]
Omega=2*pi*f; % Idem [rad/s]

% EIGENVALUE ANALYSIS
nMode=2; % Number of modes to take into account
[phi,omega]=eigfm(K,M,nMode); % Calculate eigenmodes and eigenfrequencies
xi=0.08; % Constant modal damping ratio

% EXCITATION
b=nodalvalues(DOF,5.01,*1); % Spatial distribution, nodal (nDOF * 1)
Gamma=phi.'*b; % Spatial distribution, modal (nMode * 1)

q=zeros(1,N); % Time history (1 * N)
q((t>=1) & (t<2))=1; % Time history (1 * N)
Q=fft(q); % Frequency content (1 * N)
Q=Q(1:N/2); % Frequency content, positive freq (1 * N/2)
Pm=Gamma*Q; % Modal excitation, positive freq (nMode * N/2)

% MODAL ANALYSIS
[X,H]=msupf(omega,xi,omega,Pm); % Modal response, positive freq (nMode * N/2)

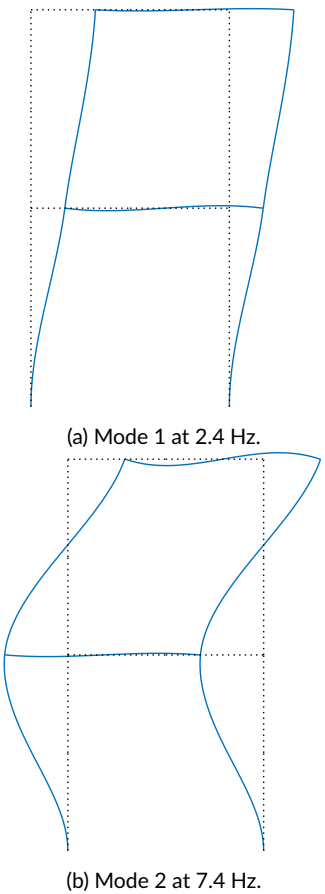
% FOURIER TRANSFORM FROM FREQUENCY TO TIME DOMAIN
X=[X, zeros(nMode,1), conj(X(:,end:-1:2))]; % Add negative frequencies
x=ifft(X,[],2); % Modal displacements (nMode * N)
H=[H, zeros(nMode,1), conj(H(:,end:-1:2))];
h=ifft(H,[],2);

% MODAL DISPLACEMENTS -> NODAL DISPLACEMENTS
U=phi*X; % Nodal displacements (nDOF * N)
u=phi*x;

% PLOT AND ANIMATE THE MODES
figure; % mode 1
plotdisp(Nodes,Elements,Types,DOF,phi(:,1));
figure; % mode 2
plotdisp(Nodes,Elements,Types,DOF,phi(:,2));

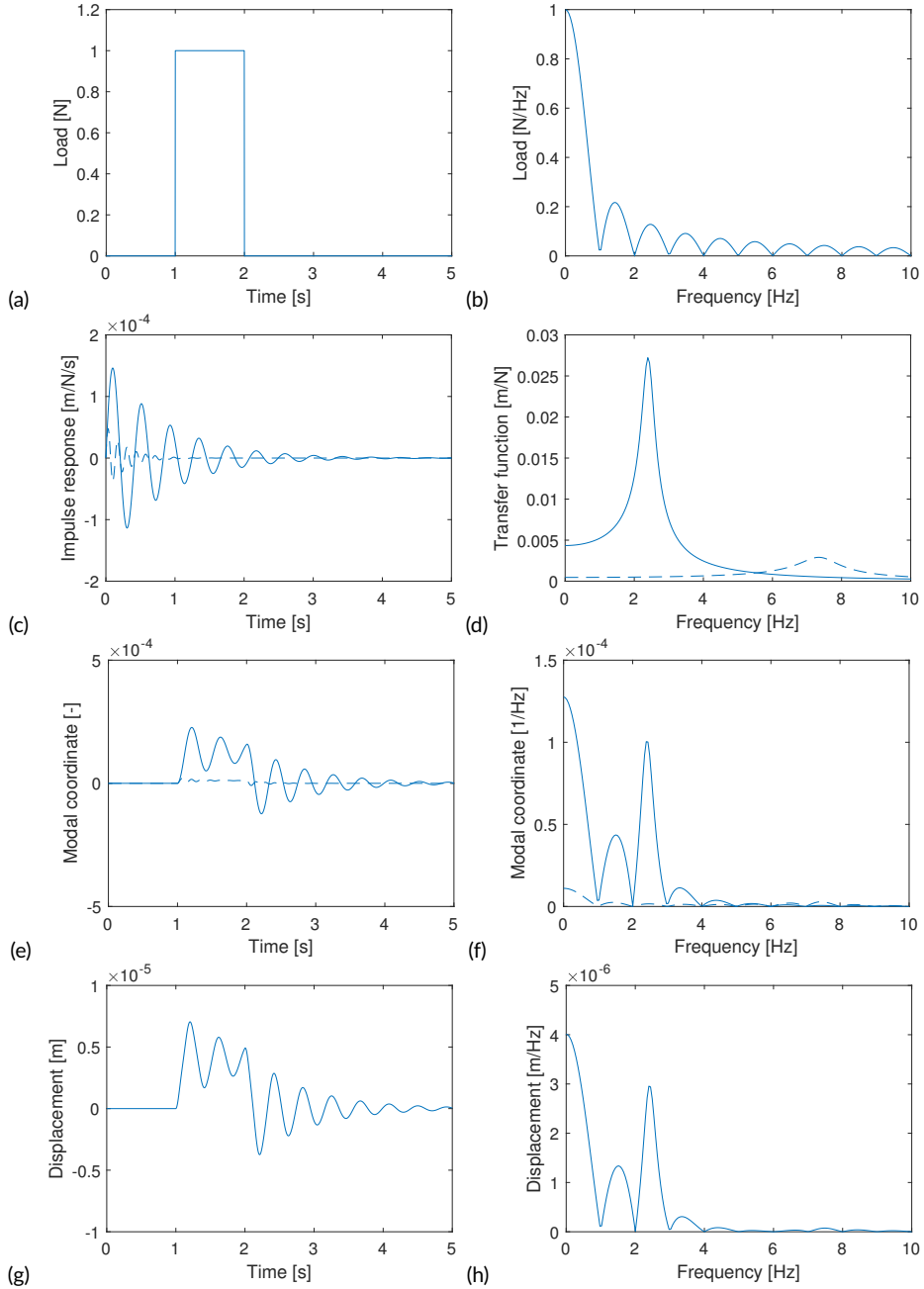
```

**FIGURE 11** Stabil code for the calculation of the dynamic response of a two-story steel portal frame using modal superposition in the frequency domain.



**FIGURE 12** (a) First and (b) second lateral mode of the two-story steel portal frame.





**FIGURE 13** (a) Time history and (b) frequency content of the load, (c) modal impulse response and (d) modal transfer function, and (e) time history and (f) frequency content of the modal coordinate. The results are plotted for mode 1 (solid line) and mode 2 (dashed line). (g) Time history of the vertical displacement at the load point and (h) corresponding frequency content are also shown.

```

for iGauss = 1:nGauss
    xi = xg(iGauss,1);
    eta = xg(iGauss,2);

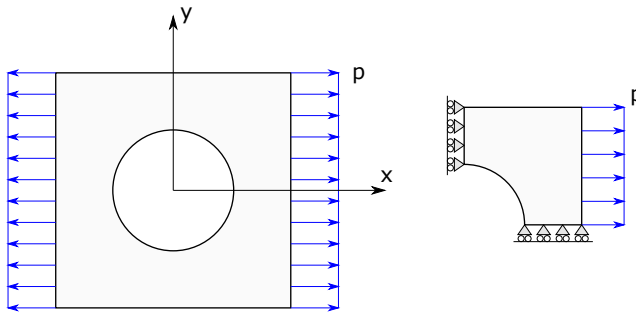
    % Shape functions, Jacobian, ...
    [Ni,dN_dxi,dN_deta] = sh_qe4(xi,eta);
    J=[dN_dxi.' *Node(:,1) dN_dxi.' *Node(:,2);
        dN_deta.' *Node(:,1) dN_deta.' *Node(:,2)];
    detJ=det(J);
    dNi=J\[dN_dxi dN_deta].';

    % Shape function derivatives
    B=zeros(3,8);
    B(1,1:2:7)=dNi(1,:);
    B(2,2:2:8)=dNi(2,:);
    B(3,1:2:7)=dNi(2,:);
    B(3,2:2:8)=dNi(1,:);

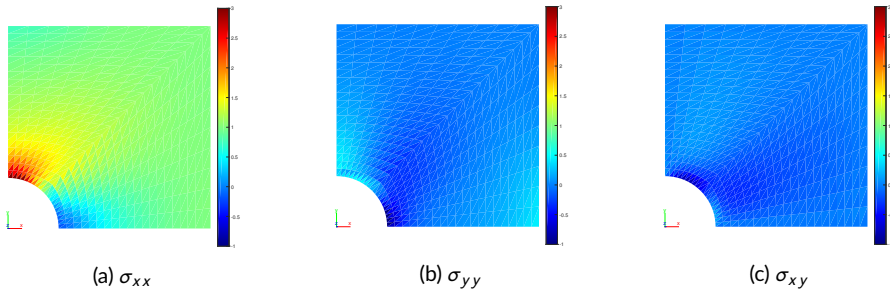
    % Add contribution to element stiffness matrix
    K11=K11+B.'*D*B*Hg(1,iGauss)*detJ;
end

```

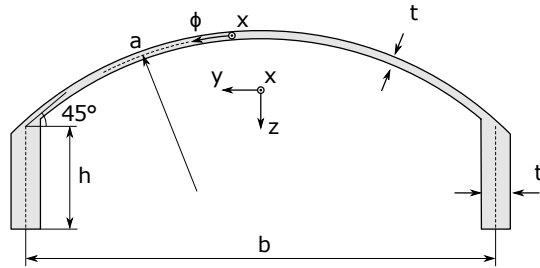
**FIGURE 14** Stabil code for Gaussian integration in the function `ke_plane4`.



**FIGURE 15** Square plate with a circular hole, subjected to uniaxial loading along the  $x$ -axis.



**FIGURE 16** Stresses (a)  $\sigma_{xx}$ , (b)  $\sigma_{yy}$ , and (c)  $\sigma_{xy}$  in a square plate with a circular hole, subjected to a uniform traction along the  $x$ -axis. Figure (a) illustrates the stress concentration at the edge of the hole.



**FIGURE 17** Concrete barrel vault under gravity loading, adopted from Frey and Studer [41].

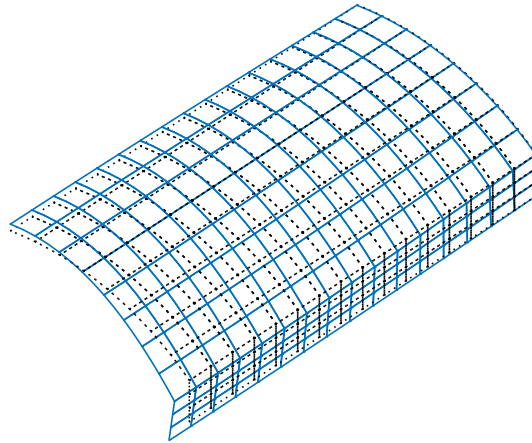
the resulting normal stresses  $\sigma_{xx}$  and  $\sigma_{yy}$ , as well as the shear stress  $\sigma_{xy}$ . Figure 16a clearly illustrates that the tensile stress at the edge of the hole is about three times larger than the applied uniform traction. The finite dimensions of the plate, together with the finite element discretization used, however, explain the deviations from Kirsch's classical solution.

### 3.4 | Theory of Shell and Spatial Structures

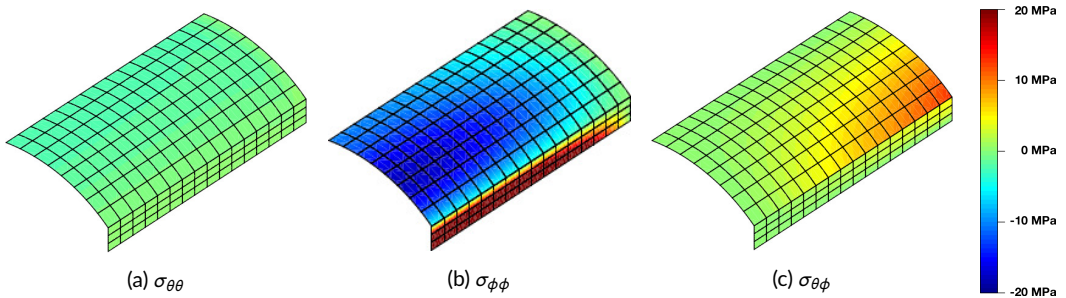
The optional course on Theory of Shell and Spatial Structures starts by deriving the governing equations of curved shells, as discussed in classical textbooks on the subject [39, 40], subsequently discussing the distribution of member forces in typical structures such as domes, cylinders, and folded plates. The finite element implementation of curved shell elements is also discussed. Next, students model and analyze an actual shell structure by means of the finite element method in Stabil.

An example of such assignment is the analysis of a concrete barrel vault under gravity loading, as shown in figure 17 [41]. The concrete barrel vault consist of a curved slab supported by edge beams in the longitudinal direction. The curved slab has a span  $b = 9.9$  m, a radius  $a = 7$  m, and a thickness  $t = 0.08$  m. The edge beams are characterized by a height  $h = 1.05$  m and a thickness  $t' = 0.22$  m. The structure has a length  $L = 25$  m in the longitudinal  $x$ -direction. The concrete has a Young's modulus of 30 GPa, a Poisson's ratio of 0.2, and a unit weight of  $25 \text{ kN/m}^3$ . The vault has end diaphragms with a large in-plane stiffness, but negligible out-of-plane stiffness, so that  $u_y = u_z = 0$  applies.

The concrete barrel vault is modeled in Stabil using shell elements, providing a reference solution for displacements (figure 18) and stresses (figure 19) in the local coordinate system  $(x, \phi)$  (figure 17). The displacement solution



**FIGURE 18** Displacements of the concrete barrel vault under gravity loading.



**FIGURE 19** Stresses (a)  $\sigma_{\theta\theta}$ , (b)  $\sigma_{\phi\phi}$ , and (c)  $\sigma_{\theta\phi}$  in the concrete barrel vault under gravity loading.

shows that the edge beams twist around their axis due to the weight of the structure. The curved shell is not stiff enough to prevent this twisting of the edge beams, resulting in a deformation of the cross section. Furthermore, this induces compressive stresses in the curved slab.

Since the length of the barrel vault is much larger than its other dimensions, it is reasonable to alternatively analyze the vault as a beam with a curved cross section. The aim of the assignment is to compare the results from the finite element analysis with such a solution obtained from beam theory, and to study the effect of the edge beams. Explaining the differences and similarities between various modeling strategies, and understanding the kinematic assumptions of shell theory is an important part of the course on Theory of Shell and Spatial Structures. An in-depth analysis of a case study in Stabil as outlined above is a very effective way to achieve this objective.

### 3.5 | Integrated Project: Design of a Quay Wall

The master in Civil Engineering program at KU Leuven incorporates project based cooperative learning activities, referred to as integrated projects, where student teams elaborate multi-disciplinary projects, employing insights and skills acquired in fundamental courses. The Stabil toolbox is used in several of these projects to support structural analysis. Examples include the design of high rise buildings under wind loading, the design of a vibration sensitive floor in a clean room of a micro-electronics laboratory, the design of a sea-bound construction under hydrodynamic loading, and the design of a quay wall in a container terminal. We restrict the following discussion to this last project.

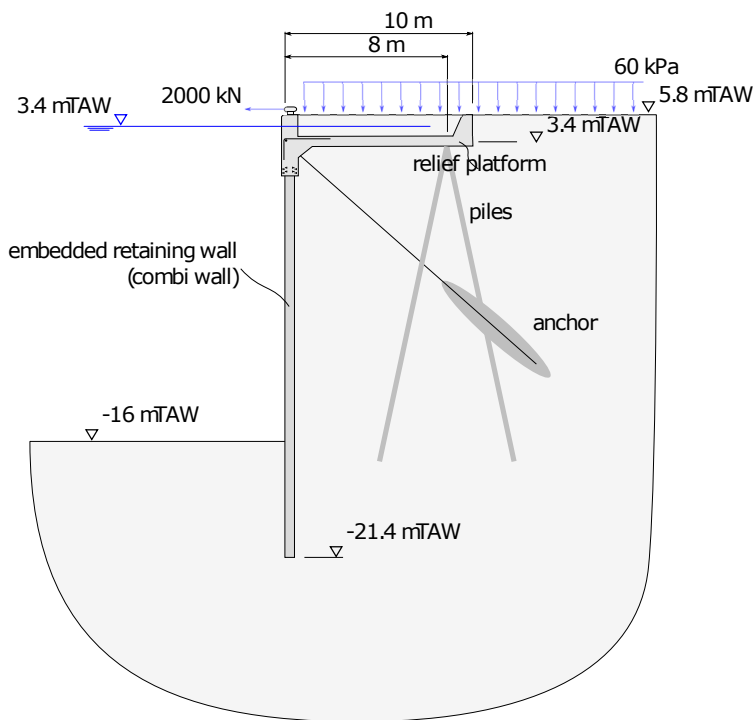
In this project, students start from site conditions, geotechnical parameters, water and ground levels, and static and dynamic loads to study different design alternatives for a quay wall to be constructed in a container terminal. The preliminary design is based on Blum's method [42] to compute the bending moment in embedded retaining walls. Students use Stabil to model the retaining wall with beam elements and impose active and passive soil pressures, as well as water pressures, on the wall.

Figure 20 shows the geometry of an anchored quay wall designed by a group of students for a case in the port of Bruges-Zeebrugge. The design specifications include the design depth of -16 mTAW, a ground level of 5.8 mTAW and a water level of 3.4 mTAW, where mTAW is the vertical datum used in Belgium. The quay wall should sustain a surface surcharge of 60 kPa and a mooring load of 2000 kN, in addition to horizontal soil pressures. The design consists of an embedded steel combi-wall that is rigidly connected to a relief platform. The combi-wall is composed of cylindrical foundation pipes with a diameter of 1.83 m and a wall thickness of 22 m, spaced 3.2 m apart and interconnected by sheet piling with negligible stiffness. The concrete relief platform has a thickness of 1 m and a width of 10 m. The relief platform is supported by a pile system of tension and bearing piles, while an additional grout anchor is connected to the capping beam. The pile system and grout anchor take up the horizontal mooring load and provide a horizontal support for the retaining wall. Furthermore, the relief platform bears the weight of the overlying soil and the surcharge load. This results in horizontal loading of the combi-wall and the pile system, and effectively reduces the soil stresses. As a result, the effective vertical soil stresses as shown on figure 21a are zero underneath the relief platform. In Blum's method, it is assumed that active soil stresses on the right side of the quay wall and passive soil stresses on the left side fully develop (figure 21b), resulting in the vertical loading of the wall as shown in figure 21c. Since the relief platform, the pile system, and the grout anchor take up both the surcharge and mooring loads, they are not considered here. Blum's method considers the bottom of the retaining wall to be clamped and the embedment depth is found as the depth at which the reaction moment is zero [43]. The corresponding deformation, bending moment, and shear forces are plotted in figures 21d to 21e.

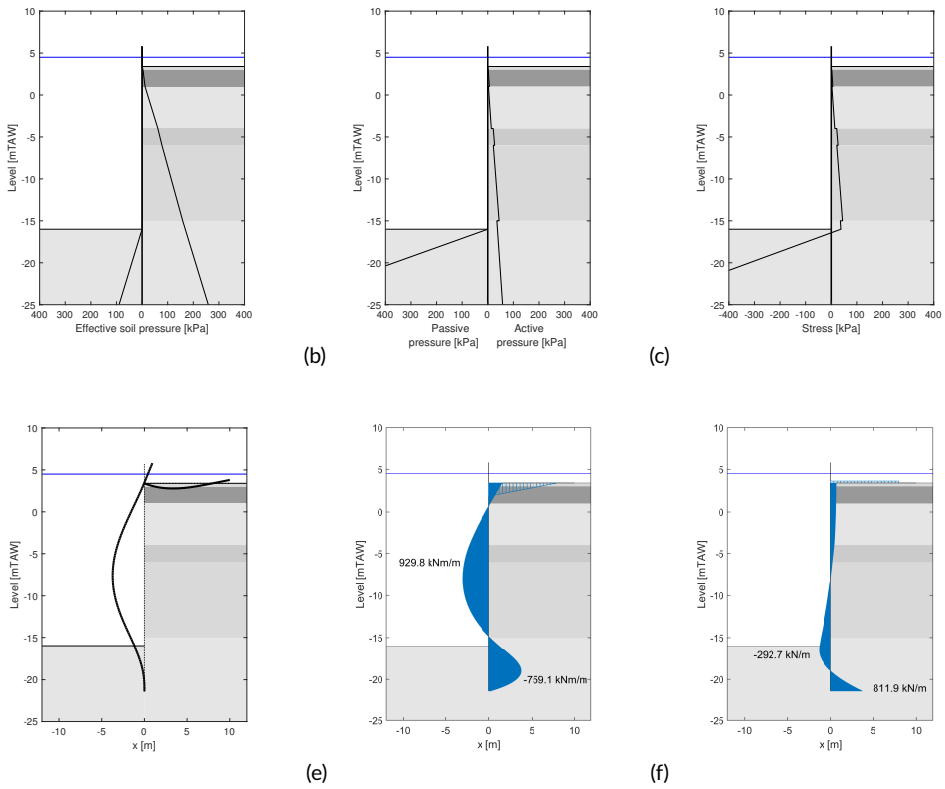
Students study the effect of the relief platform on the displacements and member forces in the retaining wall, apply Blum's method to determine the embedment depth, and dimension the retaining wall and the reinforcement in the relief platform.

The implementation of Blum's method takes a limited amount of time, while the added value compared to commercially available design tools is considerable as the basic assumptions and the overall sequence of calculations are very transparent. Intermediate results are available and contribute to increased insight in the structural behaviour and the design process.

The preliminary design using Blum's method is based on characteristic values of the loads, applying a global factor of safety. In a later phase of the project, load combinations using partial safety factors in accordance with the design codes are used. The envelope of bending moments can easily be computed and forms the basis for the design of the reinforcement. Results obtained with Stabil are also compared with a subgrade reaction analysis and a 2D non-linear finite element analysis that fully accounts for soil-structure interaction. Explaining the differences and similarities



**FIGURE 20** Geometry of the quay wall structure consisting of an embedded retaining wall and a relief platform.



**FIGURE 21** (a) Effective soil pressures, (b) passive pressures at the front and active soil pressures at the back of the wall, and (c) resulting distributed pressure on the quay wall. These soil pressures are used as an input for Blum's method, where the results are characterized by (d) displacements, (e) bending moments, and (f) shear forces in the embedded retaining wall. The water table is indicated in blue.

between results obtained with different models (in terms of assumption and complexity) also is an important learning outcome of this integrated project.

### 3.6 | MSc and PhD theses

Stabil is sufficiently flexible to be used in MSc and PhD theses, addressing a variety of problems in structural mechanics and structural optimization. As students master the key steps of a structural analysis, they easily customize and extend the Stabil toolbox for their specific problems. This has resulted in a variety of applications, ranging from uncertainty quantification in structural dynamics [44], optimization of frame structures [45, 46], geotechnical applications such as dynamic soil-structure interaction [47, 48] and modeling of grouted anchors in soils [49], to phase-field modeling of brittle and ductile fracture [50, 51].

## 4 | CONCLUSION

In this paper, we explained the main concepts of Stabil, a Matlab toolbox for static and dynamic structural analysis. The concepts of Stabil were introduced by a number of examples, demonstrating how students program basic static and dynamic structural analyses. The source code of all routines is available to students and discussed in detail throughout the different courses in order to effectively link theory to practical implementation and engineering applications. As students are familiar with Matlab from early in the curriculum, they are quickly acquainted with the practical use of Stabil.

The Stabil toolbox, the manual, and more examples can be downloaded free of charge from <http://www.kuleuven.be/bwm/stabil>, subject to a software license agreement. The toolbox can be used and modified for educational and research purposes. However, it cannot be used for consulting, nor commercialized in any form. Stabil is compatible with both Matlab and GNU Octave.

## Acknowledgements

Stabil was developed within the framework of the OOI project 2006/20 “An interactive and adaptive application for the static and dynamic analysis of structures”, funded by the KU Leuven Educational Policy Unit. This financial support is gratefully acknowledged.

## References

- [1] Brady S. Hartford stadium collapse: why software should never be more than a tool to be used wisely. *The Structural Engineer: journal of the Institution of Structural Engineer* 2015;93(8):20–22.
- [2] Pospíšil M, Vavrušková M, Veřtátová E. New Ways of Teaching Statics and Applied Structural Mechanics to Architects. In: *Applied Mechanics and Materials*, vol. 732 Trans Tech Publ; 2015. p. 417–420.
- [3] Karl-Eugen Kurrer ER Philip Thrift. *The History of the Theory of Structures: Searching for Equilibrium*. 2nd ed. Construction History, Ernst & Sohn; 2018.
- [4] Fernández-Sánchez G, Millán MÁ. Structural analysis education: Learning by hands-on projects and calculating structures. *Journal of Professional Issues in Engineering Education and Practice* 2013;139(3):244–247.
- [5] Emami N, von Buelow P. Teaching Structures to Architecture Students through Hands-On Activities. In: *Proceeding of the Canadian International Conference on Advances in Education, Teaching & Technology*; 2015. p. 1–7.



- [6] Van Mele T, Lachauer L, Rippmann M, Block P. Geometry-based understanding of structures. *Journal of the international association for shell and spatial structures* 2012;53(4):285–295.
- [7] Li B, Zhang M, Jin R, Wanatowski D, Piroozfar P. Incorporating Woodwork Fabrication into the Integrated Teaching and Learning of Civil Engineering Students. *Journal of Professional Issues in Engineering Education and Practice* 2018;144(4):05018007.
- [8] Aparicio AC, Ruiz-Teran AM. Tradition and innovation in teaching structural design in civil engineering. *Journal of Professional Issues in Engineering Education and Practice* 2007;133(4):340–349.
- [9] Romero ML, Museros P. Structural analysis education through model experiments and computer simulation. *Journal of Professional Issues in Engineering Education and Practice* 2002;128(4):170–175.
- [10] Estes AC, Baltimore C. Using K'Nex to Teach Large Scale Structures to Architects and Construction Students. In: 121st ASEE Annual Conference & Exposition Proceedings: Indianapolis, IN; 2014. p. 1–12.
- [11] Patil A, Abraham A. Intelligent and Interactive Web-Based Tutoring System in Engineering Education: Reviews, Perspectives and Development. *Computational Intelligence for Technology Enhanced Learning* 2010;p. 79–97.
- [12] Zechner J, Ebner M. Playing a game in civil engineering-The Internal Force Master for Structural Analysis. In: 14th International Conference on Interactive Collaborative Learning (ICL) IEEE; 2011. p. 417–422.
- [13] Mosquera JC, Plaza LF, González B. A framework for adaptive e-learning for continuum mechanics and structural analysis. In: Gómez L, López A, Candel I, editors. *Proceedings of the 9th International Technology, Education and Development Conference IATED Academy*; 2015. p. 1–8.
- [14] Pedron C. An innovative tool for teaching structural analysis and design. VDF Hochschulverlag AG; 2006.
- [15] Abrams M. Learn engineering using game apps. *Mechanical Engineering* 2013;135(7):12.
- [16] Holzinger A, Ebner M. Interaction and Usability of Simulations & Animations: A Case Study of the Flash Technology. In: *Interact*; 2003. p. 777–780.
- [17] McDonald CV. STEM Education: A Review of the Contribution of the Disciplines of Science, Technology, Engineering and Mathematics. *Science Education International* 2016;27(4):530–569.
- [18] Lucignano L. Augmented Reality to Facilitate a Conceptual Understanding of Statics in Vocational Education. PhD thesis, Ecole Polytechnique Fédérale de Lausanne; 2018.
- [19] Quesada C, González D, Alfaro I, Cueto E, Huerta A, Chinesta F. Real-time simulation techniques for augmented learning in science and engineering. *The Visual Computer* 2016;32(11):1465–1479.
- [20] Krathwohl DR. A revision of Bloom's taxonomy: An overview. *Theory into practice* 2002;41(4):212–218.
- [21] Wilson EL. CAL - A computer analysis language for teaching structural analysis. *Computers and Structures* 1979;10(1-2):127–132.
- [22] Oxley A. Computer Assisted Learning (CAL) of Structural Analysis. *Programmed Learning and Educational Technology* 1984;21(3):176–184.
- [23] Dahlblom O, Peterson A, Petersson H. CALFEM - A program for computer-aided learning of the finite element method. *Engineering Computations* 1986;3(2):155–160.
- [24] Paultre P, Lapointe E, Carboneau C, Proulx J. LAS: A programming language and development environment for learning matrix structural analysis. *Computer Applications in Engineering Education* 2016;24(1):89–100.

- [25] Geyskens P, Nijs B, Wens J, De Roeck G. CALM: Computer Analysis Language (Modified version). KU Leuven, Department of Civil Engineering; 1990.
- [26] Moore H. MATLAB for Engineers. Pearson; 2017.
- [27] Quarteroni A, Saleri F, Gervasio P. Scientific computing with MATLAB and Octave, vol. 2. Springer; 2006.
- [28] Weaver W, Gere JM. Matrix analysis framed structures. Springer science & business media; 2012.
- [29] Przemieniecki JS. Theory of matrix structural analysis, vol. 1. McGraw-Hill New York; 1968.
- [30] Kassimali A. Matrix analysis of structures SI version. Nelson Education; 2012.
- [31] Doyle JF. Static and dynamic analysis of structures: with an emphasis on mechanics and computer matrix methods, vol. 6. Springer Science & Business Media; 2012.
- [32] Chopra AK. Dynamics of structures: theory and applications to earthquake engineering. Englewood Cliffs, New Jersey: Prentice-Hall; 1995.
- [33] Clough RW, Penzien J. Dynamics of structures. Berkeley. CA: Computers and Structures 2003;.
- [34] Bathe KJ. Finite Element Procedures. second ed. Englewood Cliffs, NJ: Prentice-Hall; 1996.
- [35] Cook RD. Finite element modelling for stress analysis. John Wiley and Sons; 1995.
- [36] Cook RD, Malkus DS, Plesha ME, Witt RJ. Concepts and applications of finite element analysis. Fourth ed. John Wiley and Sons; 2002.
- [37] Zienkiewicz OC, Taylor RL. The finite element method for solid and structural mechanics. Elsevier; 2005.
- [38] Kirsch C. Die Theorie der Elastizität und die Bedürfnisse der Festigkeitslehre. Zeitschrift des Vereines Deutscher Ingenieure 1898;42:797–807.
- [39] Billington DP. Thin shell concrete structures. McGraw-Hill New York; 1965.
- [40] Márkus G. Theorie und Berechnung rotationssymmetrischer Bauwerke. Werner-Verlag; 1967.
- [41] Frey F, Studer MA. Analyse des structures et milieux continus: coques. Presses Polytechniques et Universitaires Roman-des, Lausanne; 2000.
- [42] Blum H. Einspannungsverhältnisse bei Bohlwerken und deren vereinfachte Berechnung mit Hilfe ideeller und stellvertre-der Belastung. PhD thesis, Technische Universität Braunschweig; 1930.
- [43] Simpson B, Powrie W. Embedded retaining walls: theory, practice and understanding. In: Proceedings of the Interna-tional Conference On Soil Mechanics and Geotechnical Engineering, vol. 4 Balkema; 2002. p. 2505–2526.
- [44] Simoen E, Moaveni B, Conte JP, Lombaert G. Uncertainty quantification in the assessment of progressive damage in a seven-story full-scale building slice. Journal of Engineering Mechanics 2013;139(12):1818–1830. [http://dx.doi.org/10.1061/\(ASCE\)EM.1943-7889.0000610](http://dx.doi.org/10.1061/(ASCE)EM.1943-7889.0000610).
- [45] Schevenels M, Pizzi M, Van Mellaert R. Optimal placement of braces in steel frames using deterministic (non-evolutionary) optimization methods. In: Proceedings of IASS Annual Symposia, vol. 24 International Association for Shell and Spatial Structures (IASS); 2015. p. 1–12.
- [46] Van Mellaert R, Mela K, Tiainen T, Heinisuo M, Lombaert G, Schevenels M. Mixed-integer linear programming approach for global discrete sizing optimization of frame structures. Structural and Multidisciplinary Optimization 2018;57(2):579–593.

- [47] François S, Galvín P, Museros P, Lombaert G, Degrande G. Dynamic soil–structure interaction analysis of a telescope at the Javalambre Astrophysical Observatory. *Soil Dynamics and Earthquake Engineering* 2014;65:165–180.
- [48] Papadopoulos M, Van Beeumen R, François S, Degrande G, Lombaert G. Modal characteristics of structures considering dynamic soil-structure interaction effects. *Soil Dynamics and Earthquake Engineering* 2018;105:114–118.
- [49] Smet J, Huybrechts N, Van Lysebetten G, Verstraelen J, François S. Optical Fiber Strain Measurements and Numerical Modeling of Load Tests on Grouted Anchors. *Journal of Geotechnical and Geoenvironmental Engineering, Proceedings of the ASCE* 2019;145(12):1–11.
- [50] Ulloa J, Wambacq J, Alessi R, Degrande G, François S. Phase-field modeling of fatigue coupled to cyclic plasticity in an energetic formulation. *Computer Methods in Applied Mechanics and Engineering* 2021;373:113473.
- [51] Wambacq J, Ulloa J, Lombaert G, François S. Interior-point methods for the phase-field approach to brittle and ductile fracture. *Computer Methods in Applied Mechanics and Engineering* 2021;375:113612.