

Optimal collapse simulator for three-dimensional structures

Ferran Vidal Codina

E.T.S. Enginyers de Camins, UPC Barcelona Tech

C/ Jordi Girona 1, 08034 BARCELONA

February 16, 2011

Abstract

In this project limit analysis for 3D structures is studied. The goal is to obtain for a certain structure the load factor λ that applied to the external loads induces collapse to the structure. The static theorem of limit analysis is the theoretical basis for the Structural Collapse Simulator (SCS), that is finding a stress distribution in equilibrium that does not violate yield criteria anywhere. This theorem is employed combined with linear programming techniques. Thereby a tutorial on LP problems is presented first. Then a brief summary of the progresses in study of limit analysis for structures is offered, being a useful introduction for understanding the very nature of SCS functioning. Moreover, limit analysis is developed and written as a LP problem, which consists of the maximization of the collapse load factor subject to equilibrium and yield criteria.

Two major contributions are presented for finding the collapse load. Firstly, the yield curve of standard 2D beam cross sections is adaptively approximated with inscribed and circumscribed polygons that yield to lower and upper bounds of λ respectively. Secondly, an interesting approach for accounting with uniform distributed loads is shown, producing bounding of the load factor. Combining these two techniques the bound gap can be reduced arbitrarily, observing convergence of the upper and the lower bounds to the exact load factor. A tutorial for using SCS and computing structures is provided, and numerical examples are thoroughly studied in order to illustrate the functioning of the program and the limits of the method. Finally, recent developments and future branches of research are detailed in order to widen the applicability range of SCS, the most important being the adaptive approximation of the yield surface for 3D beams.

Contents

1	Motivation and acknowledgements	7
2	Optimization and LP, Duality and Lagrange Multipliers	8
2.1	Optimization Problems	8
2.2	Lagrange dual function and duality	9
2.2.1	Lagrangian	9
2.2.2	Lagrange dual function	9
2.2.3	Lower bounds on optimal value	9
2.2.4	The Lagrange dual problem	10
2.2.5	The weak duality	10
2.2.6	The strong duality	11
2.3	Linear optimization problems	11
2.3.1	LP in standard form, its dual and extended dual theorem . . .	11
2.3.2	LP in general form, lagrangian and duality	12
3	Evolution in limit analysis techniques	13
4	Limit state analysis as a LP problem	17
4.1	Equilibrium	18
4.1.1	Local equilibrium	19
4.1.2	Rotation	19
4.1.3	The 2D Structure	20
4.1.4	Equilibrium for trusses	21
4.1.5	Global equilibrium	21
4.2	Yield criteria	22
4.2.1	Class type	23
4.2.2	Yield function definition	24
4.2.3	Yield curve linearization	25
4.2.4	Yield criteria as a LP problem	27
4.2.5	Yield criteria for trusses	30
4.3	Inclusion of Uniform Distributed Loads	31
4.3.1	Combination of yield and UDL bounding	34
4.4	The LP problem	34
4.4.1	Upper Bound	35
4.4.2	Lower Bound	36
4.4.3	Problem normalization	37
4.4.4	Evaluation of the bound gap	37
5	Use of Structural Collapse Simulator	38
5.1	Data input	39
5.1.1	Logicals for structure characterization	39
5.1.2	Nodal and Connectivity matrices	39

5.1.3	External Load matrices	40
5.1.4	Geometric Restrictions matrix	40
5.1.5	Example of data input	41
5.2	Data processing	43
5.2.1	Initialization of variables	43
5.2.2	Setting approximation for yield criteria	43
5.2.3	Dimensional check	44
5.3	Creation of matrices	44
5.3.1	Equilibrium matrices	45
5.3.2	Yield criteria matrices	45
5.4	LP Solver	47
5.5	Postprocessing	48
5.5.1	Gap Evaluation	48
5.5.2	Plotting plastic hinges and displacement rates	48
5.5.3	Plotting elements failure	49
5.5.4	Plotting bending moment diagram	49
5.5.5	Time of computation	50
5.5.6	Example of postprocessing	50
5.6	Cross Section Library	51
5.6.1	I/H-shaped	52
5.6.2	C4 Sections	53
5.7	Utility of SCS	53
5.7.1	Linear analysis	53
5.7.2	Nonlinear analysis	54
5.7.3	SCS	54
6	Numerical examples	54
6.1	Simply supported beam	54
6.1.1	Simply supported beam with no subdivision	56
6.1.2	Simply supported beam with one subdivision	57
6.1.3	Simply supported beam with two subdivisions	59
6.1.4	Simply supported beam with three subdivisions	59
6.2	Embedded frame with point load	62
6.3	Steel tower	64
6.4	Model of a 2D three-storey building	65
6.4.1	Frame with no span subdivision	66
6.4.2	Frame with one span subdivision	67
6.4.3	Frame with two span subdivisions	68
6.4.4	Frame with manual subdivisions	69
6.4.5	Study of convergence	70
6.5	Model of a 3D two-storey building	73
7	Conclusions	74

8	Future research	77
8.1	3D Frames yield criteria	77
8.2	Cross Section Library	78
8.3	LP solver	78
8.4	Adaptivity	78

List of Figures

1	Generalized stresses in a 3D beam element	18
2	Nodal forces and moments in a beam element	18
3	I/H-shaped symmetric cross section	24
4	Yield curve for IPE550 (red) and HEB300 (green) cross sections	25
5	Inscribed and circumscribed polygons of a IPE550 yield curve using 8/10 lines per node respectively	26
6	Simplest linearization for a general yield surface	28
7	Moment diagram with secant and tangent approximation for a beam with UDL	32
8	Schematic representation for a basic triangular structure	41
9	SCS output of <i>PlotStructures</i> function for the triangular frame	42
10	SCS output of <i>PlotHinges</i> function for the triangular frame	50
11	SCS output of <i>PlotMoments</i> function for the triangular frame	51
12	Simply supported beam of length l : (a) Loading and reactions, (b) Bending moment, (c) Shear stress	55
13	Simply supported beam with no span subdivision	56
14	Plastic hinge detection by SCS for the simply supported beam with no subdivision	57
15	Bending moment distribution for the simply supported beam with one span subdivision	58
16	Plastic hinge detection by SCS for the simply supported beam with one span subdivision	58
17	Bending moment distribution for the simply supported beam with two span subdivisions	59
18	Bending moment distribution for the simply supported beam with three span subdivisions	60
19	Convergence for λ and Δ versus the number of elements	61
20	Representation of the embedded frame with point load	61
21	SCS detection of the plastic hinges to form the collapse mechanism	62
22	Schematic representation and 3D view of the steel tower	63
23	Collapse mechanism. In red elements that fail for traction and in blue for compression	64
24	Amplified deformed structure in blue.	64
25	Representation of the three-storey frame	65
26	Convergence of λ and Δ versus the number of lines for the frame with no span subdivision	66
27	Convergence of λ and Δ versus the number of lines for the frame with one span subdivision	67
28	Convergence of λ and Δ versus the number of lines for the frame with two span subdivisions	68
29	SCS output for moment distribution for the frame with two span subdivisions	69

30	Convergence of λ and Δ versus the number of lines for the frame with an extra manual span subdivisions	70
31	Plastic hinge detection by SCS for the frame with an extra manual subdivision	71
32	Convergence of λ and Δ versus the number of lines for the frame . . .	72
33	Representation of the 3D two-storey frame	73
34	Convergence of λ and Δ versus the number of subdivisions	74
35	Plastic hinge detection by SCS for the 3D two-storey frame with three subdivisions	75
36	Yield surface for IPE360 cross section	77

List of Tables

1	Variable assignation depending on dimension and structure's nature .	43
2	Results from SCS computations of the simply supported beam	60
3	Results from SCS computations for the frame with no span subdivision	67
4	Results from SCS computations for the frame with one span subdivision	67
5	Results from SCS computations for the frame with two span subdivisions	68
6	Results from SCS computations for the frame with an extra manual subdivision	70
7	Results from SCS computations of the 3D frame	73

1 Motivation and acknowledgements

The present document consists of the collection of the work I have been developing for over a year in the project CETICA. This project is focused in developing an optimal collapse simulator for structures, a theme that combines applied mathematics along with structures, two of the disciplines I have been more interested in since I entered ETSECCPB and when I later joined CFIS. Moreover, the main researchers of the project (Huerta, Peraire, Bonet) are themselves an excellent reason for accepting this challenge.

This project collects the theoretical approach to the problem and the solution that so far has been implemented and achieved satisfying results. CETICA project has enabled me to improve programming skills in Matlab[®], since my research colleagues and I have designed the software almost entirely, by implementing the theoretical model later described. Thereby, I would like to give my sincere thanks to Josep Sarrate Ramos and Joel Saà Seoane, for collaborating closely in the development of the software, as without them it would have been impossible to accomplish this task. Although CETICA project is not finished, as it is explained in the project the basic programming tasks are already done, and the future research paths have already been set.

Finally, I would like to mention enthusiastically my tutor, Antonio Huerta Cerezuela, for giving me the opportunity of joining this program and collaborating with CETICA. This research project has encouraged me to pursue further training in computational engineering, so the final balance could not have been more positive.

2 Optimization and LP, Duality and Lagrange Multipliers

2.1 Optimization Problems

We define an optimization problem in the standard form as follows

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && g_i(\mathbf{x}) \leq 0, i = 1, \dots, m \\ & && h_i(\mathbf{x}) = 0, i = 1, \dots, p \end{aligned} \tag{1}$$

This approach states to finding the value \mathbf{x} that minimizes $f(\mathbf{x})$ among all \mathbf{x} that satisfy the restrictions $g_i(\mathbf{x})$ and $h_i(\mathbf{x})$. The variable $\mathbf{x} \in \mathbb{R}^n$ is called the optimization variable, whereas $f(\mathbf{x})$ is the objective function of the optimization problem. The inequality constraint functions g_i (respectively equality constraint functions h_i) are all of them continuously differentiable functions from $\mathbb{R}^n \rightarrow \mathbb{R}$. They constitute the inequality constraints $g_i(\mathbf{x}) \leq 0$ (equality constraints $h_i(\mathbf{x}) = 0$). This problem, for reasons that will be soon seen, is often called the primal problem.

The domain of the optimization problem is

$$D = \text{dom } f(\mathbf{x}) \cap \bigcap_{i=1}^m \text{dom } g_i(\mathbf{x}) \cap \bigcap_{i=1}^p \text{dom } h_i(\mathbf{x})$$

A point \mathbf{x} in D is feasible if it satisfies all the constraints $g_i(\mathbf{x})$, $h_i(\mathbf{x})$. A problem is said to be feasible if it exists at least one feasible point, i.e., the feasibility region defined as

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) \leq 0, i = 1, \dots, m; h_i(\mathbf{x}) = 0, i = 1, \dots, p\}$$

is nonempty. The optimal value b of the above defined optimization problem can be defined as follows

$$p^* = \inf\{f(\mathbf{x}) \mid g_i(\mathbf{x}) \leq 0, i = 1, \dots, m; h_i(\mathbf{x}) = 0, i = 1, \dots, p\}$$

This optimal value is allowed to take the extended values $\pm\infty$. When b takes the value ∞ , we say the problem to be infeasible (feasibility region is an empty set). Otherwise, if there exists points \mathbf{x}_k which $f(\mathbf{x}_k) \rightarrow -\infty$ as $\mathbf{x}_k \rightarrow \infty$ then $b = -\infty$ and the problem is unbounded below.

2.2 Lagrange dual function and duality

2.2.1 Lagrangian

Considering an optimization problem as (1), we can take into consideration the constraints by increasing the objective function with a weighted sum of the constraint functions. The Lagrangian associated with the problem (1) is a functional $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ that can be defined as

$$L(\mathbf{x}, \boldsymbol{\eta}, \boldsymbol{\nu}) = f(\mathbf{x}) + \sum_{i=1}^m \boldsymbol{\eta}_i g_i(\mathbf{x}) + \sum_{i=1}^p \boldsymbol{\nu}_i h_i(\mathbf{x})$$

With $\text{dom } L : D \times \mathbb{R}^m \times \mathbb{R}^p$. We refer to $\boldsymbol{\eta}_i$ as the Lagrange multiplier associated with the i th inequality constraint, and similarly $\boldsymbol{\nu}_i$ is the Lagrange multiplier associated to the i th equality constraint. The vectors $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$ are called the Lagrange multipliers vectors or the dual variables associated with (1).

2.2.2 Lagrange dual function

The dual function, or Lagrangian dual function $G : \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$, can be defined as the minimum value for the Lagrangian over \mathbf{x}

$$G(\boldsymbol{\eta}, \boldsymbol{\nu}) = \inf_{\mathbf{x} \in D} L(\mathbf{x}, \boldsymbol{\eta}, \boldsymbol{\nu}) = \inf_{\mathbf{x} \in D} \left(f(\mathbf{x}) + \sum_{i=1}^m \boldsymbol{\eta}_i g_i(\mathbf{x}) + \sum_{i=1}^p \boldsymbol{\nu}_i h_i(\mathbf{x}) \right)$$

If the Lagrangian is unbounded below, the dual function takes on the value $-\infty$. For being the pointwise minimum of a family of affine functions $(\boldsymbol{\eta}, \boldsymbol{\nu})$, the Lagrange dual function is concave even if the primal problem is nonconvex¹.

2.2.3 Lower bounds on optimal value

It is important to note that the dual function by definition yields lower bounds to the optimal value b for the primal problem. Taking a pair $(\boldsymbol{\eta}, \boldsymbol{\nu})$, where $\boldsymbol{\eta} \geq 0$, is easy to proof that $G(\boldsymbol{\eta}, \boldsymbol{\nu}) \leq b$.

PROOF: Take $\hat{\mathbf{x}}$ in \mathcal{S} (feasibility region). According to the choice of $(\boldsymbol{\eta}, \boldsymbol{\nu})$, it verifies that

$$\sum_{i=1}^m \boldsymbol{\eta}_i g_i(\hat{\mathbf{x}}) + \sum_{i=1}^p \boldsymbol{\nu}_i h_i(\hat{\mathbf{x}}) \leq 0$$

¹A convex problem is one in which the objective and the constraint functions are convex, i.e., they satisfy $f(\alpha \mathbf{x} + \beta \mathbf{y}) \leq \alpha f(\mathbf{x}) + \beta f(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and all $\alpha + \beta = 1 \in \mathbb{R}$.

as $g_i(\hat{\mathbf{x}}) \leq 0$ and $h_i(\hat{\mathbf{x}}) = 0$, whereby every term of the first sum is nonpositive and every term of the second is equal to zero.

Thus, evaluating the Lagrangian on $\hat{\mathbf{x}}$, we obtain $L(\hat{\mathbf{x}}, \boldsymbol{\eta}, \boldsymbol{\nu}) \leq f(\hat{\mathbf{x}})$. Automatically, we have $G(\boldsymbol{\eta}, \boldsymbol{\nu}) = \inf L(\mathbf{x}, \boldsymbol{\eta}, \boldsymbol{\nu}) \leq L(\hat{\mathbf{x}}, \boldsymbol{\eta}, \boldsymbol{\nu}) \leq f(\hat{\mathbf{x}})$. Since this inequality holds for any feasible point, the proof is complete.

□

Hence, the following inequality verifies

$$G(\boldsymbol{\eta}, \boldsymbol{\nu}) \leq \inf_{\substack{g(\mathbf{x}) \leq 0 \\ h(\mathbf{x}) = 0}} f(\mathbf{x}) \quad (2)$$

Nevertheless, it should be taken into account that if $G(\boldsymbol{\eta}, \boldsymbol{\nu}) = -\infty$ the lower bound is trivial. Hence, for a nontrivial lower bound for b need to be $\boldsymbol{\eta} \geq 0$. If the pair $(\boldsymbol{\eta}, \boldsymbol{\nu})$ attain to this condition, they belong to the dual feasibility region \mathcal{T} , which contains the pairs $(\boldsymbol{\eta}, \boldsymbol{\nu})$ such that the dual function is not unbounded below.

2.2.4 The Lagrange dual problem

As stated before, for each feasible pair $(\boldsymbol{\eta}, \boldsymbol{\nu})$ the Lagrange dual function yields a lower bound of the optimal value b of the primal problem, it is natural to seek the maximum lower bound via the optimization problem

$$\begin{aligned} & \text{maximize} && G(\boldsymbol{\eta}, \boldsymbol{\nu}) \\ & \text{subject to} && \boldsymbol{\eta} \geq 0 \end{aligned} \quad (3)$$

which is the so-called Lagrange dual problem or simply the dual problem, associated to the (1). The pair $(\boldsymbol{\eta}, \boldsymbol{\nu})$ belongs to the dual feasibility region \mathcal{T} if it is feasible for the dual problem. Moreover, it is common to refer to $(\hat{\boldsymbol{\eta}}, \hat{\boldsymbol{\nu}})$ as the optimal Lagrange multipliers or the dual optimal if they are optimal for (3). One last thing that is worth mentioning is the fact that the dual problem is always convex, due to the concavity of the objective function and the convexity of the constraint, no matter the convexity of the primal problem.

2.2.5 The weak duality

The optimal value of the Lagrange dual problem, $\sup G(\boldsymbol{\eta}, \boldsymbol{\nu})$, which will be denoted as d^* , is the best lower bound for the primal problem that can be obtained. Henceforth we have the following inequality

$$\sup_{\boldsymbol{\eta} \geq 0} G(\boldsymbol{\eta}, \boldsymbol{\nu}) \leq \inf_{\substack{g(\mathbf{x}) \leq 0 \\ h(\mathbf{x}) = 0}} f(\mathbf{x})$$

which stands even if the problem is not convex. This property is known as the weak duality. The weak duality holds even in case one of the inequality members is infinite. For example, if the dual problem is unbounded above ($\sup G(\boldsymbol{\eta}, \boldsymbol{\nu}) = \infty$), the primal problem is infeasible, i.e. $p^* = \infty$. Otherwise, if the primal is unbounded below ($p^* = -\infty$), the dual must be infeasible ($\sup G(\boldsymbol{\eta}, \boldsymbol{\nu}) = -\infty$).

We refer to the difference between both quantities, $p^* - d^*$, as the optimal duality gap, which is the gap between the optimal value of the primal problem and the greatest lower bound, and it is always nonnegative.

The weak duality can be used to find lower bounds of the primal problem, and it is useful in cases where the primal is difficult to solve, as the dual is always convex.

2.2.6 The strong duality

If the optimal duality gap is zero, the strong duality holds. If the primal problem is convex, then the strong duality usually (but not always) holds. A primal convex problem is as follows

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && g_i(\mathbf{x}) \leq 0, i = 1, \dots, m \\ & && \mathbf{Ax} = \mathbf{b} \end{aligned}$$

Where $f(\mathbf{x})$ and $g_i(\mathbf{x})$ are convex. Many results can be found about establishing conditions under which strong duality holds, but there will be no further discussed in this project.

2.3 Linear optimization problems

2.3.1 LP in standard form, its dual and extended dual theorem

When the objective and the constraint functions in our optimization problem are linear², the problem is called a Linear Program (LP). A standard LP problem can be written as

$$\begin{aligned} & \text{minimize} && \mathbf{c}^T \mathbf{x} \\ & \text{subject to} && \mathbf{Ax} = \mathbf{b} \\ & && \mathbf{x} \geq 0 \end{aligned}$$

with the only equalities as the component-wise nonnegativity constraints. Linear programs are, of course, convex optimization problems. Since it is possible to maximize an affine objective function $\mathbf{c}^T \mathbf{x}$ by minimizing $-\mathbf{c}^T \mathbf{x}$ (convex as well),

²A linear function is one that satisfies $f(\alpha \mathbf{x} + \beta \mathbf{y}) = \alpha f(\mathbf{x}) + \beta f(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and all $\alpha, \beta \in \mathbb{R}$.

2 OPTIMIZATION AND LP, DUALITY AND LAGRANGE MULTIPLIERS

it is also referred as a LP a maximization problem with linear objective and constraint functions. It should be noted that in this case the feasible set is a generalized polyhedron \mathcal{P} , and the problem is to minimize the affine function $\mathbf{c}^T \mathbf{x}$ over \mathcal{P} . The problem dual to this latter is

$$\begin{aligned} & \text{maximize} && \mathbf{u}^T \mathbf{b} \\ & \text{subject to} && \mathbf{u}^T \mathbf{A} \leq \mathbf{c}^T \end{aligned}$$

With this definitions it is immediate that

LEMMA 2.1 *If $\mathbf{Ax} = \mathbf{b}$, $\mathbf{u}^T \mathbf{A} \leq \mathbf{c}^T$ then*

$$\mathbf{u}^T \mathbf{b} = \mathbf{u}^T \mathbf{Ax} \leq \mathbf{c}^T \mathbf{x}$$

Finally is announced the dual theorem, first conjectured by J. von Neumann and proved afterwards by Gale, Kuhn and Tucker, of LP.

Theorem 2.1 (Extended dual theorem) *For any dual pair of problems, precisely one of the following occurs:*

1. *There exists \mathbf{x}^* , \mathbf{u}^* with $\mathbf{Ax}^* = \mathbf{b}$, $\mathbf{x}^* \geq 0$ and $\mathbf{u}^{*T} \mathbf{A} \leq \mathbf{c}^T$ such that $\mathbf{u}^{*T} = \mathbf{c}^T \mathbf{x}^*$*
2. *$\mathbf{Ax} = \mathbf{b}$, $\mathbf{x} \geq 0$ has a solution, $\mathbf{u}^T \mathbf{A} \leq \mathbf{c}^T$ has none and $\min \mathbf{c}^T \mathbf{x} = -\infty$*
3. *$\mathbf{Ax} = \mathbf{b}$, $\mathbf{x} \geq 0$ has no solution, $\mathbf{u}^T \mathbf{A} \leq \mathbf{c}^T$ has one and $\max \mathbf{u}^T \mathbf{b} = \infty$*
4. *Neither $\mathbf{Ax} = \mathbf{b}$, $\mathbf{x} \geq 0$ nor $\mathbf{u}^T \mathbf{A} \leq \mathbf{c}^T$ have a solution*

COROLLARY 2.1 *If $\mathbf{Ax} = \mathbf{b}$, $\mathbf{x} \geq 0$ has a solution and $\mathbf{c}^T \mathbf{x}$ is bounded below (or alternatively if $\mathbf{u}^T \mathbf{A} \leq \mathbf{c}^T$ has a solution and $\mathbf{u}^T \mathbf{b}$ is bounded above) then there exists optimal solutions \mathbf{x}^* , \mathbf{u}^* to both problems above.*

2.3.2 LP in general form, lagrangian and duality

A general linear program can be expressed as

$$\begin{aligned} & \text{minimize} && \mathbf{c}^T \mathbf{x} \\ & \text{subject to} && \mathbf{Dx} \leq \mathbf{e} \\ & && \mathbf{Ax} = \mathbf{b} \end{aligned} \tag{4}$$

And the variables $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n$, $\mathbf{e} \in \mathbb{R}^m$, $\mathbf{b} \in \mathbb{R}^p$, $\mathbf{D} \in \mathbb{R}^{m \times n}$ and $\mathbf{A} \in \mathbb{R}^{p \times n}$. The Lagrangian $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ associated to (4) can be defined as

$$L(\mathbf{x}, \boldsymbol{\eta}, \boldsymbol{\nu}) = \mathbf{c}^T \mathbf{x} + \boldsymbol{\eta}^T (\mathbf{Dx} - \mathbf{e}) + \boldsymbol{\nu}^T (\mathbf{Ax} - \mathbf{b}) = -\mathbf{e}^T \boldsymbol{\eta} - \mathbf{b}^T \boldsymbol{\nu} + \mathbf{x}^T (\mathbf{D}^T \boldsymbol{\eta} + \mathbf{A}^T \boldsymbol{\nu} + \mathbf{c})$$

so the dual function is

$$G(\boldsymbol{\eta}, \boldsymbol{\nu}) = \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\eta}, \boldsymbol{\nu}) = -\mathbf{e}^T \boldsymbol{\eta} - \mathbf{b}^T \boldsymbol{\nu} + \inf_{\mathbf{x}} (\mathbf{x}^T (\mathbf{D}^T \boldsymbol{\eta} + \mathbf{A}^T \boldsymbol{\nu} + \mathbf{c}))$$

The infimum of a linear function is $-\infty$, except in the case where is identically zero, so the dual function can be expressed as

$$G(\boldsymbol{\eta}, \boldsymbol{\nu}) = \begin{cases} -\mathbf{e}^T \boldsymbol{\eta} - \mathbf{b}^T \boldsymbol{\nu} & \text{if } \mathbf{D}^T \boldsymbol{\eta} + \mathbf{A}^T \boldsymbol{\nu} + \mathbf{c} = 0 \\ -\infty & \text{otherwise} \end{cases}$$

The dual variable $\boldsymbol{\eta}$ is dual feasible if $\boldsymbol{\eta} \geq 0$ and $\mathbf{D}^T \boldsymbol{\eta} + \mathbf{A}^T \boldsymbol{\nu} + \mathbf{c} = 0$, hence the lower bound property (2) holds, and the lower bound to the optimal value happens to be $-\mathbf{e}^T \boldsymbol{\eta} - \mathbf{b}^T \boldsymbol{\nu}$ (4). The Lagrange dual problem defined by (3) can be reformulated by including the dual feasibility conditions as explicit constraints

$$\begin{aligned} & \text{maximize} && -\mathbf{e}^T \boldsymbol{\eta} - \mathbf{b}^T \boldsymbol{\nu} \\ & \text{subject to} && \mathbf{D}^T \boldsymbol{\eta} + \mathbf{A}^T \boldsymbol{\nu} + \mathbf{c} = 0 \\ & && \boldsymbol{\eta} \geq 0 \end{aligned} \tag{5}$$

The problem that will be dealt throughout this project responds to problems (4) and (5). Nevertheless, many different forms of LP problems can be used depending on the LP solver chosen (`linsolve`, `SDPT3`, `CVX`,...) and the very nature of the problem itself. Further information about LP problems approaches can be found in [1].

3 Evolution in limit analysis techniques

Limit analysis has been an increasingly and widely used tool for structure designing and soil mechanics analysis since its initial developments in the 19th century. The problem aimed to be solved by means of limit analysis consists of finding the minimum multiple of the load distribution in a solid subject that drives to the complete collapse of the body, assuming a plastic behavior of the subject, i.e. elastic range is left. In this project it will be discussed the limit analysis applied to finding conditions of failure of statically loaded 3D and 2D-structures of ductile materials, particularly steel, and the process of loading will be proportional. Continuous beams and frames of steel can carry loads considerably greater than the ones which cause to reach the elastic limit of the material. In general, when loading increases plastic yield is attained in some elements of the structure, which implies the partial loss of its bearing capacities. If the process of loading does not cease it may incur the physical failure of the structure, when the load has reached a certain value called collapse load. Above this factor, small loading increases may result in much larger permanent deformations than the ones experienced before. The so-called plastic methods attempt to estimate the collapse load factor, and hence provide both a

knowledge of its bearing capacity and a better use of materials in the design process.

Plastic analysis is based on the idealization of the stress-strain surface as elastic-perfectly-plastic. The relation between the bending moment and the curvature at each member of the structure is the starting point for the limit analysis. The basic hypothesis is that if the bending moment of an initially unstressed and unstrained element under pure bending approaches a value noted by M_p , which depends on the nature and characteristics of the material, the curvature of the element increases indefinitely. This value is commonly referred to as the plastic moment of the member. The formation of a plastic hinge is closely related to the attainment of the plastic moment in some sections. The concept of plastic hinge, which is key in limit analysis, was first proposed according to Maier-Leibnitz [2] by G.V.Kazinczy in the Hungarian journal *Betonszemle* in 1914. Theoretically, if the plastic moment is reached in a section of a member it would lead to infinite curvature, hence this section would be able to change the slope angle in infinitesimal distances. Hence the members would behave as if they were attached to a hinge which transmits only a constant moment equal to $\pm M_p$.

Progresses in the development of efficient plastic methods for limit analysis calculation were made by Greenberg and Prager [3], and the most relevant ones that should be pointed out are the static and kinematic theorem. Let P_c be the actual collapse load of a given frame, accepting that all the loads can be combined with certain ratios so that can be expressed as a single quantity. Admit P_{lb} as a load at which it is possible to find a feasible system of bending moments satisfying equilibrium equations and plastic moment is not reached at any section. Under these circumstances, Greenberg and Prager showed that $P_{lb} \leq P_c$.

This kinematic method is based on the combination of elementary mechanisms and the virtual work principle. The plastic collapse loads correspond to several failure mechanisms, and are found by equating the internal work at the plastic hinges with the external work performed by the loads during the virtual displacement. Rotations and deflections must thereby be computed. One of the hypotheses assumed is that the frame remains rigid between the supports and the hinges, thus plastic rotation only occurs at plastic hinges. Combining mechanisms various collapse loads can be found, and the kinematic theorem states that the minimum load factor is an upper bound of the real collapse load. Its immediate consequence is that $P_c \leq P_{ub}$, where P_{ub} is the minimum load found combining mechanisms. This method constitutes a useful tool for having an reference value of the load necessary for a structure to lose its stability and collapse, but it always yields to unsafe values. Thus, considering both principles the following inequality can be obtained

$$P_{lb} \leq P_c \leq P_{ub}$$

Which expresses that there one and only one collapse load: there exists a unique load value such that, with all bending moments satisfying statical equilibrium and

nowhere exceeding a plastic moment magnitude, plastic hinges exist at sufficient sections to convert the frame into a kinematic mechanism. This load can only be found explicitly if the results of both static and kinematic principles coincide. The methodology of collapse load calculation based on these theorems was developed and widespread by Symonds and Neal [4, 5].

Symonds and Neal [6] were also relevant for their studies about computation of plastic moments of various cross sections and techniques for its computation in general cases. Moreover, observations were made regarding the influence of shear stress and axial forces on the variation of the plastic moment, and together with Chwalla [7], Baker, Horne and Roderick [8] further investigations were carried out concerning conditions under which these forces should be taken into account.

Paralelly to the demonstration of these two theorems in 1951, which revealed to be crucial in the development and study of limit analysis, it gained relevance the approach to the problem of finding the exact collapse load as a linear programming problem. It had been studied previously, but it always appeared the obstacle of a physical interpretation of the variables intervening in the dual problem. Charnes and Greenberg [9] established the equivalence, for trussed structures, of dual programming problems and the static and kinematic theorems. This equivalence regarded the stress in a bar as a primal variable and the displacement of a joint as a dual variable. Henceforth, these identification allowed the study of trussed structures using the theory and the computational advantages of linear programming.

The conjectured equivalence for frames, the most important structural application, was not proved until 1959 by Charnes, Lemke and Zienkiewicz [10] using virtual work and geometric compatibility. In this article the authors developed a parametric form for the static equations of equilibrium combined with the compatibility conditions, as well as giving a physical interpretation of the dual variables. Using these results, the LP problem was first formulated, being the primal the maximization of the load factor subject to the static equilibrium equations (static theorem), and its dual the minimization of the load factor subject to the conditions of constituting a kinematic mechanism attaining the plastic moment at the critical sections. It follows by the application of the extended dual theorem (2.1) that there is a finite optimum for either the primal or the dual, then there must be one for the other. As a consequence of this duality and the application of the extended dual theorem conditions for the existence of a collapse mechanism were established.

Once the LP problem was formulated, further progresses were made in this field, aiming to improve the approach to the actual problem and the solving techniques. There should be cited Heyman [11], Horne [12], Baker and Heyman [13] and Watwood [14]. In 1972, Anderheggen and Knopfel [15] first introduced in the problem of finding the collapse load factor the necessity of a combined yield condition. Beforehand it was only taken into account the influence of the bending moment as a

condition for attaining plastic flow with the aforementioned plastic hinges. It had been noted the negative influence of axial and shear forces on the fully plastic moment, but no mathematical approach had been made. Anderheggen and Knopfel introduced the yield condition as a feasible stress domain in the bending moment - axial force plane (in two dimensions) that should not be violated at any point. As it will be seen throughout this project, yield surfaces are in general nonlinear convex sets, therefore a process of linearization of the yield conditions was proposed. By means of this process the restriction of not exceeding the (reduced) plastic moment could be placed as a constraint in the LP primal problem. Furthermore, the finite element method was used as a mathematical tool for modeling the problem and formulating it as an LP, by assuming stress and displacement fields and then using the virtual work principle to find the coefficients corresponding to linear equilibrium and kinematic compatibility.

This latter article set the basis of finding the collapse load factor as it is stated nowadays. Other referent authors which contributed to the development of solving the LP problem were Munro [16], Livesley [17] and Maier, Giacomini and Paterlini [18]. The results achieved by the last quoted authors involve formulating the problem as a restricted basis linear program (RBLP), which implicates that in the basic LP problem there is an extra complementarity nonlinear condition, able to be solved by the standard simplex method with a similar computational cost, although it provided the deformation history of the structure.

Jennings and Tam [19] proposed in their paper a modified simplex technique based on a minimum weight criteria for the selection of relative member sizes, for structures with only flexural actions and concentrated loads, minimizing an objective function which regards the possible positions of the plastic hinges and checks equilibrium via the static theorem. Further innovations in methods of solving the problem can be attributed to Thierauf [20], who presented an iterative method for solving the limit analysis problem with alternative loads with a linear objective function and quadratic yield conditions.

Damkilde and Hoyer [21] proposed a new approach to the standard LP problem based on the reduction of degrees of freedom a priori so as to deal with a reduced problem with less equations and also the LP problem without the restriction of non-negative variables (which is actually more accurate) using slack variables. Damkilde used the linearization of the yield surface as well, thus checking yield conditions are not violated at any point. Another interesting idea stated in this paper was the possibility of the inclusion of the material optimization, so a double load and material optimization can be carried out.

From the early 90s until nowadays great progresses have been made in solving the collapse load factor problem using different computer approaches. For instance, Tin-Loi [22] used the mathematical programming language GAMS for pla-

nar frames; Kaveh and Jahanshahi [23] applied heuristic algorithms such as Ant Colony System; Corradi, Luzzi and Vena [24] developed the limit analysis problem for anisotropic structures; Tjhin and Kuchma [25] used the strut-and-tie method as an equilibrium method for the limit analysis; Van Long and Dan Hung [26] combined direct LP methods with step-by-step methods for the analysis of 3D frames; Peraire, Bonet and Ciria [27] designed a program with an adaptive mesh system for the computation of both lower and upper bounds, solving in this case second-order cone programs.

Furthermore, if the focus is on the evolution of the LP solving techniques, authors such as Borges et al.[28], Lyamin [29], and Krabbenhoft and Damkilde [30] made decisive contributions with much more efficient methods than the simplex methods, which are capable of solving both linear and nonlinear programs. In addition to that, in these methods the number of iterations is largely independent of the problem size, meaning that problems with thousands of variables can be solved within minutes.

As stated before, this project will focus on the development of limit analysis of both trussed and framed 3D and 2D structures writing it as a LP problem. Firstly global equilibrium equations will be introduced in matrix notation, bearing in mind geometric restrictions and kinematic constraints. One of the new approaches to the problem is the yield condition linearization. Yield surface of standard beam cross-sections is explicitly written, and it is adaptively approximated in a manner that every element of the structure can have its yield surface differently approximated if desired. Besides that, this approximation yields to lower and upper bounds of the exact collapse load whether the yield surface (always convex) is approximated inwards or outwards. The second major innovation is the possibility of considering uniform distributed loads, and an adaptive procedure is sought as well. Combining both the yield conditions, UDL and adaptivity the bound gap can be arbitrarily reduced, and therefore a more precise collapse load factor can be found.

4 Limit state analysis as a LP problem

Following some of the aforementioned developments and work in limit analysis, a general and complete approach to the problem of finding the exact load factor that scales the external loadings in order to induce collapse to the structure is explicitly proposed here. This load factor, which will be referenced from now on using the notation λ , is forced to be nonnegative, thus the direction of the loads is decisive. This procedure is based on the static theorem, which states that a given statically admissible stress distribution which doesn't violate yield criteria anywhere induces a lower-bound of the actual collapse load. This condition will therefore be expressed as a LP problem. Its dual problem, which happens to be the kinematic theorem, will not be sought in this project, nevertheless it would lead to the same exact result

(with a sensible stress and displacement fields and accounting the fact that the dual of the dual is again the primal problem). However, for a computer-based approach it is simpler to consider the static theorem. In this section the equilibrium equations and yield criteria will be presented, as well as its writing as linear conditions for its further use in the LP problem.

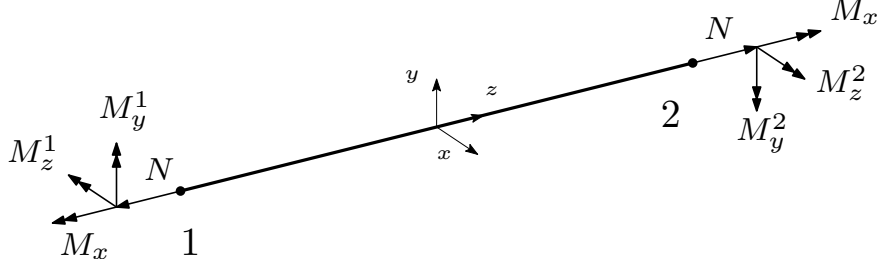


Figure 1: Generalized stresses in a 3D beam element

4.1 Equilibrium

The lower bound theorem states that equilibrium at the structure must be satisfied both at beam and global level. The first important assumption to be made is the external point loads to be applied at the nodes of the structure. Even if the desired structure has point loads applied in the middle of the span, the user is required to introduce an extra node in the exact position of the point load application, in order to account for this load. Hence, a 3D framed beam, has a constant axial force N , a constant torsional moment M_x , and linear bending moments M_z and M_y . The stress state at the local level is expressed via the *element generalized stress vector*, $\sigma_{\text{ele}}^T = (N, M_x, M_z^1, M_z^2, M_y^1, M_y^2)$, where the superscripts 1, 2 refer to the node of the element. The coordinate system chosen from now on is the one shown in Figure 1.

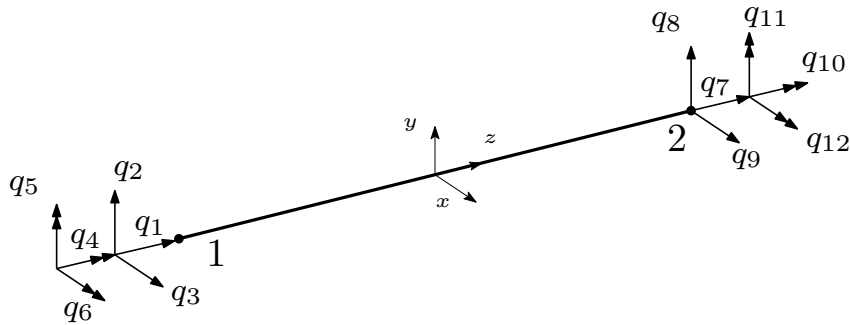


Figure 2: Nodal forces and moments in a beam element

The nodal forces and vectors on the beam, \mathbf{q} , can be determined via the generalized stress vector and the element flexibility matrix \mathbf{H}_{ele} as follows

$$\mathbf{q} = \mathbf{H}_{\text{ele}} \sigma_{\text{ele}} \tag{6}$$

The vector $\mathbf{q}^T = (q_1, \dots, q_{12})$ contains the nodal force vectors (q_1, q_2, q_3) and (q_7, q_8, q_9) at nodes 1 and 2 respectively, and the nodal moment vectors (q_4, q_5, q_6) and (q_{10}, q_{11}, q_{12}) as shown in Figure 2. Remark that $\mathbf{H}_{e1e} \in \mathbb{R}^{\dim \mathbf{q} \times \dim \sigma_{e1e}}$, and in general this is $\mathbb{R}^{12 \times 6}$.

4.1.1 Local equilibrium

Equilibrium at the beam element determines the structure of the so-called element flexibility matrix. Recall that shear stresses are determined from bending moments. If l is the length of the beam element, the element flexibility matrix can be described as:

$$\mathbf{H}_{e1e} = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1/l & 1/l & 0 & 0 \\ 0 & 0 & 0 & 0 & -1/l & 1/l \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/l & -1/l & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/l & -1/l \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (7)$$

Since this matrix reflects equilibrium at the beam, the forces are expressed in the local coordinate system aforementioned. In the assembly process of a general frame, multiple directions can be taken, whereby the flexibility matrix is needed to rotate in order to be expressed in the global coordinate system, which will be defined by the user.

4.1.2 Rotation

The rotation matrix, referred to as \mathbf{T} is a real-valued 3×3 orthogonal rotation matrix in 3D, i.e. its transpose equals its inverse and its determinant is equal to 1. Therefore $\mathbf{T}^T \mathbf{T} = \mathbf{I} = \mathbf{T} \mathbf{T}^T$, where \mathbf{I} is the identity matrix.

For the three dimensional case, geometry is not only important, but also how the beams are placed within the structure. Namely, the transformation matrix is simply the global coordinates of the local axial unit vectors, see Figure 1. In fact, the unit vector in the x -direction of the local coordinate system is given by the beam/bar geometry. Let \mathbf{n}_{e1} be the number of elements in a general frame. For an arbitrary element e , $e = 1, \dots, \mathbf{n}_{e1}$. If $\mathbf{u}_e^T = (u_x, u_y, u_z)$ denotes this unit vector for element e , then

$$\mathbf{u}_e^T = \frac{(x_2^e - x_1^e, y_2^e - y_1^e, z_2^e - z_1^e)}{\sqrt{(x_2^e - x_1^e)^2 + (y_2^e - y_1^e)^2 + (z_2^e - z_1^e)^2}}$$

where (x_i^e, y_i^e, z_i^e) denote the coordinates of node i , $i = 1, 2$, of element e .

The unit vector for element e in the y -direction of the local coordinate system, is denoted \mathbf{v}_e and is assumed to be in the web direction of the beam. The beam web direction is data that must be given a priori. Finally, the unit vector in the z -direction of the local coordinate system is denoted \mathbf{w}_e . The direction of the axis is the one depicted in Figure 1. The rotation matrix is constructed as follows

$$\mathbf{w}_e = \mathbf{u}_e \times \mathbf{v}_e \quad \mathbf{T}^e = [\mathbf{u}_e \ \mathbf{v}_e \ \mathbf{w}_e]$$

Hence, the rotation matrix for an element e is $\bar{\mathbf{T}}^e \in \mathbb{R}^{12 \times 12}$, and can be written as

$$\bar{\mathbf{T}}^e = \begin{pmatrix} \mathbf{T}^e & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^e & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{T}^e & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{T}^e \end{pmatrix}$$

and its left product with the element flexibility matrix \mathbf{H}_{e1e}^e gives the element flexibility matrix in global coordinates $\bar{\mathbf{T}}^e \mathbf{H}_{e1e}^e$.

4.1.3 The 2D Structure

For a 2D frame the element generalized stress vector becomes $\boldsymbol{\sigma}_{e1e}^T = (N, M_z^1, M_z^2)$ and the element flexibility matrix is reduced to:

$$\mathbf{H}_{e1e} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1/l & 1/l \\ 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 1/l & -1/l \\ 0 & 0 & 1 \end{pmatrix}$$

For rotating the element flexibility matrix in 2D, two vectors are defined. For an arbitrary element e , naming $\mathbf{u}_e^T = (u_1, u_2)$ the geometry vector of the beam in global coordinates, with origin at the node 1 and end at the node 2 of the element, the orthogonal vector that configures a proper axis orientation is $\mathbf{v}_e^T = (-u_2, u_1)$, and the rotation matrix is shaped as

$$\mathbf{T}^e = \begin{pmatrix} u_e^1 & -u_e^2 & 0 \\ u_e^2 & u_e^1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thus, the rotation matrix for an element e is $\bar{\mathbf{T}}^e \in \mathbb{R}^{6 \times 6}$, and can be written as

$$\bar{\mathbf{T}}^e = \begin{pmatrix} \mathbf{T}^e & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^e \end{pmatrix}$$

and the element flexibility matrix in global coordinates is defined by left multiplying with this latter matrix.

4.1.4 Equilibrium for trusses

It was regarded important to remark the simple case where the structure has hinges at every node. The elements only carry axial loads, therefore $\sigma_{\text{ele}} = N \in \mathbb{R}$ and the element flexibility matrix is $\mathbf{H}_{\text{ele}}^T = (-1, 0, 1, 0)$ in 2D and $\mathbf{H}_{\text{ele}}^T = (-1, 0, 0, 1, 0, 0)$ in 3D.

4.1.5 Global equilibrium

Once local equilibrium is clarified, the next step is equilibrium for the whole structure. Nodal equilibrium for the global frame is determined after summation of the element contributions at the nodes, the so-called \mathbf{p} vectors in (6), and the external loads. The several element flexibility matrices defined in (7) must be assembled in the global flexibility matrix \mathbf{H} , and accounting the external loads and the reaction forces due to kinematic constraints, global equilibrium can be expressed as follows

$$\mathbf{H}\boldsymbol{\sigma} = \mathbf{G}\mathbf{r} + \lambda\mathbf{f} \quad (8)$$

where \mathbf{f} is the vector containing all the external nodal forces³, λ is the plastic load multiplier (the aim of the problem), \mathbf{r} is the vector containing the reactions due to the boundary conditions and kinematic constraints and \mathbf{G} is a matrix that relates each reaction force to the corresponding geometric restriction (embedded, simply supported,...). $\boldsymbol{\sigma}$ is the global generalized stress vector, which contains all the element stress vector assembled, therefore

$$\boldsymbol{\sigma} = (\boldsymbol{\sigma}_{\text{ele}}^1{}^T, \dots, \boldsymbol{\sigma}_{\text{ele}}^{n_{\text{el}}}{}^T)$$

This vector belongs to $\mathbb{R}^{6n_{\text{el}}}$ but need not to have this exact dimension since static constraints such as internal hinges can be present in the structure, and the stress parameters can be known *a priori* and the dimension of $\boldsymbol{\sigma}$ must be reduced accordingly. The number of columns of \mathbf{H} is equal to the dimension of $\boldsymbol{\sigma}$; thus in general is six times the number of elements, $6n_{\text{el}}$. The number of rows corresponds to nodal equilibrium for each degree of freedom. In general it is six times the number of nodal points, $6n_{\text{np}}$. The structure of the global flexibility matrix can be written, where rows represent nodes and columns represent elements, as follows

³UDL inclusion will be later discussed.

$$\mathbf{H} = \begin{matrix} & 1 & \dots & d & \dots & e & \dots & n_{e1} \\ \begin{matrix} 1 \\ \vdots \\ i \\ \vdots \\ j \\ \vdots \\ k \\ \vdots \\ n_{np} \end{matrix} & \left(\begin{array}{cccccc} & & & \mathbf{0} & & \mathbf{0} \\ & & & \vdots & & \vdots \\ & & & \mathbf{0} & & \mathbf{0} \\ & & \bar{\mathbf{T}}^d \mathbf{H}_{e1e,i}^d & & & \mathbf{0} \\ & & \mathbf{0} & & & \mathbf{0} \\ & & \vdots & & & \vdots \\ & & \mathbf{0} & & & \mathbf{0} \\ & & \bar{\mathbf{T}}^d \mathbf{H}_{e1e,j}^d & & \bar{\mathbf{T}}^e \mathbf{H}_{e1e,j}^e & \\ & & \mathbf{0} & & & \mathbf{0} \\ & & \vdots & & & \vdots \\ & & \mathbf{0} & & & \mathbf{0} \\ & & \mathbf{0} & & \bar{\mathbf{T}}^e \mathbf{H}_{e1e,k}^e & \\ & & \mathbf{0} & & & \mathbf{0} \\ & & \vdots & & & \vdots \\ & & \mathbf{0} & & & \mathbf{0} \end{array} \right) \end{matrix}$$

It should be noted that $\mathbf{H}_{e1e,i}^d \in \mathbb{R}^{6 \times 6}$, as it represents the element d flexibility matrix associated to node i . This node can be the origin (node 1) or the end (2), and it depends on the user introduction of the initial data.

Moreover, it should be noticed that the dimension of \mathbf{r} is the exact number of geometric restrictions and the matrix \mathbf{G} links each reaction to its corresponding equilibrium equation, and has as many rows as the number of equilibrium equations, which coincides with the number of rows in \mathbf{H} , $6n_{np}$. Finally, it is worth mentioning that is usual in the problems that will be solved by this software to have a fair larger number of stress parameters than equilibrium equations, which implies that the majority of structures will be hyperstatic, and thus can carry the external loads in diverse ways.

4.2 Yield criteria

The function that describes the yield surface is in general a nonlinear function of the stress, geometry, material and cross section parameters. Since external loads are only considered at the nodes, yield is only checked at the element nodes. If the aim were to use nonlinear optimization programs the original yield surface could be used. As in this project only the LP approach is discussed, firstly a linearization process must be carried out. This linearization process is the key for a successful yield surface approximation, hence it is worth highlighting the main features of yield conditions. Linearization of the yield surface will only be considered for **two-**

dimensional frames, since the 3D case is still being implemented. Thus, from now on yield criteria will be referred as yield curve rather than yield curve.

4.2.1 Class type

According to its behavior facing normal stress, the cross section of a beam element in a structure can be classified as follows

1. Plastic
2. Compact
3. Semi-compact
4. Slender

This classification provides an average idea of how much local instability (dent) which arise during the process of loading can be capable of limiting both the strength of the cross section (moment that can be reached without collapsing) and its rotation capability (curvature that can be adopted without collapsing).

Depending on the sensitivity of a cross section to bear with local instabilities four types or section classes can be defined:

- **Class 1 (plastic)** sections are capable of not only reaching its fully plastic moment without arising instability problems, but also have enough rotation capacity to form a plastic hinge, and allow the perfect plasticity behavior demanded for a global plastic analysis.
- **Class 2 (compact)** sections are capable of reaching its fully plastic moment without arising instability problems, although don't have enough rotation capacity to form a plastic hinge for a global plastic analysis. As a consequence of this latter fact, in isostatical structures the same global exploitation of the material as if it were a plastic section is permitted. However, material exploitation in hyperstatical structures is strictly lower than the one that can be attained using plastic sections.
- **Class 3 (semi-compact)** sections present local dent problems before attaining the fully plastic moment and once surpassed the elastic moment. The section resistant moment will be considered equal to the elastic moment.
- **Class 4 (slender)** sections are incapable of even developing its elastic capacity in the most compressed metallic fiber, due to instabilities at the compressed sheets.

The assignation of a class to a determined cross section involves a number of parameters such as the material’s elastic limit, the geometry of the section, the slenderness of its fully or partially compressed panels and the loads at which the section is undergone. Actually, the section type will be known *a priori* by using the various steel cross section catalogues available⁴, and as the functioning of the software is shown this matter will be deepened.

4.2.2 Yield function definition

As stated before, plastic analysis only makes sense if considering cross sections with C1 or C2 collapse state, since these are the ones that attain plastic flow. Furthermore, geometry of the cross section has revealed as a decisive factor in the nonlinear yield curve, thus it is crucial to analitically know its description of the most commonly used cross sections. This project will focus on the doubly symmetric I/H-shaped cross sections, basically the IPE and HEB series, due to its vast presence in steel structures and widespread use. Rectangular cross sections should also be mentioned, but will be no further discussed as a rectangular solid steel section is seldom found. Nowadays only doubly symmetric I/H-shaped cross sections can be found on the Cross Section Library of the Structural Collapse Simulator, nonetheless there are currently being implemented non-symmetric cross sections such as T, U or L shape, as its yield function is found and the linearization procedure is developed.

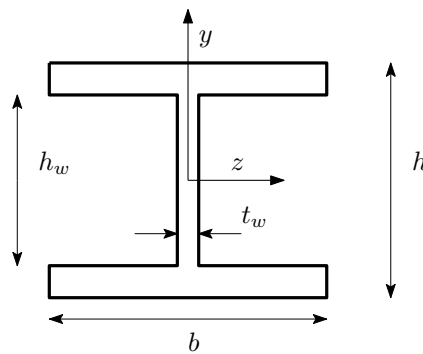


Figure 3: I/H-shaped symmetric cross section

Finding the yield curve definition in an axial force-bending moment cartesian plane consists of finding the relationship between N and M when the section undergoes combined bending and axial load, by forcing equilibrium in the zero strain axis (ZSA). In a doubly symmetric I/H-shaped cross section, the axial tension (N_T) and compression (N_C) forces that plastify the cross section with no bending are $N_T = -N_C = N_p = (t_w h_w + b(h - h_w))\sigma_p$ where σ_p , the yield stress, is assumed to coincide in tension and compression due to symmetries. The plastic bending moment that induces a plastic hinge to form under pure bending is

⁴The Structural Collapse Simulator is based on the *ArcelorMittal* catalogues.

$M^+ = -M^- = M_p = (t_w h_w^2 + b(h^2 - h_w^2))\sigma_p/4$, where h is the depth of the beam, h_w is the depth of the web, t_w is the thickness of the web and b is the width of the flanges as depicted in Figure 3. The area of the section can be approximated as $A = t_w h_w + b(h - h_w)$ ⁵.

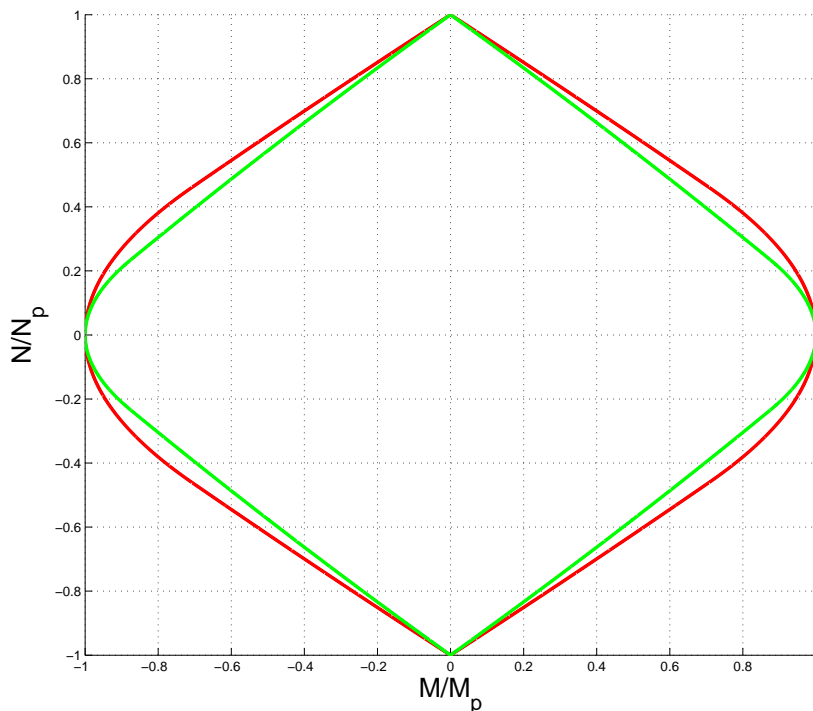


Figure 4: Yield curve for IPE550 (red) and HEB300 (green) cross sections

In this type of section, as the ZSA can be placed in the web or in the flanges, the yield curve is defined piecewise in each quadrant of the (M, N) plane. There are a total of eight different possibilities, depending on whether the ZSA is on the upper/lower flange/web and the sign of the axial and bending load. In Figure 4 the yield curve for different sections is represented.

4.2.3 Yield curve linearization

In the previous section the yield curve of the I/H-shaped cross section was explicitly found. Following the procedure briefly explained yield curves for standard beam cross sections can be determined as well. The yield function is clearly nonlinear,

⁵Real sections are not usually a plain union of three rectangles, as present a small curvature between the web and the flanges. In the problem modeling, as this assumption is made, the area given by the manufacturer is taken as A , and with this expression an equivalent t_w is deduced and used.

4 LIMIT STATE ANALYSIS AS A LP PROBLEM

although is interesting to notice from the yield curve depiction and from the general theory that every curve defines a convex set⁶ of feasible loads in the (M, N) plane. For comodity it is advisable to work on a normalized plane, i.e. the $\left(\frac{M}{M_p}, \frac{N}{N_p}\right)$ plane.

Since this project is headed into expressing the limit analysis as a LP problem, the nonlinear yield criteria must be expressed with a set of lines (linear constraints) that approximate the yield curve as precisely as desired. The strategy to follow here is to replace the nonlinear yield curve by two new polygons, inscribed and circumscribed respectively. The inscribed polygon will lead to a lower bound of the exact collapse load factor in the LP problem, whereas the circumscribed will yield an upper one. It is worth mentioning that the convexity of the yield curve is crucial to ensure that the approximations made always yield to upper or lower bounds. The objective is then to define a sequence of polygons that converge from inside and outside of the yield curve to the exact one. For this purpose, and exploiting the condition of double symmetry of the yield curve, only the first quadrant will be approximated and the results will be extended to the remaining three.

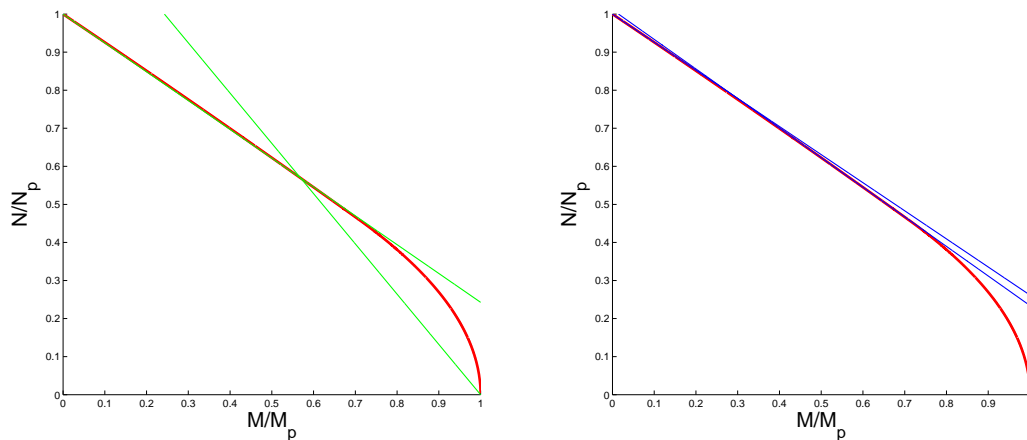


Figure 5: Inscribed and circumscribed polygons of a IPE550 yield curve using 8/10 lines per node respectively

The initial approximation of the **lower bound** (LB) is one line defined by the two extremal points $(1,0)$, $(0,1)$. This first guess corresponds the most conservative yield criteria condition, which is reflected at the engineering standards and codes as the safest approximation. The new polygon will consist of a set of lines with extreme points over the yield curve. In practice, the user will introduce the desired number of lines to approximate the yield curve of *every element* (not necessarily

⁶Similarity as defined in (2.2.2), a convex set \mathcal{C} in a real or complex vector space is one in which for all $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ and all $\alpha \in [0, 1]$, the point $(1 - \alpha)\mathbf{x} + \alpha\mathbf{y} \in \mathcal{C}$

the same lines for every element), with the only restriction to be a multiple of 4. The software will divide the first quadrant into sections with a beam of lines, and the intersection between each one of these and the yield curve will determine points over the yield curve. Since the yield curve is the one previously defined, it can be written as a compact function in the bending moment-axial force plane. As a consequence, the procedure of finding the points is as simple as a Newton-Raphson 1D technique to seek the abscissa where the difference between the yield curve and each line vanishes. Due to symmetries a collection of points over the yield curve is available, whereby is trivial to define a set of lines that link two consecutive points, forming an inscribed polygon. Every point in and on the polygon is feasible because it is either in or on the yield curve. It is obvious that this latter fact leads to the computation of a lower bound of the exact collapse load.

The **upper bound** (UB) is computed similarly. However, in this case no Newton-Raphson is needed, as the strategy is to employ the collection of points found in the lower bound approximation and define tangent lines to the yield curve at every point. It is worth noticing that in this procedure a notable step is found: the singularities at the points (0,1), (0,-1) of the $\left(\frac{M}{M_p}, \frac{N}{N_p}\right)$ plane. The impossibility of defining a tangent line at these points is solved by defining a right and left tangent line using the lateral derivatives of the yield function at a neighborhood of these singular points. Thereby a discrepancy arises when comparing to the lower bound approximation; in the LB case n points define n lines, whereas in the UB case n points define $n + 2$ points. However, this disagreement is properly treated in the software. The points on the polygon are only feasible if they are on the yield curve, and obviously this approximation leads to unsafe load multipliers as the polygon is circumscribed in the actual yield curve. Nevertheless, this upper bound gives decisive information on "how good" the lower bound previously found is, i.e. the precision of the first computation, which is certainly a matter of importance if a study of convergence by creating a sequence of polygons is carried out.

In both cases the restrictions are linear, one restriction per edge of the polygon. Finally, this procedure expresses the linear restrictions in an adequate form for its inclusion in the linear approximation of the yield criteria in the LP problem. Figure 5 shows both the lower and upper approximation (first quadrant) of a IPE550 cross section.

4.2.4 Yield criteria as a LP problem

The original nonlinear criteria would lead to a nonlinear optimization problem

$$f(N, M_x, M_z^i, M_y^i) \leq g(m_1, \dots, m_k) \quad i = 1, 2 \quad (9)$$

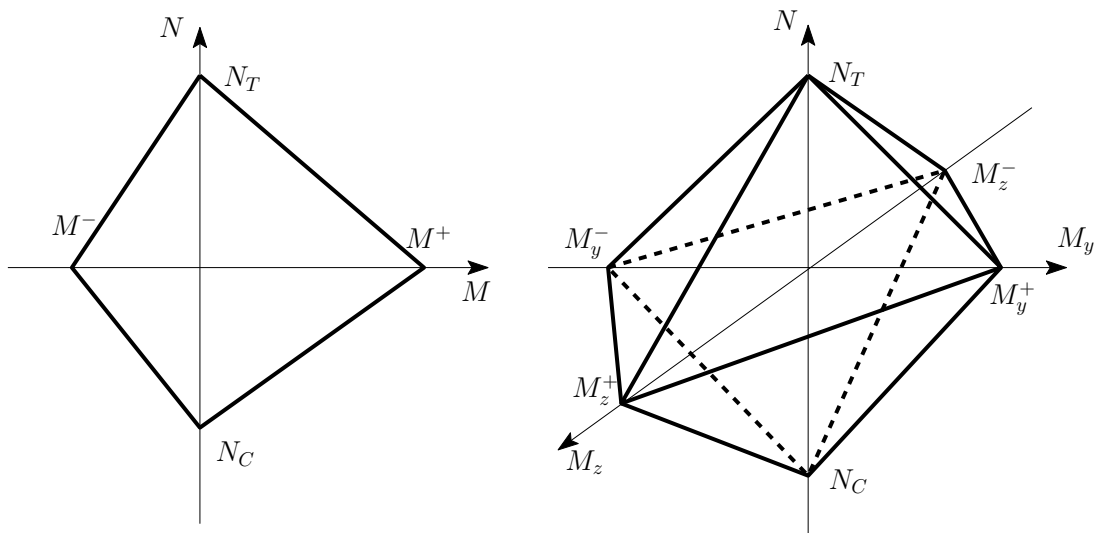


Figure 6: Simplest linearization for a general yield surface

where m_1, \dots, m_k are the material parameters that characterize the beam properties and the material, which can of course be different for every element. To perform the computation as a LP problem this criteria is linearized following the procedure described in the previous section. The case treated in this project is only concerned in finding the factor that maximizes the external load so as to induce collapse in the structure. Therefore only the left-hand side of (9) must be linearized and optimized. If the aim were to optimize the material resources for a given load, the inverse process should be carried out. For each element, the condition of not exceeding yield curve can be expressed as

$$\mathbf{F}^i \begin{Bmatrix} N \\ M_z^i \\ M_y^i \end{Bmatrix} \leq \mathbf{g}^i \text{ for } i = 1, 2$$

The linearization will be discussed for the 2D frame, since for the 3D case the method is not yet available. Matrix \mathbf{F}^i has at least 8 rows (when the yield curve is approximated with 4 lines per node, see Figure 6, if computing the lower bound or 12 rows (the tangent lines to the 8 points on the curve as detailed beforehand) if computing the upper bound. The number of rows increases as the approximation of the yield curve is more accurate, i.e. more lines are used. In fact, each edge of the polygon constitutes a restriction, therefore an extra row for \mathbf{F}^i . The previous equation can be regarded for each element as

$$\mathbf{F}_{\text{ele}} \boldsymbol{\sigma}_{\text{ele}} \leq \mathbf{g}_{\text{ele}} \quad (10)$$

and after an easy assembly procedure the yield condition is expressed in the whole structure by

$$\mathbf{F} \boldsymbol{\sigma} \leq \mathbf{g} \quad (11)$$

4 LIMIT STATE ANALYSIS AS A LP PROBLEM

The number of columns of \mathbf{F} equals the total number of generalized stress parameters, i.e. the dimension of $\boldsymbol{\sigma}$, $3\mathbf{n}_{e1}$. The number of rows is at least twice (one for each node of each element) the number of structure elements times the number of lines used to approximate the yield curve, minimum 4/6 per node, depending on the bound that is being computed. From now on, a subscript will be used in order to differentiate among the upper and the lower bound. Hence, the number of rows of \mathbf{F}_{LB} is minimum $8\mathbf{n}_{e1}$, while the number of rows of \mathbf{F}_{UB} is at least $12\mathbf{n}_{e1}$. It should be noted that if a nodal point is a junction of several bars, yield at this node will be checked as many times as the number of bars that join at the node. Thus if plastic flow is reached at a node information about which bar contributes to the plastic attainment can be obtained.

The simplest yield curve for 2D frames consists of 4/6 lines if computing either the lower or the upper bound. Lower bound inequalities for only one element are now presented as a general case, due to the triviality of the assembly process. The upper bound case is omitted because expressing the tangent lines to the yield curve with a compact expression could be tedious.

$$\begin{pmatrix} 1/N_T & 1/M^+ & 0 \\ 1/N_T & 1/M^- & 0 \\ 1/N_C & 1/M^- & 0 \\ 1/N_C & 1/M^+ & 0 \\ 1/N_T & 0 & 1/M^+ \\ 1/N_T & 0 & 1/M^- \\ 1/N_C & 0 & 1/M^- \\ 1/N_C & 0 & 1/M^+ \end{pmatrix} \begin{pmatrix} N \\ M_z^1 \\ M_z^2 \end{pmatrix} \leq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

where N_T and N_C are the limit axial values of tension and compression, whereas M^+ and M^- are the fully plastic positive and negative moments.

However, for the 3 dimensional case the inner and outer polyhedra approximating the yield surface coincide, see Figure 6. This forces $\mathbf{F}_{LB} = \mathbf{F}_{UB}$, and the elementary matrix inequalities become

$$\begin{pmatrix} 1/N_T & 0 & 1/M_z^+ & 0 & 1/M_y^+ & 0 \\ 1/N_T & 0 & 1/M_z^+ & 0 & 1/M_y^- & 0 \\ 1/N_T & 0 & 1/M_z^- & 0 & 1/M_y^+ & 0 \\ 1/N_T & 0 & 1/M_z^- & 0 & 1/M_y^- & 0 \\ 1/N_C & 0 & 1/M_z^- & 0 & 1/M_y^+ & 0 \\ 1/N_C & 0 & 1/M_z^- & 0 & 1/M_y^- & 0 \\ 1/N_C & 0 & 1/M_z^+ & 0 & 1/M_y^+ & 0 \\ 1/N_C & 0 & 1/M_z^+ & 0 & 1/M_y^- & 0 \\ 1/N_T & 0 & 0 & 1/M_z^+ & 0 & 1/M_y^+ \\ 1/N_T & 0 & 0 & 1/M_z^+ & 0 & 1/M_y^- \\ 1/N_T & 0 & 0 & 1/M_z^- & 0 & 1/M_y^+ \\ 1/N_T & 0 & 0 & 1/M_z^- & 0 & 1/M_y^- \\ 1/N_C & 0 & 0 & 1/M_z^- & 0 & 1/M_y^+ \\ 1/N_C & 0 & 0 & 1/M_z^- & 0 & 1/M_y^- \\ 1/N_C & 0 & 0 & 1/M_z^+ & 0 & 1/M_y^+ \\ 1/N_C & 0 & 0 & 1/M_z^+ & 0 & 1/M_y^- \\ 0 & 1/\text{To1} & 0 & 0 & 0 & 0 \\ 0 & -1/\text{To1} & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} N \\ M_x \\ M_z^1 \\ M_z^2 \\ M_y^1 \\ M_y^2 \end{pmatrix} \leq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

where To1 is a numerical tolerance⁷ In this case it reflects that beams under consideration do not have any torsional strength.

4.2.5 Yield criteria for trusses

If the structures only supports axial loads, the yield criteria is simpler, as it is expressed by the conditions $N \leq N_T$ and $-N \leq -N_C$, where N_T, N_C are the axial tension and compression limits respectively. Once assembled, \mathbf{F} matrix has two rows per element times the number of elements, thus its number of rows is $2n_{e1}$. Obviously, \mathbf{F} has a n_{e1} columns, one axial stress per element, in both 2D and 3D. Consequently, yield criteria as a LP problem can be written as

⁷In practice, $\text{To1} = \sqrt{\text{eps}}$, where eps is the distance from 1.0 to the next largest double-precision number in Matlab[®].

$$\begin{array}{cccc}
 & 1 & \dots & e & \dots & \mathbf{n}_{e1} \\
 1 & \left(\begin{array}{ccc} 1/N_T^1 & 0 & 0 \\ 1/N_C^1 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right. & & & & 1 & \left(\begin{array}{c} 1 \\ 1 \end{array} \right) \\
 1 & & & & & & 1 & \\
 \vdots & \left. \begin{array}{ccc} \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{array} \right) & & & & \vdots & \left(\begin{array}{c} \vdots \\ \vdots \end{array} \right) \\
 e & & & & & & e & \left(\begin{array}{c} 1 \\ 1 \end{array} \right) \\
 e & & & & & & e & \\
 \vdots & \left. \begin{array}{ccc} \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{array} \right) & & & & \vdots & \left(\begin{array}{c} \vdots \\ \vdots \end{array} \right) \\
 \mathbf{n}_{e1} & & & & & & \mathbf{n}_{e1} & \left(\begin{array}{c} 1 \\ 1 \end{array} \right) \\
 \mathbf{n}_{e1} & \left(\begin{array}{ccc} 0 & 0 & 1/N_T^{\mathbf{n}_{e1}} \\ 0 & 0 & 1/N_C^{\mathbf{n}_{e1}} \end{array} \right. & & & & \mathbf{n}_{e1} & \left(\begin{array}{c} 1 \\ 1 \end{array} \right) \\
 & & & & & & &
 \end{array}
 \left(\begin{array}{c} N^1 \\ \vdots \\ N^e \\ \vdots \\ N^{\mathbf{n}_{e1}} \end{array} \right) \leq$$

4.3 Inclusion of Uniform Distributed Loads

The previous development has been made considering only point loads which are assumed to be applied at the nodes. If an external force is not applied at a node, the user must introduce an extra node in order to account for the load. Nonetheless, in the majority of structural and engineering problems distributed loads are often present, such as selfweight or dead loads, and more specifically uniform distributed loads (UDL) are usual. The technique adopted in this project, following previous authors and works such as [15], consists of transforming the UDL into nodal point loads in a consistent manner. The idea is to split the UDL into two equal point loads each one corresponding to the half of the UDL. The first approximation is quite rough, as will be seen lately, but this procedure can be further refined by subdividing the beam and consequently improving the approximation to the UDL.

For an arbitrary element e , where $e = 1, \dots, \mathbf{n}_{e1}$, let $\mathbf{p}^e = (p_x, p_y, p_z)^T$ be the element UDL vector defined in the **local** system of coordinates, see Figures **1** and **2**. Thus equilibrium at the beam element is no longer obtained using Equation (6), and can be expressed with the modified equation

$$\mathbf{q}^e = \mathbf{H}_{e1e}^e \boldsymbol{\sigma}_{e1e}^e - \lambda \mathbf{U}_{e1e}^e \mathbf{p}^e \tag{12}$$

where \mathbf{U}_{e1e} is the matrix that translates UDL to the nodes, which can be defined in 3D

$$\mathbf{U}_{e1e}^T = \frac{l}{2} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

where l is the beam element length. It should be pointed out that UDL is also amplified by the sought load factor λ . Since the element equilibrium equation is modified, so is the global equilibrium equation (8). The definitive equation that will be provided to the LP solver will be the following

$$\mathbf{H}\boldsymbol{\sigma} = \mathbf{G}\mathbf{r} + \lambda\hat{\mathbf{f}} \quad (13)$$

with the definition of $\hat{\mathbf{f}}$ as the force vector, which contemplates both nodal and UDL loads, namely $\hat{\mathbf{f}} = \mathbf{f} + \mathbf{U}\mathbf{p}$. In this latter expression, $\mathbf{p}^T = (\mathbf{p}_1^T, \dots, \mathbf{p}_{n_{el}}^T)$ and $\mathbf{U}^T = \left(\mathbf{U}_{ele}^1 \bar{\mathbf{T}}^{1T} \quad \dots \quad \mathbf{U}_{ele} \bar{\mathbf{T}}^{n_{el}T} \right)$ where $\bar{\mathbf{T}}$ is the rotation matrix defined in (4.1.2). The software will automatically set either \mathbf{f} or $\mathbf{U}\mathbf{p}$ to the zero matrix whenever a matrix of point loads or UDL respectively is not detected.

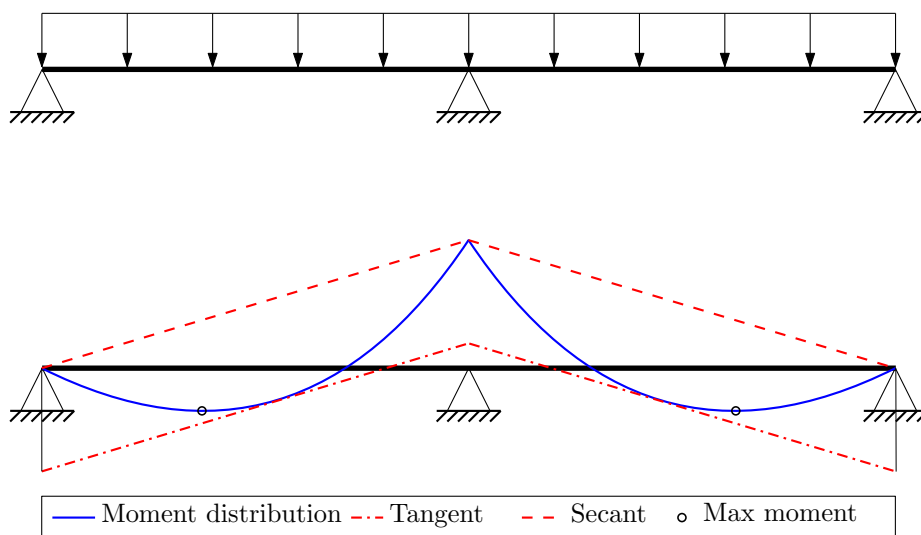


Figure 7: Moment diagram with secant and tangent approximation for a beam with UDL

Equation (13) reflects equilibrium for the case of a structure with UDL. Nevertheless, checking yield criteria is slightly more complicated if there is UDL applied at the structure. In this case, the moment distribution is nonlinear (in fact is parabolic), and checking yield criteria at the nodes, (10), could not be sufficient, as the greatest moment is often located in the middle of the span, in contrast with the point load case. Figure 7 shows the typical parabolic moment distribution of a continuous two-span beam with UDL, as well as a *secant* and a *tangent* distribution to the moment distribution.

The technique used in this software consists of defining a secant and a tangent distribution in order keep the moment distribution bounded. The secant distribution is a trivial linear interpolation of nodal moments whereas the tangent distribution is

the secant distribution translated a quantity of $p_i \frac{l^2}{8}$, $i = y, z$ depending on the axis that supports load, as can be observed in Figure 7. These two approximations of the moment diagram induce an upper bound and a lower bound on the exact collapse factor, which nature is completely different from the yield curve approximation, but both can be complemented.

If only the secant distribution is considered, yield inequalities contained in (11) will lead to an **upper bound** of the exact collapse factor. Since yield conditions are only checked at the nodes, if yield is attained at the element nodes the secant moment distribution produces a factor that coincides with the exact one. However, if yield occurs somewhere on the span, the secant load factor will be always greater than the actual one.

On the other hand, if the LP problem is computed including the both the secant and the tangent moment distribution, a **lower bound** can be obtained. Obviously, adding more inequalities to (11) can only decrease λ . Moreover, since the exact moment distribution always lies in between the secant and the moment, checking yield criteria at the nodes will always produce a worse case scenario. For this purpose, the generalized element stress vector for an arbitrary element e , σ_{e1e}^e must be modified according to the $p_i \frac{l^2}{8}$, $i = y, z$ translation when accounting the tangent distribution

$$\tilde{\sigma}_{e1e}^e = \sigma_{e1e}^e - \lambda \Sigma_{e1e}^e \iff \tilde{\sigma}_{e1e}^e = \begin{pmatrix} N \\ M_x \\ M_z^1 \\ M_z^2 \\ M_y^1 \\ M_y^2 \end{pmatrix} - \lambda \begin{pmatrix} 0 \\ 0 \\ p_y^e \frac{l^2}{8} \\ p_y^e \frac{l^2}{8} \\ p_z^e \frac{l^2}{8} \\ p_z^e \frac{l^2}{8} \end{pmatrix}$$

Thus, equation (10) transforms into

$$\mathbf{F}_{e1e} \tilde{\sigma}_{e1e} \leq \mathbf{g}_{e1e} \iff \mathbf{F}_{e1e} (\sigma_{e1e} - \lambda \Sigma_{e1e}^e) \leq \mathbf{g}_{e1e} \iff \mathbf{F}_{e1e} \sigma_{e1e} \leq \mathbf{g}_{e1e} + \lambda \mathbf{F}_{e1e} \Sigma_{e1e}^e$$

by naming $\tilde{\mathbf{g}}_{e1e} = \mathbf{F}_{e1e} \Sigma_{e1e}^e$ we obtain the expression for the tangent distribution. Obviously, if no UDL is present $\tilde{\mathbf{g}}_{e1e} = \mathbf{0}$. The assembly process is trivial, and the matrices for the whole structure are easily obtained. The compact form that will be used in the LP problem is reached by considering both the secant and the tangent distribution in the same equation, i.e.

$$\begin{pmatrix} \mathbf{F} \\ \mathbf{F} \end{pmatrix} \sigma - \lambda \begin{pmatrix} \mathbf{0} \\ \tilde{\mathbf{g}} \end{pmatrix} \leq \begin{pmatrix} \mathbf{g} \\ \mathbf{g} \end{pmatrix}$$

Finally, it is important to recall that if the UDL has a component along the x -direction of the local coordinate system the axial forces must be modified accordingly at corresponding equations in (10) (one for each node). This approach always produce an upper and lower bound of the load factor and this for any subdivision of the elements bearing UDL. Moreover, since the distance between the secant and the tangent moment distributions is $p_i \frac{l^2}{8}$ with i either y or z depending on the flexural axis considered, the bound gap decreases quadratically, obviously with the square of the subdivision length.

4.3.1 Combination of yield and UDL bounding

When computing the collapse load factor via the LP problem, two different approaches have been proposed that induce lower and upper bounds to the exact load factor. The approximation of the yield curve using an inscribed polygon produces a lower bound and the circumscribed polygon an upper bound. On the other hand, and considering UDL on the structure, using a secant distribution to the moment diagram leads to an upper bound of the load factor, whereas using both the secant and the tangent distribution a lower bound is obtained.

If loading on the structure consists only of external point loads, bounding of the load factor is originated only by the approximation of the yield curve. However, the most common case in structural calculus involves UDL, and thus it must be considered. In order to obtain consistent bounds on the exact collapse load factor, the **Lower Bound** will be the one obtained by using an inscribed polygon to the yield curve and the secant and tangent distribution to the moment diagram. Consequently, the **Upper Bound** is computed by approximating the yield curve with a circumscribed polygon and the moment distribution using the secant interpolation. Thereby, this procedure ensures that the Upper Bound is always greater than the Lower Bound, and therefore the **bound gap** (BG), which is the difference between the **UB** and the **LB**, is positive.

Recall that the latter refers to 2D frames, since bounding for 3D frames is only accomplished via the secant and the tangent moment distribution, as the yield surface is approximated inwards and outwards using the simplest linearization in Figure 6.

4.4 The LP problem

For the general case of a structure with both point loads and UDL, and once equilibrium and yield conditions are defined, the optimization problem can be written as

$$\begin{aligned}
& \text{maximize} && \lambda \\
& \text{subject to} && \mathbf{H}\boldsymbol{\sigma} = \mathbf{G}\mathbf{r} + \lambda\hat{\mathbf{f}} \\
& && \mathbf{F}_{\text{UB}}\boldsymbol{\sigma} \leq \mathbf{g}_{\text{UB}}
\end{aligned} \tag{14}$$

if the aim is to obtain the UB or

$$\begin{aligned}
& \text{maximize} && \lambda \\
& \text{subject to} && \mathbf{H}\boldsymbol{\sigma} = \mathbf{G}\mathbf{r} + \lambda\hat{\mathbf{f}} \\
& && \mathbf{F}_{\text{LB}}\boldsymbol{\sigma} \leq \mathbf{g}_{\text{LB}} \\
& && \mathbf{F}_{\text{LB}}\boldsymbol{\sigma} - \lambda\tilde{\mathbf{g}} \leq \mathbf{g}_{\text{LB}}
\end{aligned} \tag{15}$$

if computing the LB. In these formulations, $\lambda \in \mathbb{R}$, $\boldsymbol{\sigma} \in \mathbb{R}^{6n_{e1}}$ and $\mathbf{r} \in \mathbb{R}^{n_{gr}}$ are unknown variables (n_{gr} is the number of geometric restrictions), whereas $\mathbf{H} \in \mathbb{R}^{6n_{np} \times 6n_{e1}}$, $\mathbf{G} \in \mathbb{R}^{6n_{np} \times n_{gr}}$, $\hat{\mathbf{f}} \in \mathbb{R}^{6n_{np}}$, $\mathbf{F}_{\text{LB}} \in \mathbb{R}^{18n_{e1} \times 6n_{e1}}$, $\mathbf{F}_{\text{UB}} \in \mathbb{R}^{18n_{e1} \times 6n_{e1}}$, $\mathbf{g}_{\text{LB}} \in \mathbb{R}^{18n_{e1}}$, $\mathbf{g}_{\text{UB}} \in \mathbb{R}^{16n_{e1}}$ and $\tilde{\mathbf{g}} \in \mathbb{R}^{16n_{e1}}$ are given data. On this latter five the number of rows is a minimum, since improving the approximation of the yield surface provides more equations, hence more rows are added to the matrix. As has been commented, this implementation for 3D frames is not yet available, but it is for 2D frames. Note that although $\lambda \in \mathbb{R}$, the solutions of the LP problem always imply that the collapse load is nonnegative. The proof of this fact is that taking $\lambda = 0$ there exists a feasible stress state $(\boldsymbol{\sigma}^0, \mathbf{r}^0)$ that verifies equilibrium and such that yield is not attained at any node, i.e. $\mathbf{H}\boldsymbol{\sigma}^0 = \mathbf{G}\mathbf{r}^0$ and $\mathbf{F}\boldsymbol{\sigma}^0 \leq \mathbf{g}$ which can be $(\boldsymbol{\sigma}^0, \mathbf{r}^0) = (0, 0)$. Since the problem is a maximization over λ , nonnegativity is thereby guaranteed. For expressing the LP problem in the general form described by (4) it will be clearly differentiated the LB and UB computation, as well as its dual problem.

4.4.1 Upper Bound

Auxiliary variables will be used in order to write the LP problem in the compact form. Using the following definitions

$$\begin{aligned}
\mathbf{c}^T &= (-1, 0, \dots, 0), & \mathbf{x}^T &= (\lambda, \boldsymbol{\sigma}^T, \mathbf{r}^T), \\
\mathbf{D} &= \begin{matrix} & 1 & 6n_{e1} & n_{gr} \\ 18n_{e1} & \left(\mathbf{0} & \mathbf{F}_{\text{UB}} & \mathbf{0} \right), \end{matrix} & \mathbf{e} &= \mathbf{g}_{\text{UB}}, \\
\mathbf{A} &= \begin{matrix} & 1 & 6n_{e1} & n_{gr} \\ 6n_{np} & \left(\hat{\mathbf{f}} & -\mathbf{H} & \mathbf{G} \right), \end{matrix} & \mathbf{b} &= \mathbf{0},
\end{aligned}$$

problem (14) is equivalent to (4). As a direct consequence, its dual problem should be equivalent to (5), which expressed in the structural notation states as

$$\begin{aligned}
& \text{minimize} && \mathbf{g}_{\text{UB}}^T \boldsymbol{\eta} \\
& \text{subject to} && \hat{\mathbf{f}}^T \boldsymbol{\nu} = 1 \\
& && \mathbf{G}^T \boldsymbol{\nu} = 0 \\
& && \mathbf{F}_{\text{UB}}^T \boldsymbol{\eta} = \mathbf{H}^T \boldsymbol{\nu} \\
& && \boldsymbol{\eta} \geq 0
\end{aligned} \tag{16}$$

where the first restriction is a normalization of the work produced by external forces. This term, $\hat{\mathbf{f}}^T \boldsymbol{\nu} = 1$ involves external forces, both nodal and UDL, and the generalized displacement rates $\boldsymbol{\nu}$. Consequently, if external work is equalized to the internal dissipation, that is

$$\lambda \hat{\mathbf{f}}^T \boldsymbol{\nu} = \mathbf{g}_{\text{UB}}^T \boldsymbol{\eta} \tag{17}$$

this latter equation shows that the objective function of the dual problem for any kinematically admissible solution is the load factor λ found using the kinematic theorem, as it must be according to [10].

For dimensional criteria, the dual variables $\boldsymbol{\eta}$, $\boldsymbol{\nu}$ belong to $[\mathbb{R}^+]^{18n_{e1}}$ and $\mathbb{R}^{6n_{np}}$. Moreover, the latter is clearly associated to the displacement rate and angular velocity of each node of the structure. Due to it, it is easy to observe that the restriction $\mathbf{G}^T \boldsymbol{\nu} = 0$ imposes the geometric restrictions. The other dual variable, $\boldsymbol{\eta}$, is a vector of nonnegative plastic multiplier rates. At optimality, only the active constraints of the primal problem associated with plastic failure will have its corresponding positive plastic multiplier rate. Thus $\mathbf{g}_{\text{UB}}^T \boldsymbol{\eta}$ is the plastic dissipation of internal work. Finally, the third constraint, $\mathbf{F}_{\text{UB}}^T \boldsymbol{\eta} = \mathbf{H}^T \boldsymbol{\nu}$, imposes compatibility.

4.4.2 Lower Bound

The same notation is used, although variables here are slightly different

$$\begin{aligned}
\mathbf{c}^T &= (-1, 0, \dots, 0), & \mathbf{x}^T &= (\lambda, \boldsymbol{\sigma}^T, \mathbf{r}^T), \\
\mathbf{D} &= \begin{matrix} & 1 & 6n_{e1} & n_{gr} \\ 18n_{e1} & \left(\begin{array}{ccc} \mathbf{0} & \mathbf{F}_{\text{LB}} & \mathbf{0} \\ -\tilde{\mathbf{g}} & \mathbf{F}_{\text{LB}} & \mathbf{0} \end{array} \right), & \mathbf{e} &= \left(\begin{array}{c} \mathbf{g}_{\text{LB}} \\ \mathbf{g}_{\text{LB}} \end{array} \right), \\ 18n_{e1} & & & \\
\mathbf{A} &= \begin{matrix} & 1 & 6n_{e1} & n_{gr} \\ 6n_{np} & \left(\begin{array}{ccc} \hat{\mathbf{f}} & -\mathbf{H} & \mathbf{G} \end{array} \right), & \mathbf{b} &= \mathbf{0} \\ 6n_{np} & & & \end{matrix}
\end{aligned}$$

This development supposes that UDL loads are present in the structure, which is the most general case. If not, the last row of \mathbf{D} and \mathbf{e} must be suppressed, and the rename that is now proposed need not to be taken into account. Therefore, if the structure bears UDL loads, in order to obtain the dual problem, the elements of the \mathbf{D} matrix are renamed in a more compact form

$$\bar{\mathbf{F}}_{\text{LB}} = \begin{pmatrix} \mathbf{F}_{\text{LB}} \\ \mathbf{F}_{\text{LB}} \end{pmatrix}, \quad \bar{\mathbf{g}}_{\text{LB}} = \begin{pmatrix} \mathbf{g}_{\text{LB}} \\ \mathbf{g}_{\text{LB}} \end{pmatrix}, \quad \tilde{\mathbf{g}} = \begin{pmatrix} \mathbf{0} \\ \tilde{\mathbf{g}} \end{pmatrix}$$

With this new notation, the dual problem is written

$$\begin{aligned} & \text{minimize} && \bar{\mathbf{g}}_{\text{LB}}^T \boldsymbol{\eta} \\ & \text{subject to} && \hat{\mathbf{f}}^T \boldsymbol{\nu} - \tilde{\mathbf{g}}^T \boldsymbol{\eta} = 1 \\ & && \mathbf{G}^T \boldsymbol{\nu} = 0 \\ & && \bar{\mathbf{F}}_{\text{LB}}^T \boldsymbol{\eta} = \mathbf{H}^T \boldsymbol{\nu} \\ & && \boldsymbol{\eta} \geq 0 \end{aligned} \tag{18}$$

Obviously, $\boldsymbol{\eta} \in [\mathbb{R}^+]^{36n_{e1}}$ and $\boldsymbol{\nu} \in \mathbb{R}^{6n_{np}}$.

4.4.3 Problem normalization

Both problems (14) and (15) are not normalized. The inequalities that represent the yield conditions, see (11), are pseudo-normalized, as the right-hand-side is always a vector of ones, whereas the equilibrium equation (13) is not normalized. Normalization and scaling of the problem may be important to convergence of the LP problem and thus it gains interest. However, it should be noted that any scaling in the external load vector $\hat{\mathbf{f}}$ has a direct effect not only in λ , but also in the dual variables $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$.

In the software both the external load vector and the vector $\tilde{\mathbf{g}}$, which contains the UDL, are scaled with the norm of $\hat{\mathbf{f}}$, before calling the LP solver. After the optimization the variables affected and the external load vector are rescaled in order to obtain the actual values.

4.4.4 Evaluation of the bound gap

In sections (4.2.4) and (4.3) two different approaches are detailed in order to bound the exact collapse load factor. In the first one, the gap is arbitrarily reduced as the number of lines to approximate the yield curve at each node is increased, while in the second the gap decreases quadratically as beam elements are split into halves. In order to determine an adaptive strategy to reduce the bound gap, which is one of the further goals of the software, it is required to state the contribution of every element to the total gap.

Using both equations (17) and the dual problem (16) an upper bound of the load factor at collapse is simply

$$\lambda_{\text{UB}} = \mathbf{g}_{\text{UB}}^T \boldsymbol{\eta}_{\text{UB}} = \boldsymbol{\eta}_{\text{UB}}^T \mathbf{g}_{\text{UB}} \tag{19}$$

To determine a lower bound of the load factor the equilibrium equation (13) must be used, namely

$$\mathbf{H}\boldsymbol{\sigma}_{\text{LB}} = \mathbf{G}\mathbf{r}_{\text{LB}} + \lambda_{\text{LB}}\hat{\mathbf{f}}$$

A lower bound of the collapse factor can be obtained right multiplying the expression before by $\boldsymbol{\nu}_{\text{UB}}$. Since $\boldsymbol{\nu}_{\text{UB}}$ verifies the normalization restriction, $\hat{\mathbf{f}}^T \boldsymbol{\nu} = 1$, and the geometric condition $\mathbf{G}^T \boldsymbol{\nu} = 0$, the lower bound becomes $\lambda_{\text{LB}} = \boldsymbol{\nu}_{\text{UB}}^T \mathbf{H}\boldsymbol{\sigma}_{\text{LB}}$, and applying compatibility, i.e. the third restriction of the dual problem (16)

$$\lambda_{\text{LB}} = \boldsymbol{\eta}_{\text{UB}}^T \mathbf{F}_{\text{UB}} \boldsymbol{\sigma}_{\text{LB}} \quad (20)$$

The expressions for an upper bound (19) and a lower bound (20) determine the bound gap of the structure

$$\Delta = \lambda_{\text{UB}} - \lambda_{\text{LB}} = \boldsymbol{\eta}_{\text{UB}}^T (\mathbf{g}_{\text{UB}} - \mathbf{F}_{\text{UB}} \boldsymbol{\sigma}_{\text{LB}})$$

which determine the accuracy of the computations. Regard that the second constraint of (14) and the positiveness of $\boldsymbol{\eta}$ imply that the bound gap is nonnegative. This latter expression is for the bound gap of the global structure. However, if an adaptive process is sought, information about the contribution of every element is required. This contribution can easily be determined, namely

$$\Delta = \sum_e \Delta^e \quad \text{with} \quad \Delta^e = \boldsymbol{\eta}_{\text{UB},e1e}^e{}^T (\mathbf{g}_{\text{UB},e1e}^e - \mathbf{F}_{\text{UB},e1e} \boldsymbol{\sigma}_{\text{LB},e1e}^e),$$

where $\Delta^e \geq 0$ as $\boldsymbol{\eta}_{\text{UB},e1e}^e \geq 0$ by definition and $\mathbf{g}_{\text{UB},e1e}^e - \mathbf{F}_{\text{UB},e1e} \boldsymbol{\sigma}_{\text{LB},e1e}^e \geq 0$ as it is stated in (10).

5 Use of Structural Collapse Simulator

Once the problem has been studied and its formulation as a LP problem has been written, it is highly interesting to detail how the Structural Collapse Simulator (SCS) is designed and how it can be employed by the user in order to obtain the collapse load factor for the desired structure. Firstly, it should be stated that the software utilized for developing the SCS is Matlab[®], as it provides an easy treatment of matrices and some efficient LP solvers. The program is designed with a descendent structure and the main operations can be summarised as data input, data processing, creation of matrices, LP solver and postprocessing. The central commanding program that the user will work on the most is named *main*. From this file the user decides which structure to compute and the postprocessing desired. It also accounts for the time taken for the computation, and it is displayed on the screen. Further references will be made to *main* as the whole procedure is detailed.

5.1 Data input

The data input is done via the so-called *drivers*, the programs that contain the main information about the structure that is about to be computed. For every new structure, the user will be asked to create a new *driver*, with different information but with the same exact development, and therefore it is crucial to regard it.

5.1.1 Logicals for structure characterization

Firstly, two booleans must be defined, and are initially set to zero. The first boolean, `istruss`, states whereas the computation will be made for a framed or for a trussed structure. If the second boolean, `isnorm`, is activated, no approximation of the yield curve is carried out. Consequently, the yield curve is described with 4 lines per node, see Figure 6, for both the lower and the upper bound, which is exactly the yield check that is recommended in the steel normative. This is the most conservative approach to the yield curve, but it is sometimes useful in order to determine the influence of the UDL in the bound gap, since is the only condition that prevents both the lower and the upper bound from coinciding.

5.1.2 Nodal and Connectivity matrices

Once known which kind of structure will be computed and how to make the yield curve computation, geometric data must be provided. The nodal matrix $\mathbf{X} \in \mathbb{R}^{n_{np} \times n_{sd}}$, with `nsd` being the number of spatial dimensions, 2 or 3, must be entered with each row referring to a node, with the x , y and z coordinates as the first, second and third column. Node numeration will automatically be assigned by the SCS according to the node's row number in the \mathbf{X} matrix.

The connectivity matrix \mathbf{T} is defined as a Matlab[®] cell array. The first cell is a real-valued matrix belonging to $\mathbb{R}^{n_{e1} \times 3}$, which will be named \mathbf{T}_1 . In the first and the second column of the \mathbf{T}_1 matrix the beginning and the end nodes of the beam element must be introduced, according to the coordinate system defined in Figure 1. As happened before, the elements numeration depends on how the \mathbf{T}_1 matrix is constructed. In the third column the number of lines to approximate the yield curve⁸ of **each node** in the element is detailed. This number must be a multiple of 4, and can be different for each element. It should be regarded that this number is useless in case the `isnorm` is set to 1, which is the case for 3D frames.

The second cell, \mathbf{T}_2 , is a string array, with as many columns as the number of elements `ne1`. Each column contains a string that indicates the cross section of each

⁸This number will be used for the inscribed polygon, and the circumscribed polygon will consist of essentially the same number of lines plus the extra restrictions detailed in (4.2.3), which will be regarded afterwards.

element for the further check of the yield curve. There exists the option of plotting the structure to reassure the introduction of nodal and connectivity data has been successful, by using the function *PlotStructure* in the *driver*. When computing a 3D frame an additional cell T_3 must be entered, which contains the direction of the web vector for every beam in the global coordinate system.

5.1.3 External Load matrices

Furthermore, load conditions must be entered by the user. Since both point loads and UDL are accounted, load data may concern either one or two matrices. The matrix of external point loads, namely P , contains an undetermined number of rows. This latter is due to the fact that the P needs not to have an entrance for every node, but only for the nodes with point loads. Hence, it will have as many rows as the number of nodes carrying external forces, namely $1 \leq \# \leq n_{np}$. This implies that the user is not asked to fill the matrix with 0s for the non-loaded nodes, which is sensible accounting the dimension of the problems that will be solved. The number of columns and the data contained in them varies in each case, namely

- $nsd = 2$ $istruss = 0$ $P = (\text{node}, P_x, P_y, M_z)$
 $nsd = 2$ $istruss = 1$ $P = (\text{node}, P_x, P_y)$
- $nsd = 3$ $istruss = 0$ $P = (\text{node}, P_x, P_y, P_z, M_x, M_y, M_z)$
 $nsd = 3$ $istruss = 1$ $P = (\text{node}, P_x, P_y, P_z)$

The forces must be referred to a **global coordinate system**, which must be defined by the user prior to SCS computations, as X , T and P matrices highly depend on it. According to the global coordinate system, signs must be placed correctly, and in general considering counter-clockwise moments to be positive.

The matrix that carries the UDL loads will normally be named UDL , and similarly to P will have no stipulated number of rows. In fact, this matrix has as many rows as the number of elements with UDL, thus $1 \leq \# \leq n_{e1}$. Regarding the columns, the cases now are

- $nsd = 2$ $istruss = 0$ $UDL = (\text{element}, p_x, p_y)$
- $nsd = 3$ $istruss = 0$ $UDL = (\text{element}, p_x, p_y, p_z)$

The loads are defined using **local coordinate system** defined in Figure 1.

5.1.4 Geometric Restrictions matrix

Finally, geometric restrictions such as embeddings or simple supports need to be introduced. This will be done using the C matrix, with n_{gr} rows and $3k$ columns, with $k \geq 1$. With the moving index $i = 0, \dots, k - 1$, for every row column $3i + 1$

contains the node where to apply the geometric restriction, column $3i + 2$ regards the movement that is impeded, and takes values $C(3i + 3) = 1, \dots, 6$ (each number regarding the movement caused by each nodal force) and column $3i + 3$ provides the coefficient that must be applied to the geometric restriction. This enables the user to account for supports forming an angle with the global coordinate system, thus $-1 \leq C(3i + 3) \leq 1$.

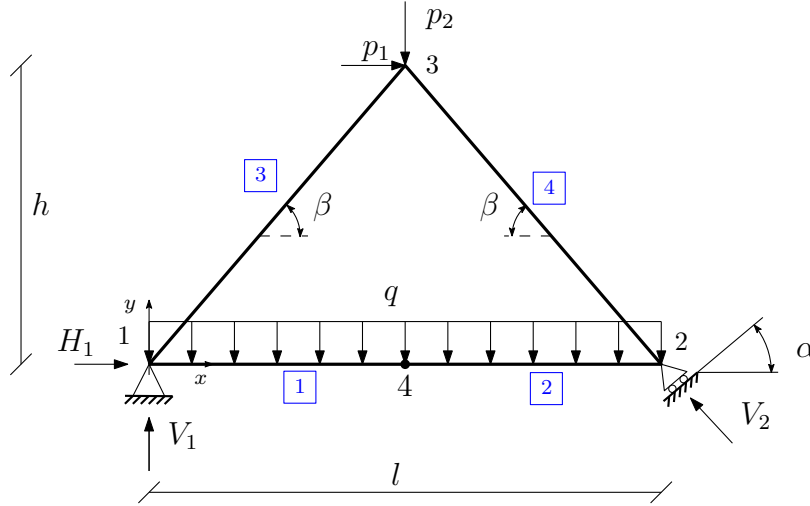


Figure 8: Schematic representation for a basic triangular structure

5.1.5 Example of data input

Finally, a simple example is now provided in order to illustrate how data has to be introduced for a general triangular structure, see Figure 8. For the diagonal beams of the structure an IPE360 cross section is used, whereas for the horizontal span an IPE550 section is employed. For this example, the simplest approximation of UDL is considered, i.e. only one subdivision is made. The global coordinate system is established with its origin on the first node, as depicted in the figure. In order to obtain the numeration seen in the schematic representation, the matrices \mathbf{X} and \mathbf{T} must be defined as

$$\mathbf{X} = \begin{pmatrix} 0 & 0 \\ l & 0 \\ l/2 & h \\ l/2 & 0 \end{pmatrix}, \quad \mathbf{T}_1 = \begin{pmatrix} 1 & 4 & 4 \\ 4 & 2 & 4 \\ 1 & 3 & 4 \\ 2 & 3 & 4 \end{pmatrix},$$

$$\mathbf{T}_2 = (\text{'IPE550'} \quad \text{'IPE550'} \quad \text{'IPE360'} \quad \text{'IPE360'})$$

supposing that the number of planes desired for the approximation of the yield curve is 4 per node, hence 8 per element. Since it is a 2D frame, the cell array containing the web vectors need not to be defined. With the nodal and connectivity matrices

defined, a trivial check is recommended to carry out in order to ensure the data is properly introduced. Giving values to the generic spans, namely $l = 3m$, $h = 2m$, the structure can be depicted activating the command *PlotStructure*, as can be seen in Figure 9.

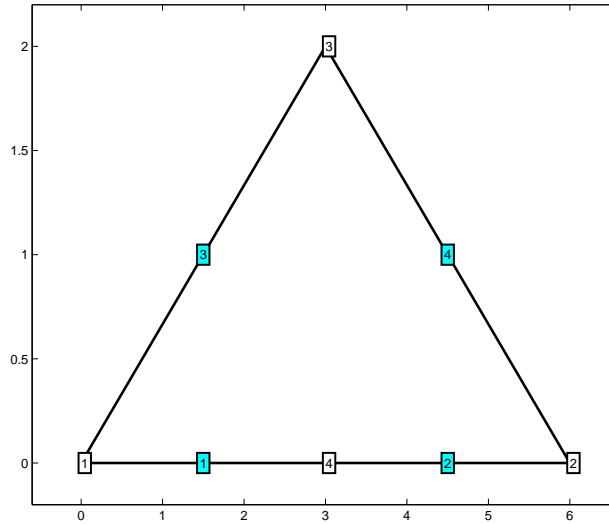


Figure 9: SCS output of *PlotStructures* function for the triangular frame

The external load matrices P and UDL will shape as follows

$$P = \begin{pmatrix} 3 & p_1 & -p_2 & 0 \end{pmatrix}, \quad UDL = \begin{pmatrix} 1 & 0 & -q \end{pmatrix}$$

accounting that the loads depicted in Figure 8 are an absolute value and the direction is provided by the arrows. Finally, the matrix C regarding the geometric restrictions must be written as

$$C = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ 2 & 1 & -\sin \alpha & 2 & 2 & \cos \alpha \end{pmatrix}$$

according to the global coordinate system defined. At the first node both horizontal and vertical deflections are clearly restricted, and as no slope is provided the coefficient must be 1. At the first node, only the vertical movement is impeded in the direction perpendicular to the ground, but the reaction forces will not appear in the direction of the coordinate system, and thus coefficients must be applied for both movements. The negative sign for the sinus is due to the fact that the reaction force V_2 has been chosen in a way that his horizontal part has the opposite sense as the coordinate system, and thus a minus sign is required.

5.2 Data processing

After the data of the structure has been introduced, *main* will call a file named *checkData*, whose interest both is to ensure that the information contained in the *driver* is correct and can be therefore used by the SCS and to create the main variables that will appear throughout the problem.

5.2.1 Initialization of variables

Firstly, the number of spatial dimensions, the number of nodes and elements of the structure is set to variables **nsd**, **numnp** (number of nodal points) and **numel** (number of elements), by extracting the dimensions of the nodal and connectivity matrices respectively. The program also checks whether the number of columns of these matrices is correct.

Furthermore, other important variables are initialized. Depending on whether the structure to compute is a frame or a truss, values will be given to variables **ndofn** (number of *element* generalized stresses, namely $\dim \sigma_{e1e}^T$) and **neqeq** (number of equilibrium equations per *node*). The values are for each case are shown in Table 1

	nsd = 2		nsd = 3	
	istruss = 0	istruss = 1	istruss = 0	istruss = 1
ndofn	3	1	6	1
neqeq	3	2	6	3

Table 1: Variable assignation depending on dimension and structure's nature

It is also initialized the array **nyiel**, which is essentially the third column of the T_1 matrix containing the number of lines used for the inner approximation of the yield curve of every element. These elements will all be useful in order to create the matrices needed for raising the LP problem. From now on, as general variables have been introduced, the definition of the matrices will consider both frames and trusses. Moreover, the two types of structures are differentiated by creating a string variable named **StruType**, which in our case will be `_2D_Truss`, `_2D_Frame`, `_3D_Truss` or `_3D_Frame`, and will be useful for calling the exact program when dealing with truss and frame different programs without rechecking the boolean **istruss**.

5.2.2 Setting approximation for yield criteria

Besides from creating variables, a thorough review of the number of lines used to describe the yield curve at every node is developed. A routine named *set_nyield* is called, and its function is to replace the last column of the T_1 matrix containing the number of lines to approximate the yield curve per each node of the element for

the total lines that will be used per **element** for the lower bound⁹. In general, this number will be multiplied by 2, since the lines for an element is twice the lines for a single node.

However, this function is significant in two special cases: where the normative criteria is applied, i.e. `isnorm` is set to 1, or where some or all cross sections of elements are classified as C4. As seen in (4.2.1), these sections do not attain plastic flow, and thus it is preposterous to propose a the yield curve approximation, and yield will be checked using the simplest approximation in Figure 6, just as if normative criteria was applied. In order to work with C4 sections, the user must explicitly add it to the name of the cross section in the T_2 string array, using notation `'_C4'` at the end of the cross section name. When working with normative criteria, the function `set_nyield` will set the whole third column of the T_1 matrix automatically to either 18 planes or 8 lines per element, depending on the dimension of the structure. On the other hand, if the computation is carried without normative criteria (only 2D case can be computed this way so far) but some or all of beam elements are expected to behave as a C4¹⁰, the function will set the corresponding number of planes to 8, it does not have to be necessarily every element of the structure.

5.2.3 Dimensional check

Finally, `checkData` is relevant as it confirms whether the data for the **P**, **UDL** and **C** is properly introduced. Besides checking that the number of columns of the matrices are appropriate, it creates either an empty **P** or **UDL** matrix in case only **UDL** loads or point loads are considered respectively. Moreover, it displays an error if the user is trying to compute a trussed structure with **UDL** loads, stopping the execution of the **SCS**.

5.3 Creation of matrices

Once the data provided has proven to be correct, the **SCS** advances to create and assemble the matrices needed for setting the LP problem, namely **H**, $\hat{\mathbf{f}}$, **G** and all the matrices involved in the yield criteria, for upper and lower bound calculations. This is done via the program `Create_var`, which is addressed from the `main` file. The program creates the matrices one-by-one and stores them in the workspace, ready to be used by the LP solver.

⁹The upper bound is properly treated based on the lower bound information and the extra restrictions

¹⁰Class types of different cross sections ought to be known in advance by consulting catalogues. For instance, **IPE** and **HEB** sections are usually C1, sometimes C2. For further information, *Arcelor-Mittal* catalogues can be checked out.

5.3.1 Equilibrium matrices

The construction of the $\mathbf{H} \in \mathbb{R}^{\text{neqeq} * \text{numnp} * \text{numel} * \text{ndofn}}$ matrix, which is done in the function *CreateH*, will no further be discussed, as both the rotation and the assembly process has been widely detailed in (4.1.5). In order to obtain $\hat{\mathbf{f}} = \mathbf{f} + \mathbf{Up}$, the two independent matrices are firstly generated. The creation of \mathbf{f} , which consists of only point loads, is as simple as reshaping the input matrix \mathbf{P} into a column array of $\text{neqeq} * \text{numnp}$ rows, since for each node horizontal, vertical and moment loads are placed in this order, and can be done by calling *Createf*. The procedure for generating the \mathbf{Up} has been discussed in (4.3), and also a $\text{neqeq} * \text{numnp} \times 1$ array is obtained after rotation, assembly and reshaping process in *CreateUp* has finished. Once the sum is made, the euclidean norm of $\hat{\mathbf{f}}$ is stored as **Scale** in order to normalize the LP problem as seen in (4.4.3) The geometric restriction matrix $\mathbf{G} \in \mathbb{R}^{\text{neqeq} * \text{numnp} * \text{numgr}}$ is obtained after properly reshaping the \mathbf{C} matrix in order that each geometric restriction relates to the corresponding node and element, and the function responsible for it is *CreateG*.

5.3.2 Yield criteria matrices

Once equilibrium matrices are stored, yield criteria matrices must be generated, namely \mathbf{F}_{UB} , \mathbf{F}_{LB} , \mathbf{g}_{UB} , \mathbf{g}_{LB} and $\tilde{\mathbf{g}}$. A routine named *CreateFg* is used for creating these latter matrices. The \mathbf{g} arrays are the simplest, as consists only of a vector of ones. The amount of restrictions used to make an inner approximation the yield curve for every node of every element, which are the number of rows of \mathbf{g}_{LB} , can be found by summing the elements of **nyiel**, that is **sum_nyiel**. The number of rows of \mathbf{g}_{UB} is essentially the number of rows of \mathbf{g}_{LB} plus the extra restrictions. As explained in (4.2.4), when dealing with a double I/H-shaped section the extra restrictions are 2 per node, i.e. 4 per element. Nevertheless, extra restrictions must be checked at every element since C4 cross section beams obviously do not have extra restrictions. The total amount of extra restrictions is a temporary variable named **extra_rest**. Of course, if is **isnorm=1** no extra restrictions are used. Hence, $\mathbf{g}_{\text{LB}} \in \mathbb{R}^{\text{sum_nyiel} \times 1}$ and $\mathbf{g}_{\text{UB}} \in \mathbb{R}^{(\text{sum_nyiel} + \text{extra_rest}) \times 1}$. Note that yield criteria for trusses has already been specified in (4.2.5). The following developments are only for 2D frames, since 3D frames are computed using normative criteria and the matrices are the ones defined in Section (4.2.4).

According to the latter, the dimensions of the \mathbf{F} matrices are easily determined, namely $\mathbf{F}_{\text{UB}} \in \mathbb{R}^{(\text{sum_nyiel} + \text{extra_rest}) * \text{numel} * \text{ndofn}}$ and $\mathbf{F}_{\text{LB}} \in \mathbb{R}^{\text{sum_nyiel} * \text{numel} * \text{ndofn}}$. The coefficients of the two matrices are found by calling the function *Create_Fgele_2D_Frame* for each element. For each element the function returns the elementary matrix that appears in (4.2.4) if **isnorm=1** or the cross section is a C4. If none of this conditions happens, then another function named *yield_Ishape_2D*¹¹ is accessed in order

¹¹So far yield has only been studied for I/H-shaped cross sections.

5 USE OF STRUCTURAL COLLAPSE SIMULATOR

to find the desired inscribed and circumscribed polygon of the yield curve using the technique described in (4.2.3). Once the collection of points is found, the lines are redefined in the symmetric form, that is $\frac{M}{A M_p} + \frac{N}{B N_p} = 1$. For a generic element whose total number of lines used to approximate the yield curve at one of the two nodes is n_{1in} , multiple of 4, a set of pairs of coefficients for both inscribed $(a_i^k, b_i^k), k = 1, \dots, n_{1in}/4$ and circumscribed $(a_c^l, b_c^l), l = 1, \dots, n_{1in}/4 + 2$, with lines for the first quadrant is obtained. Using the symmetries and the material properties, the vectors that represent the final coefficients are

$$\begin{aligned}
 A_i &= \begin{pmatrix} 1/a_i^1 N_T \\ \vdots \\ 1/a_i^{n_{1in}/4} N_T \\ 1/a_i^1 N_T \\ \vdots \\ 1/a_i^{n_{1in}/4} N_T \\ 1/a_i^1 N_C \\ \vdots \\ 1/a_i^{n_{1in}/4} N_C \\ 1/a_i^1 N_C \\ \vdots \\ 1/a_i^{n_{1in}/4} N_C \end{pmatrix} & B_i &= \begin{pmatrix} 1/b_i^1 M^+ \\ \vdots \\ 1/b_i^{n_{1in}/4} M^+ \\ 1/b_i^1 M^- \\ \vdots \\ 1/b_i^{n_{1in}/4} M^- \\ 1/b_i^1 M^- \\ \vdots \\ 1/b_i^{n_{1in}/4} M^- \\ 1/b_i^1 M^+ \\ \vdots \\ 1/b_i^{n_{1in}/4} M^+ \end{pmatrix} \\
 A_c &= \begin{pmatrix} 1/a_c^1 N_T \\ \vdots \\ 1/a_c^{n_{1in}/4} N_T \\ 1/a_c^1 N_T \\ \vdots \\ 1/a_c^{n_{1in}/4} N_T \\ 1/a_c^1 N_C \\ \vdots \\ 1/a_c^{n_{1in}/4} N_C \\ 1/a_c^1 N_C \\ \vdots \\ 1/a_c^{n_{1in}/4} N_C \\ 0 \\ 0 \end{pmatrix} & B_c &= \begin{pmatrix} 1/a_c^1 N_T \\ \vdots \\ 1/a_c^{n_{1in}/4} N_T \\ 1/a_c^1 N_T \\ \vdots \\ 1/a_c^{n_{1in}/4} N_T \\ 1/a_c^1 N_C \\ \vdots \\ 1/a_c^{n_{1in}/4} N_C \\ 1/a_c^1 N_C \\ \vdots \\ 1/a_c^{n_{1in}/4} N_C \\ 1/M^+ \\ 1/M^- \end{pmatrix}
 \end{aligned}$$

With these vectors, the element \mathbf{F} matrices can be written as:

$$\mathbf{F}_{LB}^{ele} = \begin{pmatrix} A_i & B_i & \mathbf{0} \\ A_i & \mathbf{0} & B_i \end{pmatrix}, \quad \mathbf{F}_{UB}^{ele} = \begin{pmatrix} A_c & B_c & \mathbf{0} \\ A_c & \mathbf{0} & B_c \end{pmatrix}$$

Finally, the $\tilde{\mathbf{g}} \in \mathbb{R}^{\text{sum_nyiel} \times 1}$ is obtained assembling the various $\tilde{\mathbf{g}}^{\text{ele}}$. The construction of this latter vector, for a generic element of length l , differs slightly from the one seen in (4.3), since the external loads must be scaled in order to normalize the problem. Thus, we obtain

$$\tilde{\mathbf{g}}^{\text{ele}} = \frac{l^2}{8 * \text{Scale}} \mathbf{F}_{\text{LB}}^{\text{ele}} \left(0 \ 0 \ p_y^e \ p_y^e \ p_z^e \ p_z^e \right)^T$$

Once these matrices have been generated, before advancing to the LP solver, SCS checks whether the structure is or not supporting UDL. If the problem being computed bears UDL, the matrices for the lower bound computation are renamed as explained in (4.4.2), and thus we will work with matrices $\bar{\mathbf{F}}_{\text{LB}}$, $\bar{\mathbf{g}}_{\text{LB}}$ and $\tilde{\mathbf{g}} = (\mathbf{0} \ \bar{\mathbf{g}}^T)^T$ for the lower bound. Note that it is advisable to state first the presence of UDL, as for a point loaded structure this operation is useless and in fact only adds redundant equations to the LP problem.

5.4 LP Solver

The process of solving the LP problems involves first transforming the equilibrium and yield matrices for both lower and upper bounds problems into the adequate notation, see (4.4.1) and (4.4.2). The function in where this rearrangement takes place is named *LP_solver*, and it is called twice from the *main* file. Placing the variables in order to obtain the standard LP form is due to the fact that the solver that has been employed for the SCS is Matlab[®] `linsolve`. Although it is not the most efficient LP solver, it suffices for the structures computed. However, in the future the intention is to adapt the SCS to work with SDPT3 in order to enhance the solving capacities.

For both problems, once the process of solving is successfully completed (allowing a maximum number of 500 iterations) the solution vector $\mathbf{x}^T = (\lambda, \boldsymbol{\sigma}^T, \mathbf{r}^T)$ and the Lagrange multipliers $(\boldsymbol{\eta}, \boldsymbol{\nu})$ are obtained. The function also rescales λ , the multipliers in order to obtain the adequate information, which is properly stored for postprocessing. Matrices $\tilde{\mathbf{g}}$, $\hat{\mathbf{f}}$ are rescaled as well, in order to recover the input data. Finally, the program verifies that the solution meets the hypotheses of the basic approach to the limit analysis problem, i.e. equilibrium and yield is satisfied everywhere. In case not all restrictions are satisfied, the program details which equations of either equilibrium or yield are violated, and thus enables an easy overcheck.

The default options include displaying λ and the result from the verification of equilibrium and yield, either showing a positive message or the list of restrictions that are not satisfied for both equilibrium and yield. The bound gap Δ will also be displayed.

5.5 Postprocessing

SCS has been provided with several postprocessing tools in order not only to depict the final outcome of the computation, but also to support the process of adequately understanding and analysing the results obtained, a basic task that users of SCS need to be able to develop. Thus, postprocessing tools can be extremely useful whenever SCS provides weird or nonsense results, for both programmers and users. These tools all located on the *main* page, thereby it is trivial to deactivate them if desired. They are activated as a default setting.

5.5.1 Gap Evaluation

This option provides a `numel` \times 2 matrix named `gap_eval`, in where the first column shows the element contribution Δ^e to the total bound gap Δ , and the second column displays the number of element itself. The contribution is computed using the operations discussed in (4.4.4) in a function named *GapEvaluation*, which utilizes the output of the LP solver, and the result is shown using descending order, i.e. the element with the greatest contribution occupies the first row.

This enables the user to form an average idea of which sections need more specific calculations, namely using more lines to approximate the yield curve, subdividing the beam element into more subelements or even both improvements. Generally, better computations can be made by manually adjusting these latter features of the structure in the *driver*, although the initial idea is to turn this procedure into an adaptive one, which we believe it must be hand driven. This idea will be later discussed.

5.5.2 Plotting plastic hinges and displacement rates

This is a tremendously useful tool to understand how framed structures react to the external loading and how the collapse mechanism looks like. Moreover, it can lead to detect mistakes in the implementation or even in the problem modeling. The plastic hinges and displacements will be depicted only for the lower bound problem, as they represent a safe scenario and for structures with UDL the program will be able locate plastic hinges even though before subdividing the elements. The main reason to explain this latter is that the lower bound for UDL seeks for the exact moment distribution lying between the secant and the tangent distribution, and thus the combination of stresses that attain plastic flow can be in this range with no need to have a node defined. This helps detect plastic hinges in advance.

In C1, C2 type cross sections, a plastic hinge forms when the fully plastic moment is attained at one node under pure bending stress. However, if compression bending stresses take place, a plastic hinge develops if the yield curve is reached at some point. The process of locating plastic hinges is developed in *FindHinges*. The physical condition to locate plastic hinges is to find the yield inequalities that

turn to equalities. Mathematically, as it is expressed in *FindHinges*, the constraints where $|\bar{\mathbf{g}}_{\text{LB}} - \bar{\mathbf{F}}_{\text{LB}}\boldsymbol{\sigma}_{\text{LB}} + \lambda_{\text{LB}}\tilde{\mathbf{g}}| < \text{To1}$.

The output of *FindHinges* is the matrix `hinges`, with three columns and as many rows as plastic hinges. The first column represents the node that attains plastic flow, the second shows which element this node belongs to (recall that the same node can belong to several elements as they intersect in the node, and yield must not necessarily be reached at every one of these elements), and finally the third column provides information about what restriction of the node is violated. Using the same notation as before, if the yield curve at a node is approximated using \mathbf{n}_{lin} lines, the restriction's range is 1 to \mathbf{n}_{lin} , accounting the conventional counter-clockwise numeration of quadrants.

The aforementioned information is used to plot both the plastic hinges and the displacement rates of the structures, which are represented by the dual variable $\boldsymbol{\nu}$. The program used to depict this data is *PlotHinges*, and provides both the basic representation of the structure¹², nodes and bars, and the structure at collapse, i.e. with the plastic hinges (red background and black number for the node) and the displacement rate in red arrows. Note that there is the possibility of scaling the displacement rates using an amplifying factor in order to be visible, since they are often too small to be seen in the representation.

5.5.3 Plotting elements failure

Obviously, the concept of plastic hinges lacks sense when dealing with trussed structures. For this reason the routine *PlotFailure* has been designed. When computing a truss, SCS will access this routine ignoring *PlotHinges*, employing the data extracted in *FindHinges*. This easy program represents the structure depicting in red the elements that fail due to traction and in blue the elements that fail for compression. This routine also offers the option of plotting the deformed structure in the same conditions as explained before.

5.5.4 Plotting bending moment diagram

Knowing the moment stress distribution is often a useful tool for either verifying or denying the outcome of a computation. As explained before, the moment diagram will be depicted only for the lower bound problem and for the 2D frame case.

The only information that will be used is the value of $\boldsymbol{\sigma}_{\text{LB}}$. The representation is done by the function *PlotMoments*, and the procedure is as simple as uniting the moment stresses for nodes in the same element with lines. This will lead to an accurate plot for point loaded structure, but it may be weird when computing

¹²The same that can be obtained by activating the option *PlotStructure* in the *driver* of each structure

structures with UDL. Nevertheless, the moment diagram is the result of the model previously chosen, therefore better outputs can be achieved if UDL loaded beams are subdivided, since a more-like parabolic drawing will be obtained. Similarly to what happened before, it may be necessary to scale the stresses by reducing their real value with a diminishing factor, in order to improve the quality of the representation.

5.5.5 Time of computation

The speed of the computations is a decisive factor in softwares working with large-scales programs, as it may have a crucial importance to the extent of even reconsidering a different approach to the same problem with the objective of accomplishing the same task in less time. The clock is initialized at the start of every computation, and is screen-displayed after all the postprocessing tasks have been carried out. The command used to deactivate the time option is *Time*, located at the end of the *main* file.

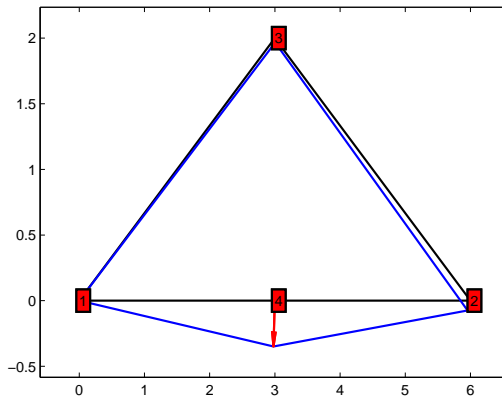


Figure 10: SCS output of *PlotHinges* function for the triangular frame

5.5.6 Example of postprocessing

Finally, the example used in (5.1.5) will be programmed and computed in order to show the aforementioned postprocessing tools. Unknown variables may be assigned the following values: $p_1 = 100 \text{ kN}$, $p_2 = 150 \text{ kN}$, $q = 50 \text{ kN/m}$, $\alpha = 60^\circ$, and the geometric variables have already been stated. Obviously, the structure is a frame, and the boolean *isnorm* is set to 1 for simplicity of computations.

Once computation is finished, in a time of 2.917s, information about the collapse load factors, the bound gap and the element contribution to the bound gap is displayed in screen, with the following values (4 decimal values)

$$\lambda_{UB} = 2.9598, \quad \lambda_{LB} = 3.5054, \quad \Delta = 0.5456$$

$$\text{gap_eval} = \begin{pmatrix} 0.2728 & 2 \\ 0.2728 & 1 \\ 0 & 4 \\ 0 & 3 \end{pmatrix}$$

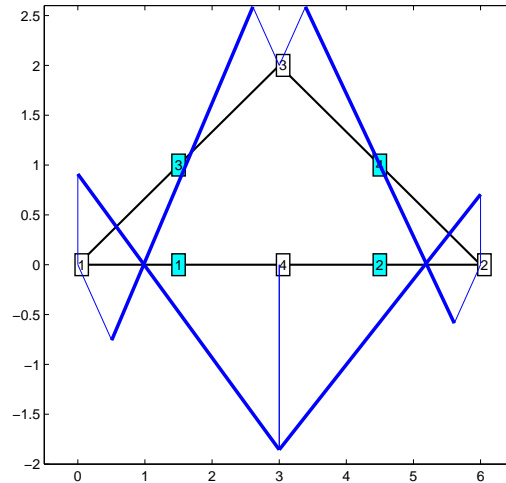


Figure 11: SCS output of *PlotMoments* function for the triangular frame

which clearly shows that UDL has a great influence on the bound gap, which could be diminished by refining the subdivision of the horizontal span. Figures 10 and 11 depict the plastic hinges location and the displacement rate of the structure (in blue with red arrows) and the bending moment distribution respectively. Although the bending moment may seem symmetrical, it actually is not, due to the nature of the loads.

5.6 Cross Section Library

The range of computations that can be carried out with the SCS highly depends on the amount of cross sections that have been considered and implemented. It has been already noted that only I/H-Sections have been studied and developed referring to the approximation the yield curve. This is not entirely true, as yield approximation for rectangular sections has also been done, but it is not further discussed as this type of cross section is rarely used in steel structures. The challenge for the future is introduce non-symmetric sections so yield criteria can be checked normally. Nevertheless, there is quite an amount of cross sections that can be utilized. The files containing the information of the cross sections are named, for instance, `CSection_HEB200`, for a HEB200, and can be found in the folder *private* of the SCS, and are directly accessed by the software whenever necessary.

The library of cross sections will suppose that the steel employed has an elastic limit of $f_y = 275 \text{ N/mm}^2 = 275000 \text{ kN/m}^2$, (the subscript is for yield) i.e. a steel characterized as **S275**. By changing data in the files appropriately, the steel used for the structure can be S235, S355 or even S460 if structural, building or economic implications must be met.

5.6.1 I/H-shaped

Obviously this is the most typical cross section employed for steel structures design, as it provides satisfying results for both economic and structural issues. The two series used in SCS so far are IPE and HEB. According to the catalogues these sections attain plastic flow, C1 or C2, almost always. The only exceptions regard the highest and widest sections under pure compression stress, when they cannot reach plastic regime. Since it is highly improbable to encounter pure compression stresses situations in actual structures, I/H-shaped structures will be considered to attain plastic flow.

The information that has to be provided to create a cross section file are both the geometric data, namely t_w , h , b , h_w , A and the plastic data, i.e. N_T , N_C , M_p^+ , M_p^- which in this case the positive and negative stresses coincide. The geometric data is directly extracted from the catalogues, and the plastic data can be easily obtained by performing the simple computations

$$\begin{aligned} N_T &= -N_C = A f_y \\ M_p^+ &= -M_p^- = W_{pz} f_y \end{aligned}$$

where W_{pz} is the plastic resistant modulus of the section around the strong axis, z , (remember Figure **3**) and can be found in the catalogues. Regard that the units mainly used in the program are kN and m , and thus both the geometrical and plastic data have to be calculated consistently.

Nevertheless, it may not always be correct to evaluate this sections using plastic data. Mostly, if the computations are desired to be made according to the elastic principles, plastic flow must not be attained. Again, it may respond to structural, safety or economical conditions, and therefore it must be considered. Setting `isnorm` to 1 implies that elastic criteria is adopted, since the yield curve cannot be approximated. In the data processing stage, the detection of `isnorm=1` automatically leads the program to add the string `'_NORM'` to the name of the file. This allows SCS to recognize which file, `CSection_HEB200` or `CSection_HEB200_NORM` (continuing with the aforementioned example), should be accessed, depending on whether calculations will be made regarding plastic or elastic regime. The `NORM` files contain the same geometric data, but the plastic data is replaced with the elastic data, N_T , N_C , M_e^+ , M_e^- . The elastic axial limit loads are the same as the plastic ones, whereas the elastic bending moments are slightly different, and can be computed as

$$M_e^+ = -M_e^- = W_{ez} f_y$$

where W_{ez} is the elastic resistant modulus of the section around the axis z , and it can be obtained from the catalogues. Obviously, the elastic magnitudes are significantly lower than the plastic ones, and thus considering calculations with the normative prescriptions will yield to lower collapse loads.

5.6.2 C4 Sections

SCS enables the users to introduce C4 sections that are not considered in the catalogues. These sections are usually used in plates or thin steel structures. It is common that slender or thin cross sections have a behavior that can be classified as C4. To enter a new cross section to the library, a file must be created and the name of the section has to end with the string '_C4' to assure the program will recognize it and consider it properly for the yield calculations. Moreover, only elastic data is required, since no computations will demand geometric data. It is advisable to provide an approximate scheme of the section and the axis orientation in order to assist the user for further employments of the file. Finally, elastic data must be provided, namely N_T , N_C , M_e^+ , M_e^- , which can be obtained performing the latter operations or by directly introducing the numerical value. This easy process widens the horizons of the SCS, since a great deal of the production of steel industries consists of beams or plates with C4 cross sections.

5.7 Utility of SCS

Finally, before studying examples to show the functioning of SCS, its utility in comparison to other softwares must be highlighted. In engineering, the majority of structures that have to be dealt with are **hyperstatic** structures, i.e. they are not statically determined. It implies that structural collapse appears when several sections attain plastic flow. Consequently, the structure turns into a mechanism, and the failure of the sections cause the structure to lose its stability and bearing capacities. Thereby, a given structure can continue to bear external load even though one or diverse plastic hinges have formed in sections where plastic limit has been reached. Namely, the structure does not collapse until the necessary number of plastic hinges to turn the structure into a mechanism have formed.

5.7.1 Linear analysis

Linear analysis, either analytical or by computer, only indicate the failure of the first section, regarding normative criteria, not the collapse of the entire structure. These methods are interesting for engineering since they are robust methods of computation, they provide reliable results, the cost of computation is reduced and they always yield to safe results.

On the other hand, there are several drawbacks in linear analysis methods. As they detect only the failure of the first section, the actual collapse load is unknown, and therefore they lead to overdimensioning of the structure. Moreover, they do not provide neither information about the security margin nor how the collapse mechanism shapes. This latter impedes to work on improvements of the structure's design. Finally, nonlinearity of the yield curve is not considered in the computations. Consequently, linear methods constitute a useful tool for having a first approach at the structure but accuracy of results and data can be improved by using nonlinear methods.

5.7.2 Nonlinear analysis

Nonlinear analysis considers the structure's real behavior and the evolution of the external loading. Methods using nonlinear analysis are realistic, as they show the actual changes the structure suffers. Furthermore, information can be obtained about the actual collapse load and the collapse mechanism.

Although nonlinear analysis provides more accurate information for an engineer, it is not always advisable to use. In general, nonlinear analysis methods tend not to be robust, and thus require an expert user. They are unefficient methods, with elevated cost and time of computation, either for the definition of the problem or for the computation. Besides from this, as a certain amount of initial data is required the accuracy of the input data lessens, lowering the reliability of the method.

5.7.3 SCS

The Structural Collapse Simulator is a computational tool of numerical calculus specifically designed for the robust, efficient and reliable evaluation of structures. SCS employs linear programming techniques to find the unique solution of the problem, thus robustness is assured. Moreover, nonexpert users can easily perform computations with the SCS. It is clearly an efficient software, as its basis in LP enables the use of several optimization packages in order to improve the calculations. Finally, it constitutes a reliable method since it provides a bounded solution, indicating the minimum or safe value of the load factor (lower bound) and the precision of the computations (bound gap). The bounding can be reduced arbitrarily by the user. SCS satisfyingly combines the characteristics of the linear and the nonlinear analysis methods, achieving interesting results as can be seen in next section.

6 Numerical examples

6.1 Simply supported beam

The simply supported beam of one span subject to a uniform distributed load along the span is perhaps the simplest, yet one of the most useful examples to both show

how SCS works and to explore the limits of this model, in order to extract conclusions that can be extrapolated for more complex structures. Thereby it is highly interesting to study this case thoroughly. Geometric data and load conditions for the simply supported beam are shown in Figure 12, along with the bending moment and shear stress distribution, which can be found analytically. Peak values are also depicted.

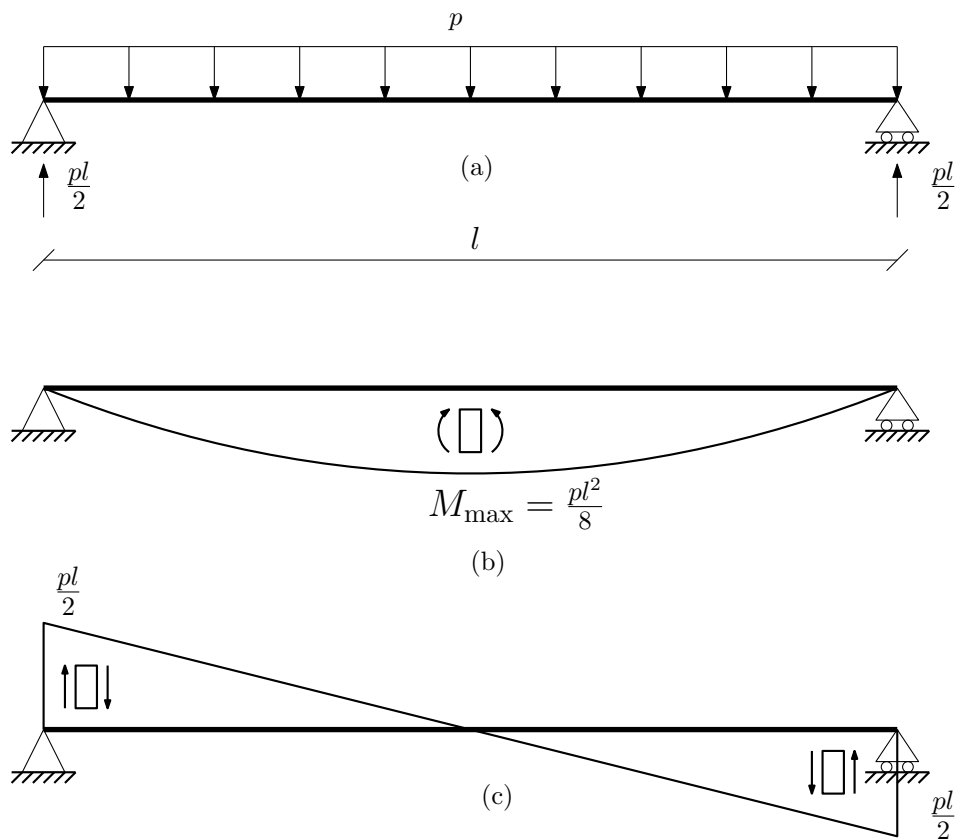


Figure 12: Simply supported beam of length l : (a) Loading and reactions, (b) Bending moment, (c) Shear stress

For this case, it will only be considered influence from the UDL. Hence, isnorm will be set to 1 in the *drivers*, and therefore the bound gap will be caused entirely from the lower bound problem's secant and tangent distribution of the moment diagram, remember (4.3). Results will shed some light on how SCS works precisely. This process will be done by continuously subdividing the main span, and thus providing a more descriptive situation with each subdivision. The reason for omitting the yield check is the very nature of the structure, since the bound gap depends entirely on how the structure is modeled, and using more lines to describe the yield curve reveals useless once it has been computed.

The simply supported beam is an isostatic frame, i.e. unknown geometric reac-

tions can be found by setting the equilibrium equations, thus when a single plastic hinge forms the structure turns into a mechanism, which is the collapse mechanism. Rotation is allowed at the supports, and hence the only plastic hinge will appear in the middle of the span. Moreover, the formation of the plastic hinge will be due to attaining the fully plastic moment, since no compression or traction axial stress is present. This implies that using the technique of the kinematic theorem, applying the virtual work theorem, the exact collapse load factor can be found. Using the kinematic theorem, both the collapse load and the collapse load factor are found, with value

$$p_c = \frac{8M_p^+}{l^2}, \quad \lambda = \frac{p_c}{p} = \frac{8M_p^+}{pl^2}$$

For the following examples, it has been used $l = 4\text{ m}$, $p = 100\text{ kN/m}$ and a cross section HEB300, with a plastic moment of $M_p^+ = 513.975\text{ kNm}$. Performing easy computations, the collapse load for the structure results $p_c = 256.9875\text{ kN/m}$, which implies that the exact value of the load factor is $\lambda = 2.569875$.

6.1.1 Simply supported beam with no subdivision

In this case, the most basic one, the UDL with value p will be replaced by two point loads of $\frac{pl}{2}$ located in the nodes, see Figure 13. It is obvious that, from the nature of the loading, the beam supports no bending moment anywhere. Computation of this structure by SCS provides the following results:

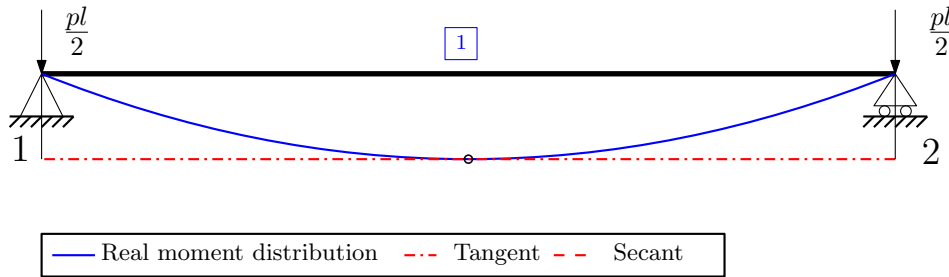


Figure 13: Simply supported beam with no span subdivision

- $\lambda_{LB} = 2.569874999995355$
- $\lambda_{UB} = \infty$
- Time = 2.608 s

Note that the lower bound value is the precise one, whereas the `linsolve` has been unable to converge successfully, since the primal problem is unbounded, and thus the dual infeasible. Both results could have been predicted in advance.

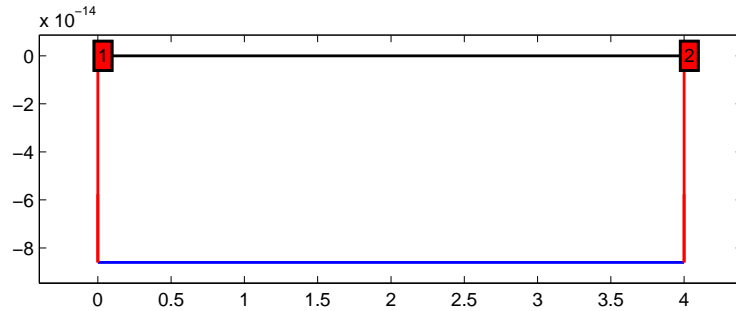


Figure 14: Plastic hinge detection by SCS for the simply supported beam with no subdivision

The lower bound problem accounts for both the secant and the tangent distribution for the bending moment diagram, the tangent being the secant translated $\frac{pl^2}{8}$. Since the exact moment distribution lies somewhere between these latter two, SCS is able to recognize the maximum moment, where the plastic hinge is more likely to be formed, and study it. Moreover, in this case the maximum moment lies *exactly* in the tangent moment distribution, regard the circle in Figure 13, since $\frac{pl^2}{8}$ is this maximum moment. This implies that SCS detects the formation of the plastic hinge, but as no node is defined in the middle of the span, it places the hinges on the supports, see Figure 14, where the software is able to check yield and equilibrium. This automatically leads to finding the exact collapse load, since the detection of the plastic hinge formation has been successful.

On the other hand, it is reasonable that no solution can be found for the upper bound problem. This problem only concerns the secant moment distribution, which in this case is equal to 0, since no loads are applied on the span. Thus, if loads are only applied at the supports, those can absorb the loads indefinitely, because the beam would not have to bear any stresses. Therefore there is no upper collapse load factor. In fact, the collapse load is ∞ , as by no means the beam will attain plastic flow if no stresses are present.

6.1.2 Simply supported beam with one subdivision

This is the first refinement, where an extra node has been placed in the middle of the span. Load conditions and bending moment diagram are depicted in Figure 15. Results for the computation are

- $\lambda_{LB} = 2.05589999999999$
- $\lambda_{UB} = 2.56987499990634$
- $\Delta = 0.5139749999064$

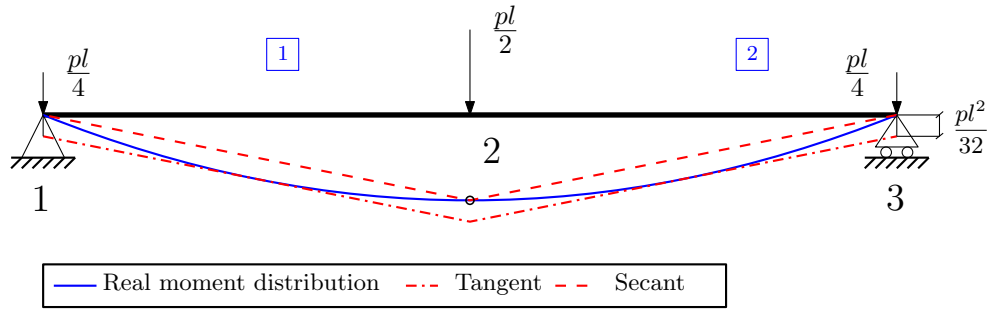


Figure 15: Bending moment distribution for the simply supported beam with one span subdivision

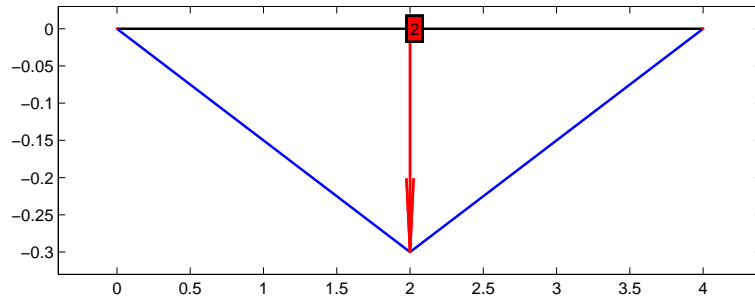


Figure 16: Plastic hinge detection by SCS for the simply supported beam with one span subdivision

- Time = 2.438 s

Direct observation of these values leads to think that there was a mistake during the computation, since the lower bound, which was precisely computed before the refinement, has now diminished and represents an incorrect value. Moreover, the upper bound, which was not even found before due to the nature of the model, has now acquired the exact value of the collapse load factor. Nevertheless, in order to understand the phenomenon happening here, Figure 15 must be carefully studied. The secant moment distribution (the dashed one) is the moment distribution originated by the point loads, i.e. is the real moment distribution for the model, since no UDL are present (in the model). In the middle of the span the moment distribution reaches the value $\frac{pl^2}{8}$, and as a node is defined there, the plastic hinge is exactly located (see Figure 16) and the load factor obtained will be the real one.

Oppositely to the upper bound, the lower bound has not shown the desired behavior. This situation is due to the fact that the tangent distribution had located the exact position of the plastic hinge in the first iteration, and when a refinement is made the new tangent distribution lies below the position of the plastic hinge, thus it is not found precisely, as can be seen in Figure 15. Both distributions are separated

$\frac{p(l/2)^2}{8} = \frac{pl^2}{32}$, and consequently the tangent distribution is unable to provide the exact result. This can be stated as one of the handicaps of SCS, although it can easily be sort out as will be seen now. Note that the fact that this more complex model took less to compute than the first one. Regarding the nature of the first case, where the upper bound LP problem did not converge, explains this condition.

6.1.3 Simply supported beam with two subdivisions

After a second refinement, with the load conditions and moment distributions depicted in Figure 17, SCS gives the following results

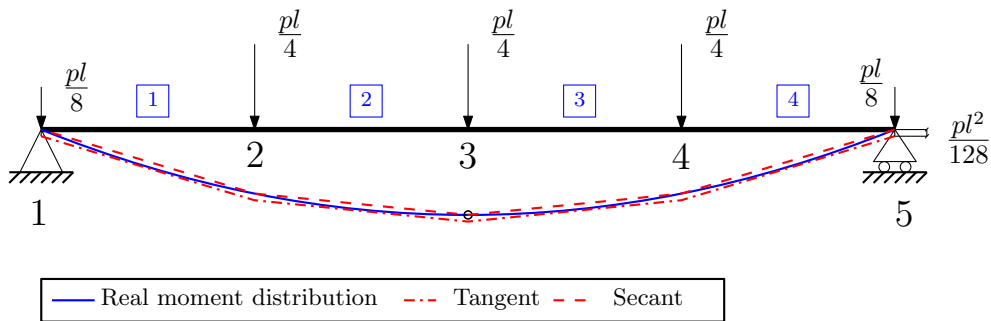


Figure 17: Bending moment distribution for the simply supported beam with two span subdivisions

- $\lambda_{LB} = 2.41870588235294$
- $\lambda_{UB} = 2.569875000000000$
- $\Delta = 0.1511691176471$
- Time = 1.737 s

This results are consistent with what was commented before. The upper bound remains equal, since the node where the plastic hinge forms is located. Comparing the results between this computation and the latter, the lower bound gradually approximates to the exact collapse load. This is due to the fact that the tangent distribution is closer to the secant every iteration, in fact it approximates quadratically, i.e. with the square of the beam subdivision length. This will be assured using the next refinement. The plastic hinge plot will not be shown as it does not differ from the one in Figure 16.

6.1.4 Simply supported beam with three subdivisions

This last section will be employed to evaluate the convergence of the boundaries to the exact load factor. In Figure 18 the secant and tangent moment distributions are not depicted, since they are indistinguishable from the bending moment distribution in blue. Results from the computation are

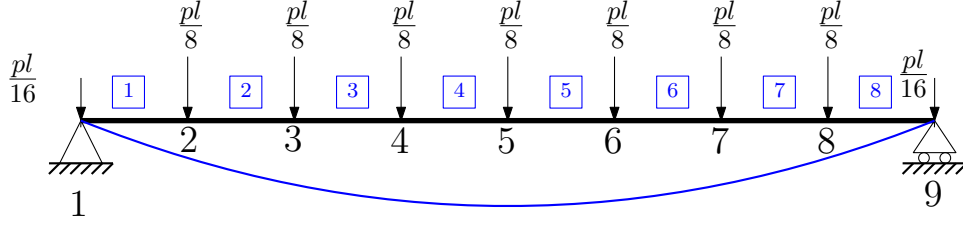


Figure 18: Bending moment distribution for the simply supported beam with three span subdivisions

Number of elements	λ_{LB}	λ_{UB}	Δ
1	2.56987499995355	∞	∞
2	2.05589999999999	2.56987499990634	0.5139749999064
4	2.41870588235294	2.56987500000000	0.1511691176471
8	2.53033846153844	2.56987499999999	0.0395365384616

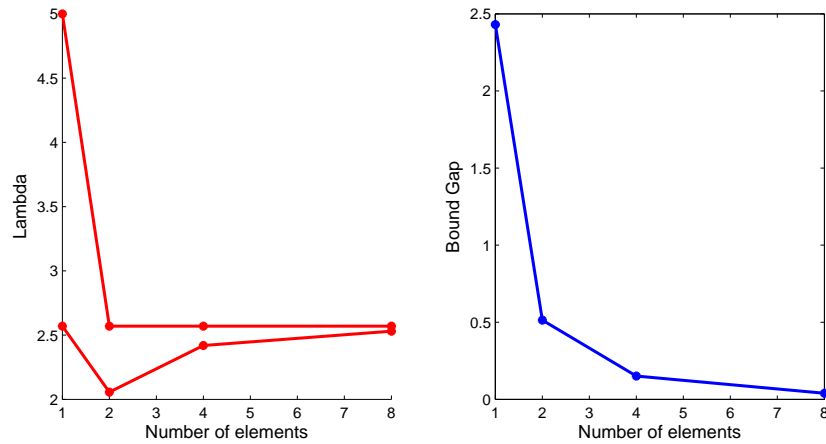
Table 2: Results from SCS computations of the simply supported beam

- $\lambda_{LB} = 2.53033846153844$
- $\lambda_{UB} = 2.56987499999999$
- $\Delta = 0.0395365384616$
- Time = 1.365 s

The results are as expected, since refining the model leads to a tangent distribution practically equal to the secant, and this results in the lower bound approximating to the upper bound. It is also interesting to check the contribution of each element to the total gap. The matrix `gap_eval` for this case is

$$\text{gap_eval} = \begin{pmatrix} 0.01976827100826 & 4 \\ 0.01976826745325 & 5 \\ 0 & 8 \\ 0 & 1 \\ 0 & 7 \\ 0 & 2 \\ 0 & 3 \\ 0 & 6 \end{pmatrix}$$

clearly the two central elements contribute the most to the gap, since moment in that region is the greatest. An adaptive procedure would detect this factor and continue to subdivide only elements 4 and 5 in order to reduce the gap. Obviously, this procedure can also be made by the user himself. The results from all computations are represented in Table 2.

Figure 19: Convergence for λ and Δ versus the number of elements

Finally, before analyzing the influence of the yield curve, it is useful to represent these results to see the convergence of the bounds and the evolution of the gap with the number of elements employed in the model. It should be stated that the first value for λ_{UB} has been set to 5, since representing it with the actual value would certainly make the graphic useless. As seen in Figure 19, the bound gap clearly decreases with the square of the subdivision length. With this example fully understood a global idea of how SCS works it achieved, and thus we can advance to more complex structures.

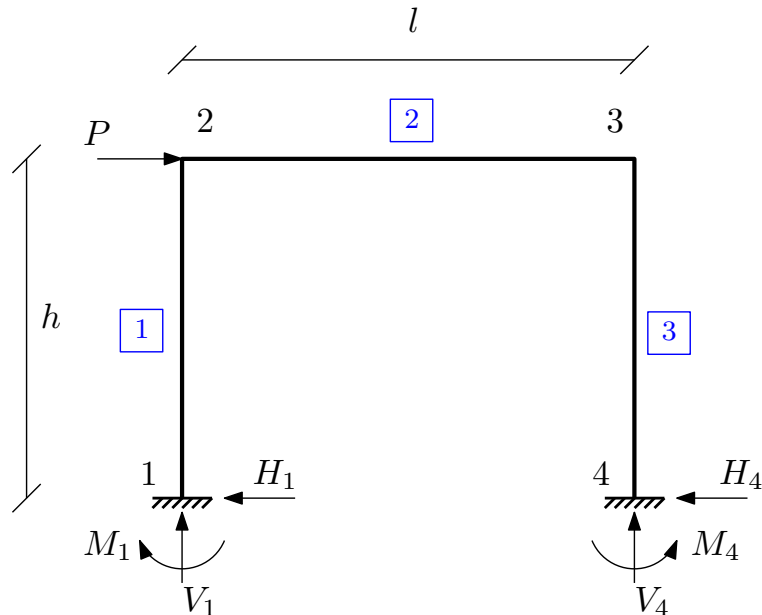


Figure 20: Representation of the embedded frame with point load

6.2 Embedded frame with point load

Once clarified how SCS bounds the exact collapse load from a given load distribution, it is worth employing it on a simple but representative case. The frame depicted schematically in Figure 20 is a useful example to show the benefits of SCS in comparison to standard linear calculations explained in section (5.7.1).

This frame is loaded horizontally in node 2. The beams employed in this structure have IPE360 cross sections. At the bases horizontal, vertical deflection and rotation are impeded. Reaction forces can be found explicitly

$$H_1 = H_4 = \frac{P}{2} \quad V_1 = V_4 = \frac{3hPk}{l(6k+1)}$$

$$-M_1 = M_4 = \frac{Ph}{2} \frac{3k+1}{6k+1}$$

where $k = h/l$. For the computations, let the spans be $h = 3m$, $l = 4m$. According to linear analysis, the first node to attain plastic flow is **node 1**. The load to form this first plastic hinge can also be found explicitly.

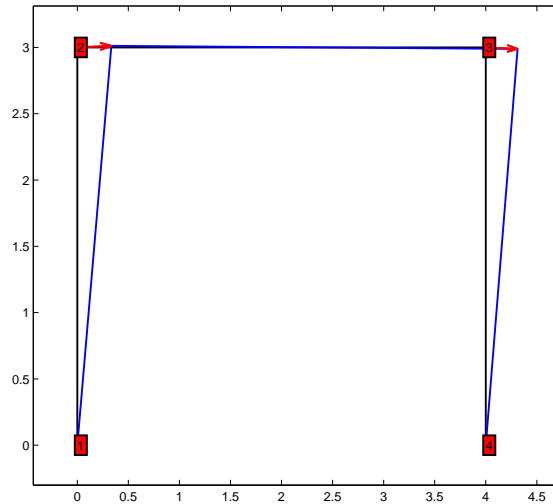


Figure 21: SCS detection of the plastic hinges to form the collapse mechanism

Regard that for linear calculations the yield curve of node 1 is a polygon of 4 sides in the $\left(\frac{M}{M_p}, \frac{N}{N_p}\right)$ plane. Using this condition and the fact that in elastic flow the stresses at node 1 are $N = V_1$ and $M_z^1 = M_1$, the collapse load for linear analysis P_{1in} can be found by imposing the combination of stresses to be on the yield curve. Performing an easy computation, the value found is $P_{1in} = 301.52 kN$. However,

this frame is clearly a hyperstatic structure, and thus it requires more than one plastic hinge to form a mechanism. Consequently, the load obtained is a safe value of the actual one, but it can be improved using SCS.

SCS effectively states that there needs to form a plastic hinge at every node in order to reach structure collapse, as can be seen in Figure 21. Considering the yield curve as a 4-edge polygon (using the normative) SCS obtains a value of the collapse load $P_{\text{norm}} = 345.53 \text{ kN}$. The easy calculation $P_{\text{norm}}/P_{\text{lin}} = 1.15$ shows that the studied frame is able to bear 15% more load than the one made with linear analysis.

Nonetheless, this result can be even more improved by using the actual yield curve. For the computations the yield curve is approximated with 40 lines per node, i.e. 10 per quadrant. This leads to the most realistic result that can be obtained. Moreover, in this case the lower and upper bound do not coincide, and the value used for the comparisons will be the lower bound, since it assures to be on the safe side of the calculations. Regarding this information, SCS provides a collapse load of $P_{\text{real}} = 351.68 \text{ kN}$. Repeating the calculus, $P_{\text{real}}/P_{\text{lin}} = 1.17$. Hence, it can be stated that the SCS enables the user to know that the structure can support 17% more load than the linear analysis case, besides providing information about the collapse mechanism. This easy example illustrates the power and the utility of SCS in comparison to linear analysis methods used in engineering, such as SAP2000[®].

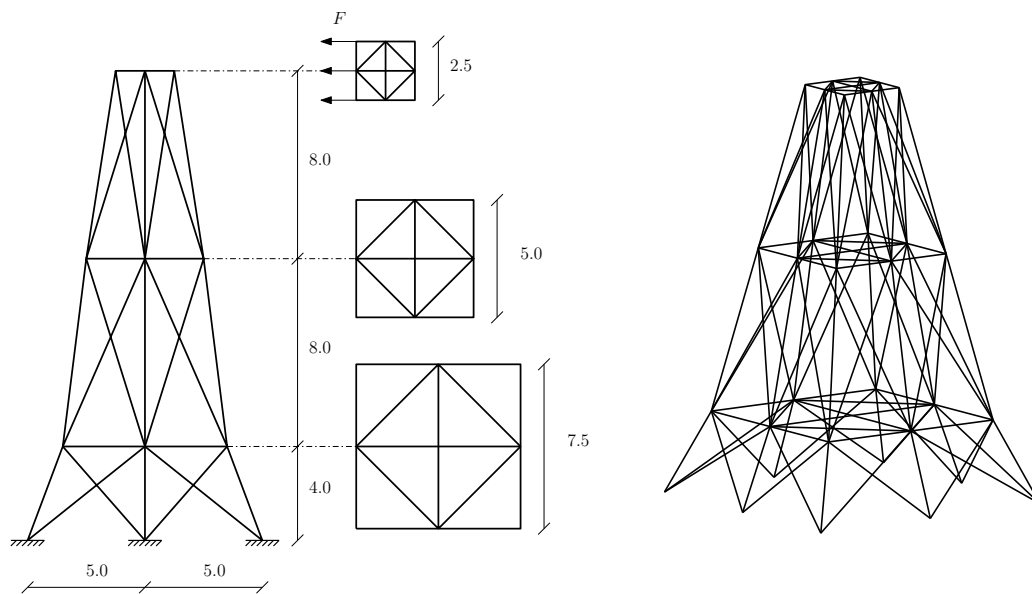


Figure 22: Schematic representation and 3D view of the steel tower

6.3 Steel tower

Another example to examine the advantages of employing SCS in the computation of structures in comparison to linear analysis methods is provided. The structure analyzed here is a trussed 3D tower. Since yield criteria is not yet available for 3D frames, trusses are the only example that can be computed in 3D.

The details of the truss are the ones in Figure 22. Loads are horizontal, and the beams used in the trussed structure have square $100 \times 100 \text{ mm}$ cross sections. The plastic values are considered to be $N_T = -2N_C$, where $N_T = 5 \cdot 10^5 \text{ kN}$. Being a trussed structure implies that the failure of a section occurs whenever the axial stress reaches the plastic limit. Hence, the normative criteria equals the real behavior of the yield curve.

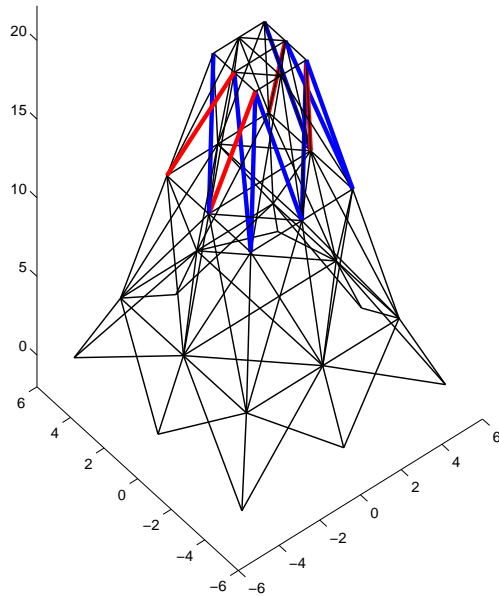


Figure 23: Collapse mechanism. In red elements that fail for traction and in blue for compression

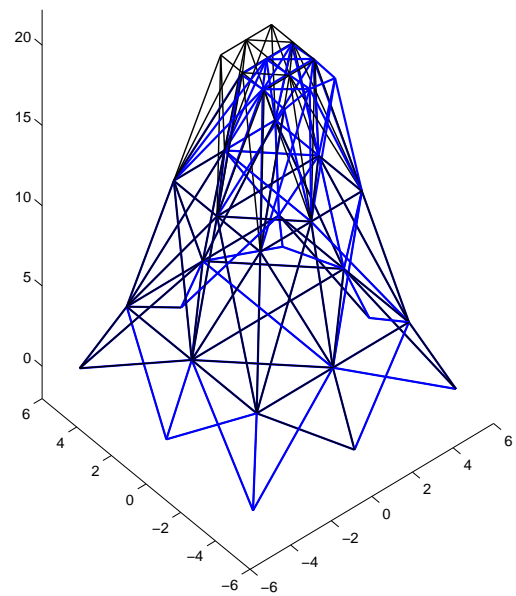


Figure 24: Amplified deformed structure in blue.

Linear analysis detects the first element to exceed the elastic limit (either traction or compression). In this case, due to problem symmetries various elements attain plastic flow at the same time. The collapse load found using SAAP2000[®] is

$F_{\text{lin}} = 216440 \text{ kN}$. Nevertheless, the result obtained is not the true mechanism of collapse, as the structure can bear more loading without collapsing.

However, computing this trussed tower with SCS leads to more realistic results. A distribution of elements that generate the collapse mechanism can be obtained. In Figure 23 the elements that fail are depicted, in red the elements that fail for traction and in blue for compression. It can be observed in Figure 24 that only the upper part of the structure fails, and therefore only this part collapses.

The collapse load obtained by SCS is $F_{\text{real}} = 300075.7 \text{ kN}$, which can be related to the one obtained with the linear analysis similarly to what done before. Performing the easy computation $F_{\text{real}}/F_{\text{lin}} = 1.4$ enables to see that SCS detects that structure can bear 40% more loading, besides providing the actual collapse mechanism.

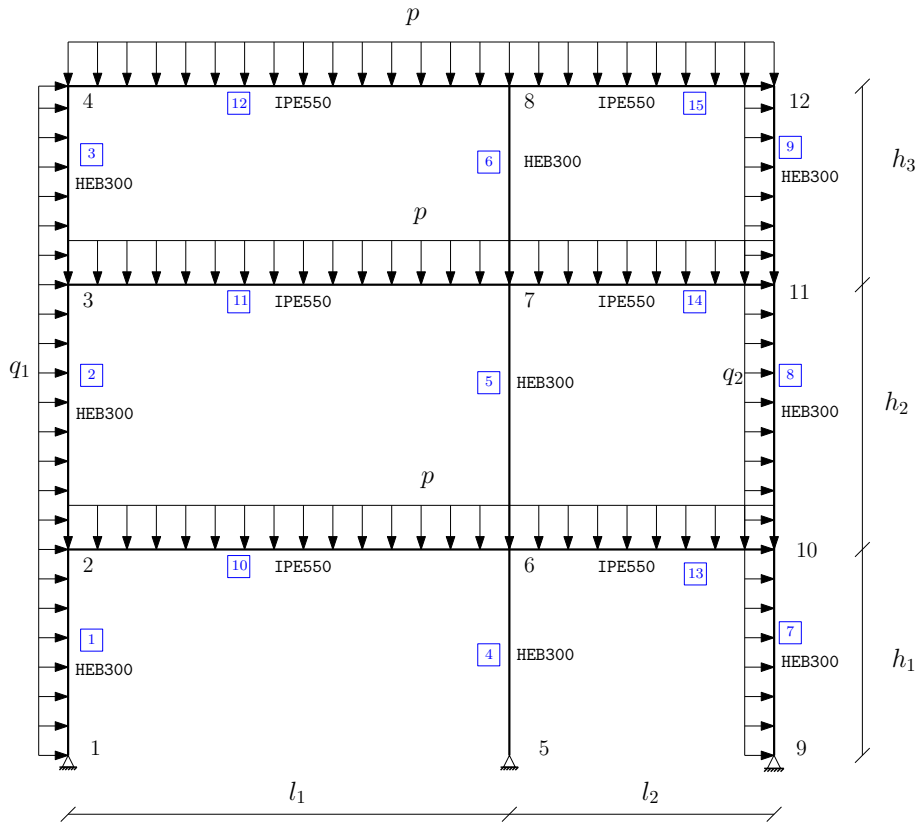


Figure 25: Representation of the three-storey frame

6.4 Model of a 2D three-storey building

Finally, SCS is employed to compute a complex three-storey frame that models a real structure. This is the case that will gauge the power of SCS. Moreover, in

a theoretical basis it will be of great interest, since influence from both the yield curve and the UDL will be determinant. A schematic representation of the frame is in Figure 25. Geometric data is $l_1 = 7.5 m$, $l_2 = 4.5 m$ for horizontal spans and $h_1 = 3.55 m$, $h_2 = 4.5 m$, $h_3 = 3.3 m$ for vertical spans. The vertical loads correspond to use loads, with a value of $p = 65 kN/m$, whereas the horizontal ones are wind loads, with values of $q_1 = 5 kN/m$, $q_2 = 3 kN/m$. All loads have already been increased by the appropriate load increase coefficients, which can be found in the normative of steel structure.

Similarly as what was done with the simply supported beam, this frame will be studied by continuously subdividing the elements bearing UDL, and at each step analysis of the convergence will be carried out varying the number of lines used to approximate the yield curve, and thus a more thorough analysis will be possible. The number of lines considered will be in the range of 4-20 per node, that is one to five lines per quadrant.

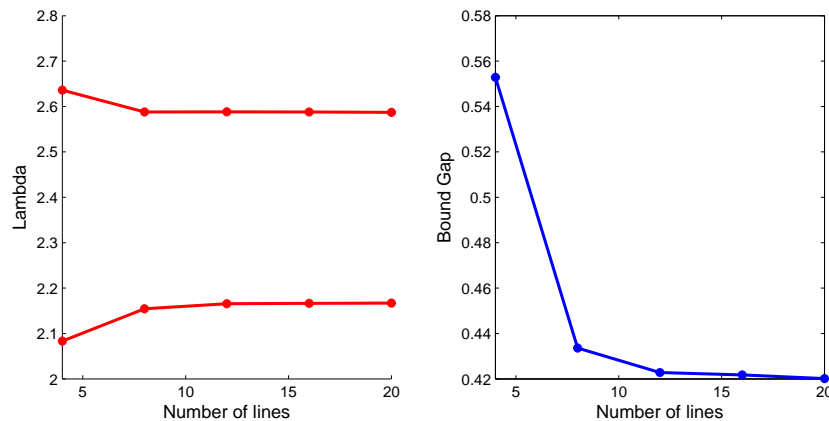


Figure 26: Convergence of λ and Δ versus the number of lines for the frame with no span subdivision

6.4.1 Frame with no span subdivision

This is the simplest and the least interesting case, since no loads are applied in the middle of the span, and thus the model is not representative. Computation with SCS gives the results in Table 3.

Observing the results, it should be detached that both the upper and the lower bound stabilize even if the number of lines increases. Obviously, this indicates that improving the bound gap is only possible by refining the UDL elements, since refining the yield curve has proved useless. These latter results are shown in Figure 26.

Number of lines	λ_{LB}	λ_{UB}	Δ
4	2.08329879539149	2.63611468706516	0.55281589167367
8	2.15443243573331	2.58805072465873	0.43361828892542
12	2.16546069016570	2.58825121363745	0.42279052347175
16	2.16628821509802	2.58804876103035	0.42176054593233
20	2.16702838902643	2.58715904701065	0.42013065798422

Table 3: Results from SCS computations for the frame with no span subdivision

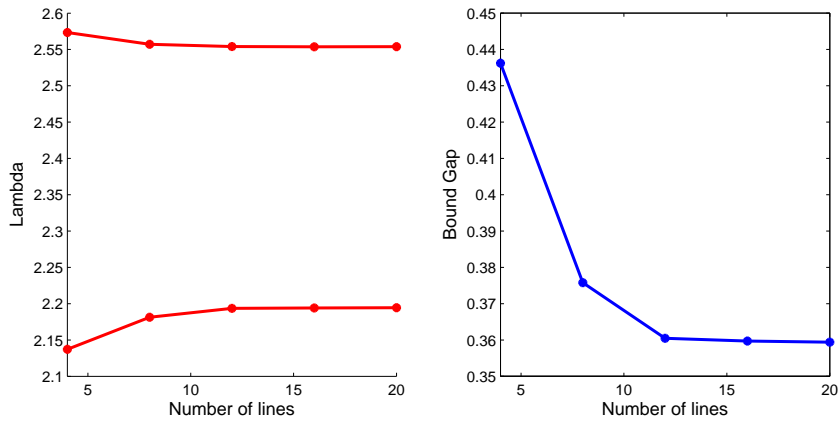


Figure 27: Convergence of λ and Δ versus the number of lines for the frame with one span subdivision

6.4.2 Frame with one span subdivision

When loads are applied at the middle of the span, the model starts to be representative. Results are grouped in Table 4.

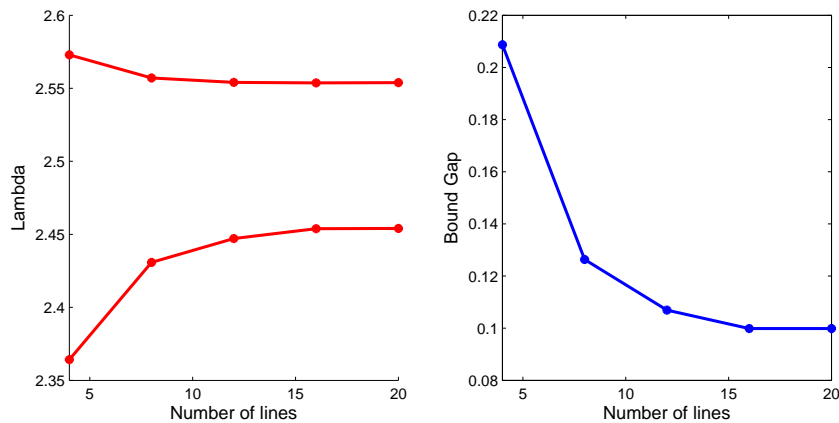
These results are depicted in Figure 27. The same tendency as before is observed. Note that the gap has reduced from 0.42 to 0.36 roughly. In order to achieve more precise computations, more subdivisions need to be done.

Number of lines	λ_{LB}	λ_{UB}	Δ
4	2.13714558994884	2.57338126475674	0.43623567480790
8	2.18133497740381	2.55711600102536	0.37578102362155
12	2.19359181218358	2.55406234816166	0.36047053597808
16	2.19403310701907	2.55375131616303	0.35971820914396
20	2.19445625797868	2.55386094581217	0.35940468783349

Table 4: Results from SCS computations for the frame with one span subdivision

Number of lines	λ_{LB}	λ_{UB}	Δ
4	2.36422675495481	2.57291606326957	0.20868930831476
8	2.43075118005173	2.55711600074626	0.12636482069453
12	2.44713656862145	2.55406234812445	0.10692577950300
16	2.45386816567384	2.55375131615935	0.09988315048551
20	2.45400067027022	2.55386094529599	0.09986027502577

Table 5: Results from SCS computations for the frame with two span subdivisions

Figure 28: Convergence of λ and Δ versus the number of lines for the frame with two span subdivisions

6.4.3 Frame with two span subdivisions

Computing the frame with this refinement provides the results in Table 5. This refinement has been successful, since the gap has decreased from 0.36 to 0.1. In this latter the influence of the yield curve in the computations is noticeable, especially in the lower bound (see Figure 28). For this model, it is worth regarding the bending moment distribution. Figure 29 represents the case with 20 lines per node, since it is the most accurate one.

It should be highlighted that the upper bound remains stagnant with the increasing number of lines to describe yield and surprisingly with the refinements. Thereby, it leads to think that the bound gap is in a major part due to the UDL, as it is only considered in the lower bound problem via the tangent distribution. It is helpful to use the matrix `gap_eval` in order to analyze the element contribution to the gap. Only the first 8 rows of the matrix are shown, as they are the most relevant. The other values have order of 10^{-4} , and therefore no significant improvement would be obtained if refining those elements.

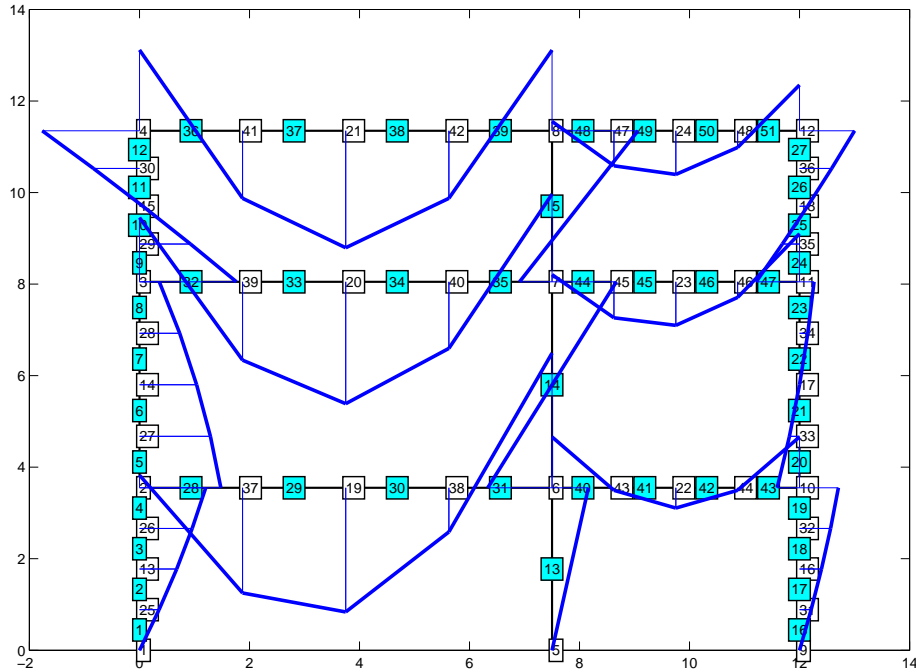


Figure 29: SCS output for moment distribution for the frame with two span subdivisions

$$\text{gap_eval} = \begin{pmatrix} 0.04324455910328 & 29 \\ 0.04324454608226 & 30 \\ 0.00184365778141 & 34 \\ 0.00184365753955 & 33 \\ 0.00178612711256 & 14 \\ 0.00175520351402 & 38 \\ 0.00175520200048 & 37 \\ 0.00121883914062 & 5 \end{pmatrix}$$

Observing the bending moment distribution in Figure 29 and it is clear that the greatest moments occur at the elements which contribute the most to the bound gap. The following step is to manually refine these elements (except 14) and analyze the convergence of the load factor and the bound gap.

6.4.4 Frame with manual subdivisions

The aforementioned elements are refined by adding an extra node. Results from the computation are shown in Table 6.

The bound gap has drastically decreased. Obviously, before performing any computation the user must establish the precision of the calculations, i.e. the number of

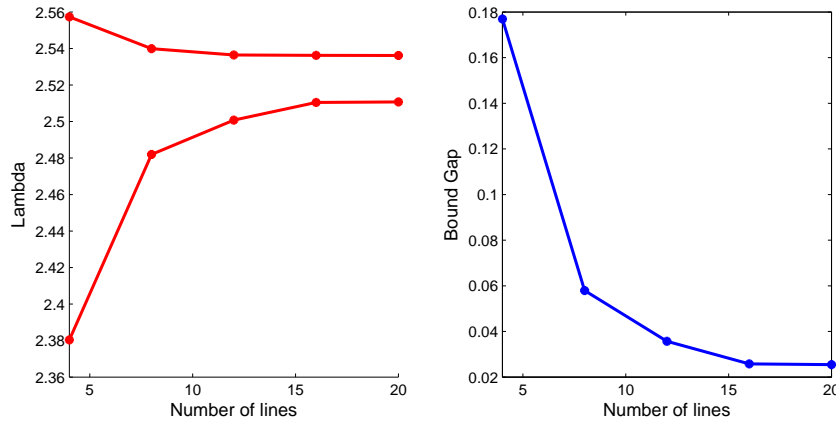


Figure 30: Convergence of λ and Δ versus the number of lines for the frame with an extra manual span subdivisions

Number of lines	λ_{LB}	λ_{UB}	Δ
4	2.38039642126137	2.55732188258492	0.17692546132355
8	2.48194860612320	2.53987728373308	0.05792867760988
12	2.50080180806465	2.53646896464822	0.03566715658357
16	2.51046989114661	2.53623332534645	0.02576343419984
20	2.51075375319028	2.53618129917413	0.02542754598385

Table 6: Results from SCS computations for the frame with an extra manual subdivision

significant digits to ensure. Since the purpose of this project is to demonstrate how SCS works, a gap of 0.025 is considered satisfying. It is noticeable that with this manual subdivision of only 7 elements a greater improvement has been achieved. Finally, the plastic hinge location has been depicted in Figure **31**, and the convergence is shown in Figure **30**.

6.4.5 Study of convergence

Once the separate cases have been detailed, it is useful to represent the results of convergence obtained in a single graphic, so as to analyze the enhancements achieved with the refinements and to extract conclusions on the effectiveness of SCS. The graphic of convergence is in Figure **32**. For the legend the notation S_i , $i = 0, 1, 2, 4$ has been used, S for Subdivision and the subscript for the number of subdivisions applied to the elements.

The influence of both the yield curve approximation and the subdivision of elements bearing UDL can be clearly appreciated with the representation in Figure **32**. The initial model, being the frame with no subdivisions, constitutes a poor representation of the structure, as can be seen after the refinements. Consequently,

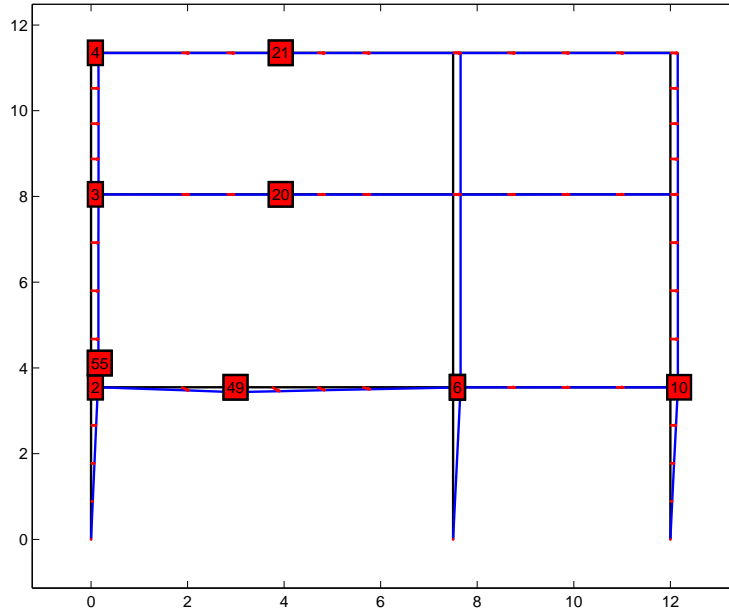


Figure 31: Plastic hinge detection by SCS for the frame with an extra manual subdivision

the gap is unacceptable, even using more lines to wrap the yield curve. This implies that is impossible to achieve satisfying results by only using the yield curve approximation, and therefore a refinement is needed.

The first two refinements induce significant improvements to the model. It is worth mentioning that the upper bound for the frame with one and two span subdivisions (blue and black respectively in Figure 32) coincide, whereas the lower bounds clearly differ. The first refinement is still unacceptable, since it yields a bound gap of 0.36 approximately. In fact, it is nonsense to talk about convergence in this case. Nevertheless, the second refinement produces acceptable results. The bound gap remains around 0.1, but the same tendency is observed. Even using more lines to describe the yield curve, the gap remains stagnant. Thereby, it is sensible to state that refining the model tends to produce better results than using more lines to wrap the yield curve.

Finally, the third refinement is the most accurate one. By manually subdividing the elements that contribute the most to the gap, satisfying results can be obtained. The gap has diminished to 0.025, which can be considered acceptable, and convergence of the lower and the upper bound can be appreciated. The main reason for this great improvement can be found in Figure 31. The latter division of elements has enabled SCS to locate the exact location of the plastic hinges that induce the collapse mechanism. Consequently, better results are obtained. Moreover, the in-

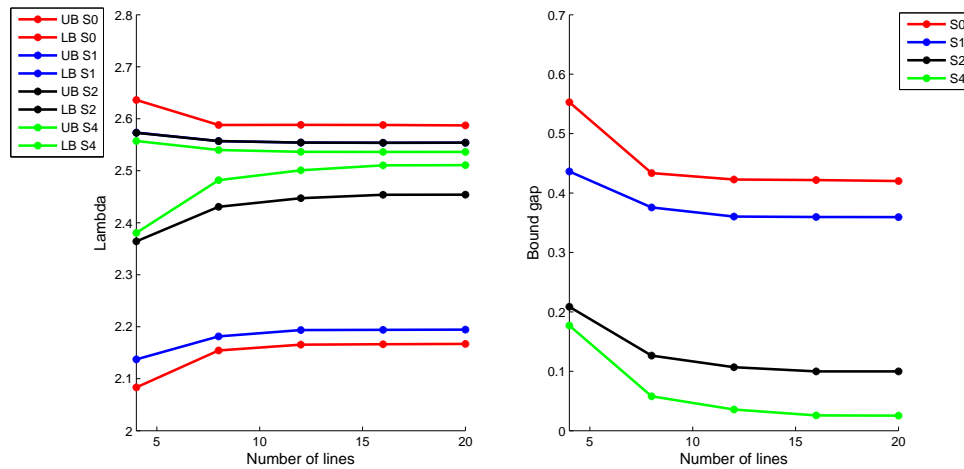


Figure 32: Convergence of λ and Δ versus the number of lines for the frame

fluence of the yield curve is decisive in the concept of convergence.

Since refining the model has proven an interesting idea of achieving satisfying results, it is sensible to focus on it for improving the software. The procedure of subdividing spans with UDL can be thought as an adaptive one. The main idea for the implementation of adaptivity consists in defining the criteria of choosing which and how many elements to refine and the accuracy of the computation desired. The first one is as simple as refining the elements that contribute the most to the gap, which can be found using the matrix `gap_eval`, using a criteria of the 30% greatest contribution, for instance. The second one depends on the user. This idea is being developed at the moment, and further details can be found in 8.4.

To summarize, SCS performs precise computations which enable the user to know a lower and an upper bound to the exact collapse load factor. The procedure has proved successful by analyzing the model of a three-storey building and the influence of UDL and the yield curve in the convergence of the lower and upper bound. The analysis that has been carried out shows that in order to achieve acceptable results, both techniques must be employed simultaneously. Subdividing the beam elements bearing UDL is necessary in order to describe the frame more accurately, ensuring to detect the collapse mechanism and thus reducing the bound gap. The simply supported beam and the frame clearly depict this situation. Furthermore, approximating the yield curve using more lines guarantees convergence of the upper and lower bounds, but is useless without the element subdivision. The proof of this latter is the fact that the bounds remain stagnant despite using more lines, therefore the need of subdividing the UDL elements arises. Consequently, the combination of these two techniques is necessary to obtain satisfying results, ensuring success of SCS.

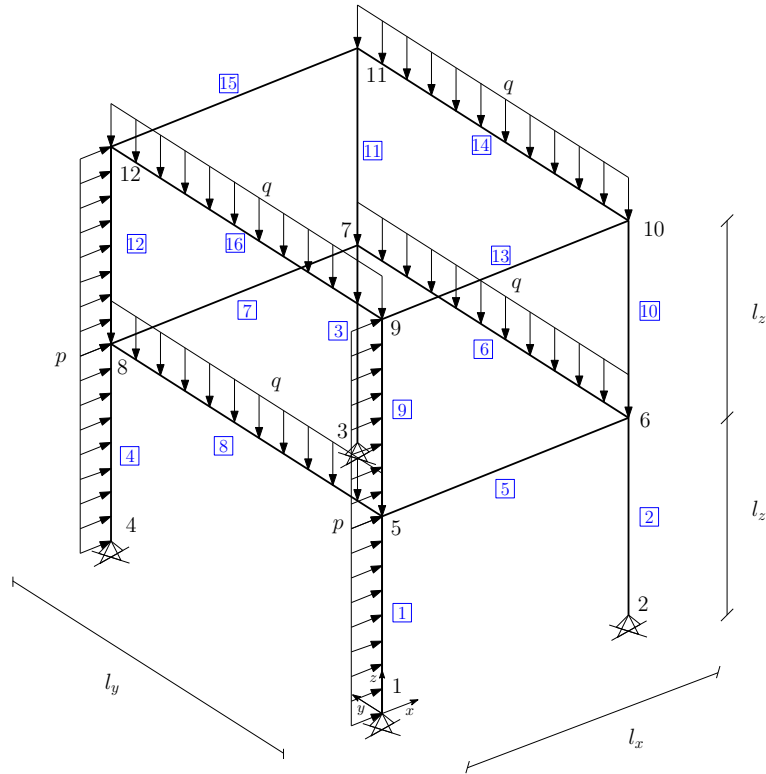


Figure 33: Representation of the 3D two-storey frame

6.5 Model of a 3D two-storey building

The last numerical example that is given consists of the framed structure in 3D that can be seen in Figure 33. Since approximation of the yield surface for 3D beams has not been implemented yet, this case cannot be analysed in depth as the earlier cases. Nonetheless, it suffices to show how SCS can compute 3D frame structures using the normative criteria for yield. Therefore, it is not presented as a closed example, but as an introduction for further improvements of SCS.

Number of subdivisions	λ_{LB}	λ_{UB}	Δ
0	1.2564283508409	1.3470217865156	0.090593435674
1	1.3231703376526	1.3470217865629	0.023851448910
2	1.3409786711463	1.3470217865604	0.006043115414
3	1.3455059072096	1.3470217865643	0.001515879354

Table 7: Results from SCS computations of the 3D frame

The geometric data is $l_x = 3m, l_y = l_z = 4m$. The vertical loads correspond to

use loads, with a value of $q = 50kN/m$, whereas the horizontal ones are wind loads, with a value of $p = 15kN/m$. All elements are designed using the IPE360 beam. The analysis in this case is much simpler, as convergence can only be observed by subdividing elements bearing UDL. The results obtained in the computations are collected in Table 7.

The convergence and the plastic hinge distribution are depicted in Figures 34 and 35 respectively.

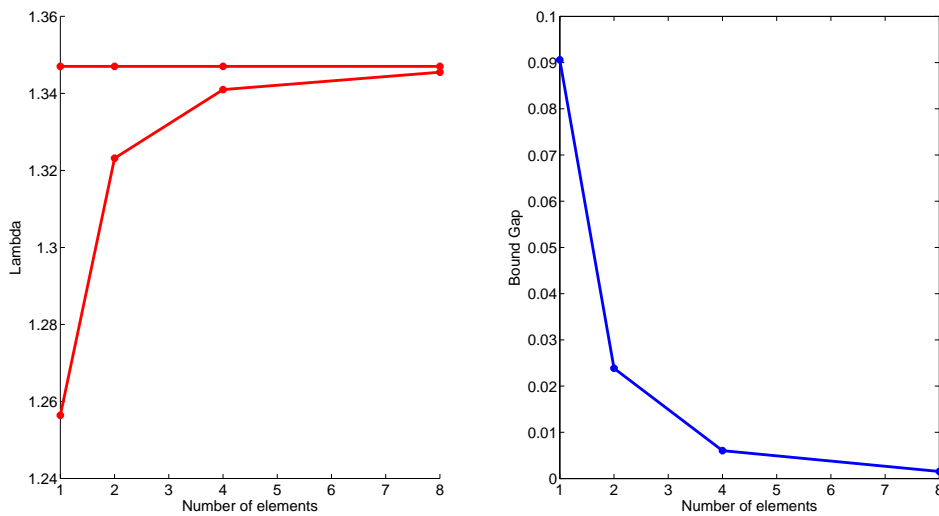


Figure 34: Convergence of λ and Δ versus the number of subdivisions

Regarding the results and the figures, it can be stated that despite working with normative criteria the results obtained from the computations are really good, achieving a gap of only 0.0015. This example reinforces the idea that despite approximating the yield surface is important, working with a faithful representation of the actual situation always leads to better results. It should also be noted that this latter case is only an example, since it is not normal to have wind loads with such great values. It may be the main reason for having a translational collapse mechanism instead of another one based on gravity loads. Nonetheless, the objective of this last example was to illustrate that despite further research needs to be accomplished, SCS can effectively compute 3D frames.

7 Conclusions

Once the analysis of the five numerical examples has concluded, it is interesting to summarise the advantages and drawbacks that have been observed so far, in order

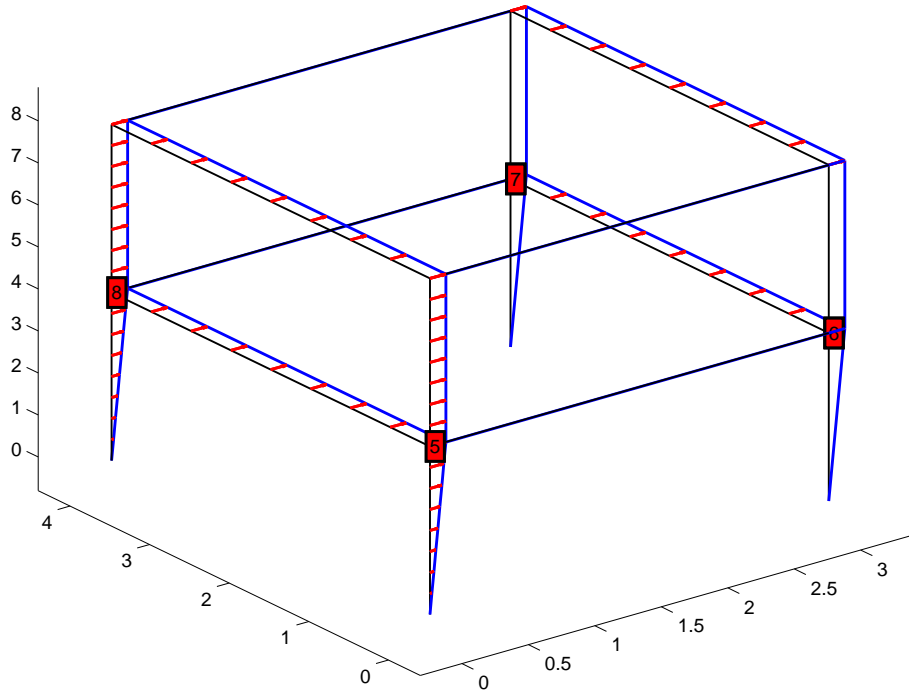


Figure 35: Plastic hinge detection by SCS for the 3D two-storey frame with three subdivisions

to effectively improve the software and orientate the future research. These five examples constitute a good test to SCS, as it explores both two and three dimensions, point and uniform distributed load, trussed and framed structured. Hence, it is sensible to draw conclusions of SCS functioning.

Firstly, when working with a certain software is important to know the limits of the model employed. The example of the simply supported beam, perhaps the easiest example to compute, has showed that the model experiences a weird behavior on lower bounds when the exact plastic hinge location has been detected in the first iteration, without refining the bars. However, by continuously dividing the bars, convergence to the real collapse load has been observed, so once the user is aware of this limitation no further problems should be derived from it.

By comparing results of SCS with the results obtained using linear analysis methods, such as SAP2000[®], we have found that for the simple case of the embedded frame with horizontal point load SCS enables to load the structure 17% more. This

fact illustrates the convenience of employing SCS for collapse computations, since it finds the load factor necessary for the structure to collapse and the real collapse mechanism. Linear elastic methods, however, only find the first plastic hinge to form. Moreover, it does not only give a greater collapse load, but also bounds it upper and lower as precisely as the user desires. This case and the steel tower justify the use of SCS in structure limit analysis.

Finally, once clarified the limits of the model and the justification of employment, a 2D and a 3D frame have been computed in order to study the influence of several parameters in the convergence of the load factor. The theoretical basis states that bounding of the actual load factor comes from approximating the yield curve/surface of each beam element inwards or outwards and from considering a secant and a tangent moment distribution to the actual one in UDL elements. In the last two examples the influence of each factor has been widely studied. According to the results obtained, it is clear that refining the UDL elements leads to much better results than the ones obtained by refining the yield surface. By subdividing the UDL elements, the actual moment distribution is progressively approximated, with the gap diminishing with the square of the beam length. This fact explains the results obtained. Nonetheless, it is advisable to combine both bounding techniques in order to obtain the best results possible. This influence is best observed in Figure 32. By improving the yield criteria results slightly get better, although refining the UDL elements -i.e. having a more representative model- is decisive in achieving convergence.

To conclude, SCS is a powerful tool that computes all kind of trussed and framed structures using plastic analysis, offering a bounding as precise as desired of the load that induces collapse in the structure. It enables the user to perform stronger computations than the ones with elastic methods, as well as being able to combine two different methods of bounding the load factor for more efficient computations.

8 Future research

SCS successfully computes upper and lower bounds for framed or trussed two-dimensional structures, formed by beams with I/H-shaped cross sections or C4 sections. The problem approach and the procedure employed to solve the problem has been detailed throughout the project. However, SCS is currently being improved in order to meet the desired requirements by the authors.

8.1 3D Frames yield criteria

Implementing the same idea for three-dimensional structures is the major enhancement that is currently being done. Besides the obvious, two major differences arise between the two and the three dimensional case concerning the implementation of SCS.

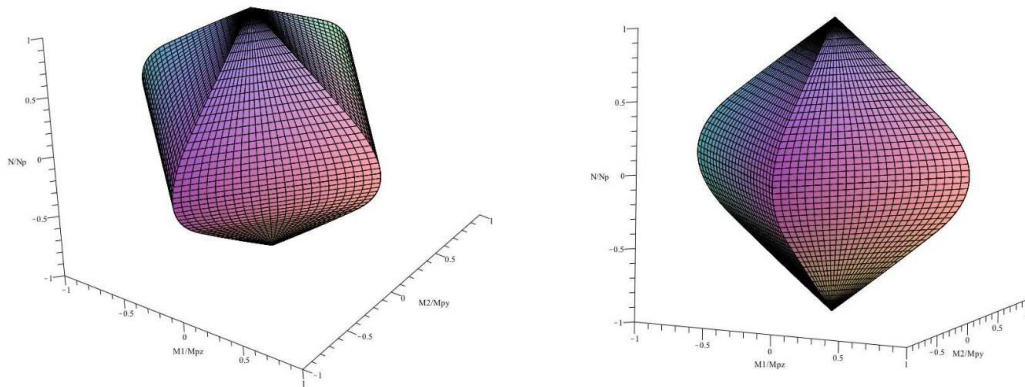


Figure 36: Yield surface for IPE360 cross section

Finally, the major drawback encountered while defining the LP problem for three-dimensional structures is defining a sequence of inscribed and circumscribed polyhedra, since in 3D there are two bending moments affecting the beam element. Hence, yield criteria will now be described by a convex surface, instead of a convex surface in 2D. See Figure 36 for an example of a yield surface for a IPE360 beam.

Typically, yield surfaces consist of the 2D yield surfaces in the $\left(\frac{M_z}{M_{pz}}, \frac{N}{N_p}\right)$ and $\left(\frac{M_y}{M_{py}}, \frac{N}{N_p}\right)$ joined using lines parallel to the plane $\frac{N}{N_p} = 0$, obtaining a ruled surface. The strategy used to create a sequence of polyhedra that approximates the surface inwards and outwards will be similar to the 2D case. Accounting that only I/H-shaped cross sections are introduced so far, the double symmetry enables to work in one octant of the space and then apply symmetries to the planes obtained.

The technique for creating the polyhedra approximating the yield surface is currently being developed.

8.2 Cross Section Library

Besides implementing the 3D case, there is a concern for widening the range of cross sections available to use in the computations. I/H-shaped cross sections and C4 sections are perhaps the most commonly used, but regardless this condition a powerful program must offer the maximum number of possibilities to the user. In Section (4.2.2) sections with T, L and U are mentioned. The first one is currently being studied, and the expression of the yield surface has been analytically found. The problem encountered is that it differs from the I/H-shaped sections that it cannot be written explicitly, i.e. the moment as a function of the axial stress or viceversa. The fact that the yield surface is an implicit function of $\frac{M}{M_p}$, $\frac{N}{N_p}$ impedes to use the Newton-Raphson routine employed for the explicit case. One of the main goals is to adapt the technique in order to make it able to operate with implicit functions. The case of L and U is quite different, since the analytical formulae have not been found yet, although it constitutes another point of major interest. Once the yield surface is written, they will likely be easily studied with the Newton-Raphson technique for implicit functions.

8.3 LP solver

The routine `linsolve` in Matlab[®] is the one currently used to resolve both the upper and the lower bound problems. For the cases studied it has provided satisfying results within reasonable time. In fact, the case that took longer to compute is the one described in Section 6.4.4 with 20 lines approximating the yield surface per node, and the computation time was around 20 s. Nonetheless, the final goal of SCS is to compute complex 3D framed structures with hundreds or thousands of elements. At this extent, `linsolve` would probably not only be slow, but also insufficient to assure convergence. The authors' idea is to introduce the convex optimization package SDPT3 to achieve a better performance of the LP solver. The main difficulty nowadays is that the LP problems (14) and (15) are described in the general LP form, see (4), and SDPT3 package requires the problem to be expressed with nonnegative variables and equality constraints. Even though it does not suppose a great effort to transform the existing problems into ones that fit this characteristics by using slack variables, it is a procedure that has not been developed yet, and it will gain interest as the 3D case reaches to completion.

8.4 Adaptivity

Throughout the project it has been proven repeatedly the importance of refining the elements bearing UDL in a framed structure, in order to reduce the bound gap in the computations. Nonetheless, the procedure of refining is nowadays hand-driven,

and hence it may result tedious when precise computations have to be carried out. The authors' idea is to transform SCS into an iterative method where the user would set manually the precision desired in the computation, and the program would use adaptivity to improve the computations after each iteration. The basic idea of adaptivity applied to SCS consists of locating after each iteration the elements that contribute the most to the gap, i.e. the top rows of the matrix `gap_eval`. The criteria employed for selecting how many elements should be refined would also be a user's choice. Thereby, after each iteration SCS would refine the selected bars, a measure that can be complemented with improving the approximation of the yield curve/surface. Using adaptivity, SCS would not only be more efficient, but also easier and faster to be used, since the work of entering the structure information would only be done once.

References

- [1] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, Cambridge, 1994.
- [2] H. Maier and Leibnitz, *Versuche mit eingespannten und einfachen Balken von I-Form aus St. 37*, vol. 7. Bautechnik, 1929.
- [3] H. Greenberg and W. Prager, “On limit design of beams and frames,” *Proc. ASCE*, vol. 77, no. 59, 1951.
- [4] B. Neal and P. Symonds, “The calculations of collapse loads for framed structures,” *J Inst Civil Eng*, 1951.
- [5] B. Neal and P. Symonds, “The rapid calculation of plastic collapse loads for a framed structure,” *Proceedings of the institution of civil engineers London*, vol. 1, no. 3, pp. 58–100, 1952.
- [6] B. Neal and P. Symonds, “Recent progress in the plastic methods of structural analysis,” *Journal of the Franklin Institute*, vol. 252, pp. 383–407, November 1951.
- [7] E. Chwalla, “Three contributions to the problem of the load carrying capacity of statically indeterminate steel frames,” *Publ. of Intern. Assn. Bridge and Structural Engrg. Zurich, Switzerland*, vol. 2, pp. 96–125, 1933-34.
- [8] M. Baker, J.F. Horne and J. Roderick, “The behaviour of continuous stanchions,” *Royal Society*, vol. 198, p. 493, 1949.
- [9] A. Charnes and H. Greenberg *Bull. Amer. Math. Soc.*, vol. 57, 1951.
- [10] C. Charnes, A. Lemke and O. Zienkiewicz, “Virtual work, linear programming and plastic limit analysis,” *Royal Society*, vol. 251, pp. 110–116, May 1959.
- [11] J. Heyman, “On the minimum weight design of a simple portal frame,” *Int J Mech Sci*, 1960.
- [12] M. Horne, “Determination of the shape of fixed ended beams for maximum economy according to the plastic theory,” fourth congress, final report, International association of bridge and structural engineering, 1953.
- [13] J. Baker and J. Heyman, *Plastic design of frames, fundamentals*, vol. 1. Cambridge University Press, 1969.
- [14] V. Watwood, “Mechanism generation for limit analysis of frames,” *Journal of the Structural Division*, vol. 105, pp. 1–15, January 1979.

-
- [15] E. Anderheggen and H. Knopfel, "Finite element limit analysis using linear programming," *International Journal of Solids and Structures*, vol. 8, pp. 1413–1431, January 1972.
- [16] J. Munro, "Optimal plastic design of frames," 71-136, Proceedings of NATO, Waterloo: Advanced Study in Engineering Plasticity by Mathematical Programming, 1977.
- [17] R. Livesley, *Linear programming in structural analysis and design*. Wiley, 1977.
- [18] S. Maier, G. Giacomini and F. Paterlini, "Combined elastoplastic and limit analysis via restricted basis linear programming," *Computer methods in applied mathematics and engineering*, vol. 19, pp. 21–48, 1979.
- [19] A. Jennings and T. Tam, "Automatic plastic design of frames," *Engineering Structures*, vol. 8, 1986.
- [20] G. Thierauf, "A method for optimal limit design of structures with alternate loads," *Computer methods in applied mathematics and engineering*, vol. 16, pp. 49–134, 1987.
- [21] L. Damkilde and O. Hoyer, "An efficient implementation of limit state calculations based on lower-bound solutions," *Computers & Structures*, vol. 49, no. 6, pp. 953–962, 1993.
- [22] F. Tin-Loi, "Plastic limit analysis of plane frames and grids using gams," *Computers & Structures*, vol. 54, no. 1, pp. 15–25, 1995.
- [23] A. Kaveh and M. Jahanshahi, "Plastic limit analysis of frames using ant colony systems," *Computers & Structures*, vol. 86, pp. 1152–1163, 2008.
- [24] L. Corradi, L. Luzzi and P. Vena, "Finite element limit analysis of anisotropic structures," *Computer methods in applied mathematics and engineering*, vol. 195, pp. 5422–5436, 2006.
- [25] T. Tjhin and A. Kuchma, "Integrated analysis and design tool for the strut-and-tie method," *Engineering Structures*, vol. 29, pp. 3042–3052, March 2007.
- [26] H. Van Long and N. Dang Hung, "Limit and shakedown analysis of 3d steel frames," *Engineering Structures*, vol. 30, pp. 1895–1904, January 2008.
- [27] J. Ciria, H. Peraire and J. Bonet, "Mesh adaptive computation of upper and lower bounds in limit analysis," *International Journal for Numerical Methods in Engineering*, vol. 75, pp. 899–904, January 2008.
- [28] N. Borges, L.A. Zouain and A. Huespe, "A nonlinear optimization procedure for limit analysis," *European Journal of Mechanics*, vol. 15, no. 3, p. 487512, 1996.

- [29] A. Lyamin and S. Sloan, “Lower bound limit analysis using non-linear programming,” *International Journal for Numerical Methods in Engineering*, vol. 55, pp. 573–611, 2002.
- [30] K. Krabbenhoft and L. Damkilde, “A general nonlinear optimization algorithm for lower bound limit analysis,” *International Journal for Numerical Methods in Engineering*, vol. 56, pp. 84–165, 2003.