RESEARCH REPORT

VTT-R-00524-10

Optimization tools for steel portal frames - Software documentation

Authors: Petr Hradil, Matti Mielonen, Ludovic Fülöp

Confidentiality: Confidential

Public after closing the project (July 2010).

Report's title	
Optimization tools for steel portal frames - Software documentation	
Customer, contact person, address Commission of the European Communities	Order reference RFS-PR-06054
Project name Prefabricated steel structures for low-rise buildings in seismic areas	Project number/Short name 12596/PRECASTEEL
Author(s) Petr Hradil, Matti Mielonen, Ludovic Fülöp	Pages 33 p.
Keywords steel, portal frames, optimization, Python script	Report identification code VTT-R-00524-10

Summary

The document presents description of two software tools for optimization of portal frames (AP Frame and EV Frame). Both tools use the methods from Eurocode 3 (EN 1993-1-1 [7]) for the design part, and genetic algorithm methods for optimization. The optimization tool developed for Abaqus (AP-Frame) models the frame with "Shell" type finite elements for out-of-plane stability calculations, while in the Excel tool (EV-Frame) stability is calculated at the level of members using analytical expressions of critical loads.

Two types of portal frames are considered in the analysis. Hot-rolled frames are composed of rolled HE and IPE sections with welded haunch on the beam. The second group covers tapered frames welded from steel plates.

Two analytical methods can be used in Abaqus tool (AP-Frame): General method with overall out-of-plane reduction factor and global non-linear analysis using initial imperfections. The third method - member checks using interaction formulae (IFM) - is included in Excel tool (EV-Frame). Both programs use genetic algorithms as optimization engine, with several possibilities of selection, crossover, and mutation methods.

This document is intended to provide background information on the use of the software. Input/Output data formats are given in detail, together with ways of interpretation of the data produced.

Public after closing the project (July 2010).

Confidentiality	Confidential

Espoo 19.1.2010

Written by	Reviewed by	Accepted by
Petr Hradil Research Scientist	Ludovic Fülöp Senior Research Scientist	Eila Lehmus Technology Manager

VTT's contact address
P.O. Box 1000, FI-02044 VTT, Finland

Distribution (customer and VTT)

Customer (partners of EU-RFCS project PRECASTEEL)	1 pdf. copy each
VTT/Register Office:	1 copy

Preface

The research report describes the software tools, for analysis and optimization of steel portal frames, developed in VTT Research Centre of Finland for the PRECASTEEL (RFS-PR-06054) project.

The tools are capable of automatic model creation, results evaluation and genetic algorithm optimization, and they are using an independent calculation core for finite element analysis.

EV-Frame is programmed as Microsoft Excel script, with open source finite element calculation [9] under GPL - general public license. AP-Frame is programmed in Python, using Abaqus commercial or academic licenses for numerical calculations. Detailed description of all required inputs and produced outputs is included in this report. The most important methods and objects of the code are also explained.

Two types of portal frames are considered in the analysis. Hot-rolled frames are composed of rolled HE and IPE sections with welded haunch on the beam. Those frames are usually fixed at the base. The second group covers tapered frames, fabricated from steel plates, which are usually pinned at the base.

Two design methods can be used in the Abaqus tool (AP-Frame): General method of analysis (GMA), with overall out-of-plane reduction factor, and global non-linear analysis using initial imperfections (GMNIA). The third method - member checks using member slenderness and interaction formulae - is included in Excel tool (EV-Frame). Both programs use genetic algorithms with several possible selection, crossover, and mutation methods.

This document provides background information on the use of the software.

Espoo 19.1.2010

Authors

Abbreviations

CAE	computer aided engineering
EC	Eurocode
FB	flexural buckling
FE	finite element
FEM	finite element method
GA	genetic algorithm
GMA	general method of analysis
GMNIA	analysis of the geometrically and materially nonlinear structure, including effects of initial imperfections
GNLA	global non-linear analysis
GPL	general public licence
GUI	graphic user interface
HR	hot-rolled
IFM	analysis method based on interaction formulae
LBA	linear eigenvalue (bifurcation) analysis
LX	Laplace crossover
MNA	materially nonlinear analysis
LTB	lateral torsional buckling
RRWS	ranked roulette wheel selection
SBX	simulated binary crossover
SDOF	single degree of freedom
SLS	serviceability limit state
SVG	Scalable Vector Graphics standard
TB	torsional buckling
TFB	torsional-flexural buckling
ULS	ultimate limit state
VBA	Visual Basic for Applications
WT	welded-tapered
XML	Extensible Markup Language

Symbols

α_{cr}	critical multiplier of the loads (in-plane buckling)
$\alpha_{cr,op}$	critical multiplier of the loads (out-of-plane buckling)
$\alpha_{ult,k}$	minimum load amplifier of the design loads to reach the characteristic resistance
α_m , α_n	coefficients of variable cross-sections (Šapalas et al [1])
γ_G , γ_Q	partial safety factors for loading
γ_{M0} , γ_{M1}	partial safety factors for material
$\bar{\lambda}_y$, $\bar{\lambda}_z$	non-dimensional slenderness (“y” axis and “z” axis)
$\bar{\lambda}_{op}$	overall out-of-plane non-dimensional slenderness
χ_y , χ_z	buckling reduction factor (“y” axis and “z” axis)
χ_{op}	overall out-of-plane buckling reduction factor
ψ_0 , ψ_2	combination factors

$e_{z,\max}$	maximum eccentricity of lateral support from the shear centre of the CS
$e_{z,\min}$	minimum eccentricity of lateral support from the shear centre of the CS
f_y	yield stress
i_0	polar radius of gyration
k_z	effective length factor considering weak axis end-support conditions (k_z = 1 for pinned-pinned, k_z = 0.5 for fixed-fixed, k_z = 0.7 for pinned-fixed)
k_w	effective length factor considering end warping support (k_w = 1 for no warping support, k_w = 0.5 for both end warping support)
z_g	load position (z-distance between load and shear centre)
A	gross cross-section area
A_f	second order amplification factor
A_V	shear area
C_1, C_2	moment gradient factors
C_{my}	equivalent moment factors for FB
C_{mLT}	equivalent moment factors for LTB
E	Young's modulus of elasticity
G	shear modulus
I_t	torsional constant
I_w	warping constant
$I_{y,\max}$, $I_{z,\max}$	strong/"y" and weak/"z" axis second moment of area at the large member cross-section
$I_{y,\min}$, $I_{z,\min}$	strong/"y" and weak/"z" axis second moment of area at the small member cross-section
L	length of member
L_h	(horizontal) length of the haunch
$L_{cr,y}$, $L_{cr,z}$	critical lengths for strong/"y" and weak/"z" axis FB
N_{Ed}	design axial forces
N_{Rk}	characteristic resistance in compression
M_{cr}	critical bending moment
M_{eq}	earthquake mass
$M_{y,Ed}$	design bending moment to major/"y" axis bending moment
$M_{y,Ed,1}$ to $M_{y,Ed,5}$ - $M_{y,Ed}$	in quarter points along the length of the member
$M_{y,Rk}$	bending resistance (elastic or plastic depending on design method) to "y"/major axis
$S_d(T)$	the ordinate of the design spectrum for seismic calculation
$V_{z,Ed}$	shear force in "z" direction
$V_{z,Rk}$	shear resistance in "z" direction

Contents

1 Introduction

Optimization of structures is a difficult engineering task. In the everyday design practice, optimization is carried out integrated into the design process. It is self understood that the design engineer tries to provide an optimized solution for the design task at hand. Rarely the task of optimizing is tackled systematically in everyday design. As a result, the quality, in the sense of optimal or not, of the resulting engineering structures depend very much on the expertise of the designer.

More systematic optimization is usually not carried out, because of lack of expertise or time pressure during the design process. One of the solutions for overcoming both this factors is to give the designer easy to use and efficient software tools for optimization of structures.

In this work two such tools have been developed.

2 Goals

The main goal of this document is to provide documentation for the users (primarily in the PRECSATEEL project) of the two optimization tools, AP-Frame and EV-frame. The document presents: the input/output cycle of the software, the techniques of using the software, and some hints on the structure of the scraps.

The aim developing AP-Frame and EV-Frame was to systematically optimise the portal frame configurations pre-designed by Varelis *et al* [2].

3 AP-Frame: Python analytical script in Abaqus

Python analytical script is a software module designed to perform the design calculations for a single frame configuration, or of a pre-defined set of frames, using global non-linear (GNLA or Method 1) or the general method of analysis (GMA or Method 2). The procedures used in AP-Frame, are refined and greatly expended versions of the ones presented by Fülöp and Beaucaire [3].

The module can be used to evaluate the load bearing capacity of frames with known geometry, or integrated into an optimization procedure. The design procedures are refined versions of the procedures reported.

The *AP-Frame* module is coded in Python, in order to easily implement Abaqus/CAE commands for creating the numerical model, and evaluating the results. The method for exporting frame drawings uses XML specific language. Parts of the code also contain UNIX or Windows specific system commands.

The module was tested with Abaqus version 6.9, using Abaqus/CAE license for modelling and result interpretation; and up to four Abaqus/Standard licences for calculations.

3.1 Executing the *AP-Frame* module

The module is contained in the file *"mframe*xxx*.pyc"* in a compiled form (Note: xxx stands for the number of version), and is called by the Python *"run_single.py"* script. The alternative usage from within the optimization procedures will be explained in the following chapters (Chapter 3.7). There are following options how to execute the *"run_single.py"* script:

- from operating systems command line - write "**abaqus cae noGUI=run_single.py".**
- from the Abaqus/CAE GUI -select "**File-Run Script…**" (**Alt+F**, **Alt+R**) and choose *"run_single.py"*.

The script *"mframe*xxx*.py*" (or its compiled version *"mframe*xxx*.pyc"*) has to be in the same folder as *"run_single.py"*, for correct execution. The same folder has to contain also the file *"input.txt"*.

By default, the calculation starts with 4 parallel threads in UNIX based systems and with a single thread in Windows. This can be modified by editing *"run_single.py"* in a text editor. The optimal number of parallel threads depends on the number of processors and Abaqus licenses available. Generally, it is efficient to run as many threads as the number of processor cores. Calculation progress can be monitored in Abaqus/CAE GUI interaction line or by reading Abaqus replay file *"abaqus.rpy"*.

On execution, the geometric, material, support etc. parameters are read from the *"input.txt"* file. In a single thread mode the first line will be read, the model built in Abaqus and analysis started. The next line will be read only after the analysis of the first configuration is finished and results are reported in *"output.txt"*. In multi-thread mode, several lines are read and Abaqus models are simultaneously submitted for analysis.

The basic results are reported in the *"output.txt"* in a line by line format. The same basic results are written also to the *"database.txt"* file. This file is a repository of all previous results, each result within it having a unique identifier. If a configuration from *"input.txt"*, already exists in the database file that configuration will not be analysed, but the result will be extracted from the database file. In order to force the AP-Frame module to re-run an analysis, one has to remove the corresponding line from the *"database.txt"* file first. For a clean run of an input file, the database should be empty.

The detailed individual results are reported in files *"report-*xxxxxxxxxx*.txt"*, where xxxxxxxxxx is the unique identifier of that result from the database file.

Detailed description of the structures, and content, of the Input/Outpus files used by the AP-Frame follows in the next section.

3.2 Single frame calculation using the *AP-Frame* module

If the *"input.txt"* file has only one input line, and the database.txt file is empty, the steps executed during the calculation of the single frame are explained in the diagram from Figure 1.

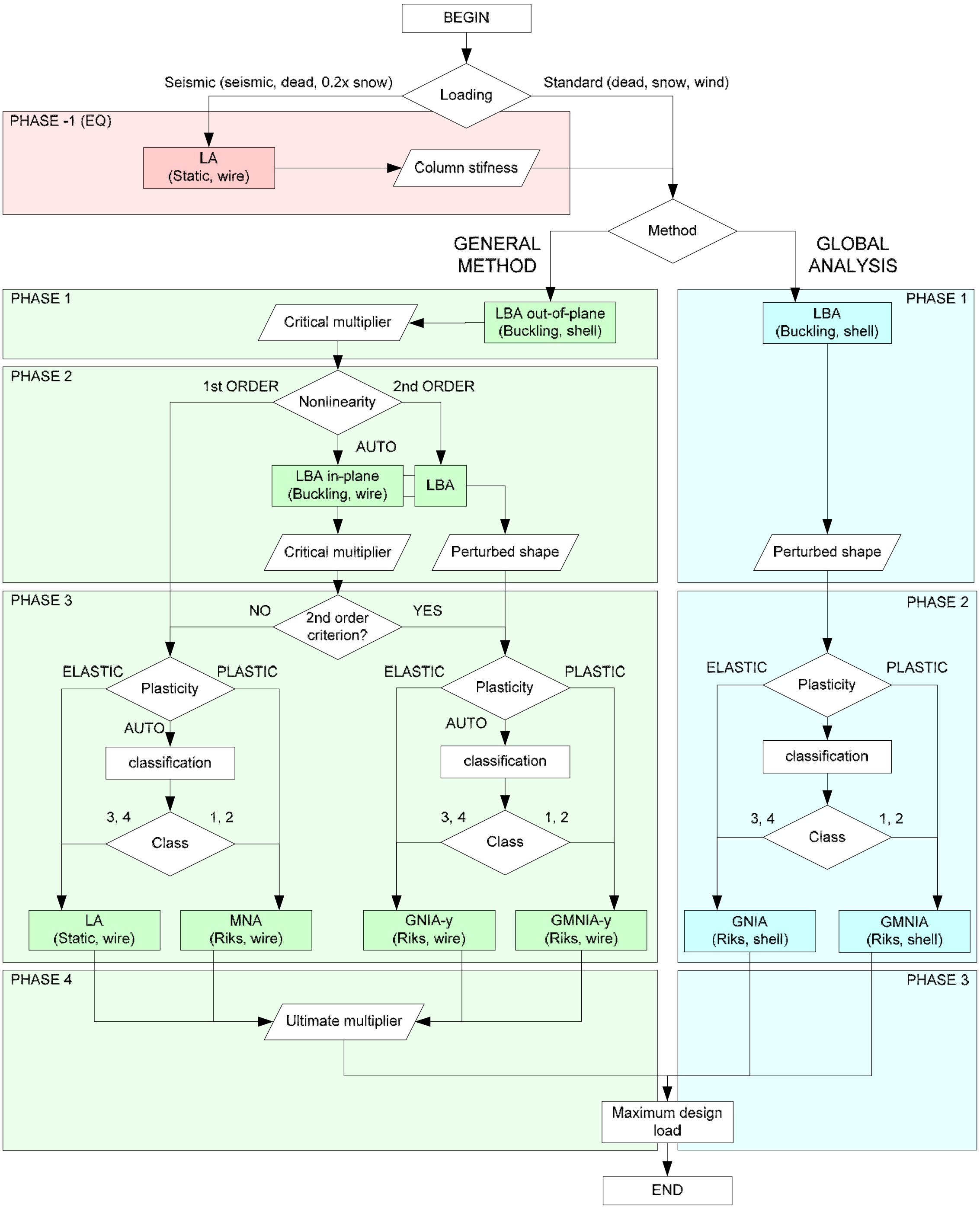


Figure 1. Abaqus script AP-Frame.

3.2.1 Inputs, the *"input.txt"* file

The input file *"input.txt"* is required for the calculation. It can be created: (1) using a text editor, (2) from *"input.xls"* with enabled macros and by pressing

"**Export file**" button, or (3) automatically by the optimization script (explained in Chapter 4.4), and it contains the analysis controls described in Table 1.

Table 1. Detailed content of the AP-Frame input file "input.txt".

Input file *"input.txt"* specification			
Input name		Values accepted	Description
Inp. num.	Input number	any value	The unique name of the frame that can be recognized in "output.txt" file
H (mm)	Height	positive number	Eaves height in mm (e.g. 6000)
S (mm)	Span	positive number	Axis-to-axis span (e.g. 20000). For double-span frames start with "2x" (e.g. 2x20000)
T (mm)	Frame distance	positive number	Frame distance in mm (e.g. 6000)
Ang (%)	Pitch	0 to 100	Roof pitch in percent (e.g. 15)
Lh/S	Haunch ratio	higher than 2	Frame span / Length of the haunch
Column hc	Column height or HR profile name*	positive number or text*	Height of the column profile at the base in mm (* profile name when hot-rolled profile used e.g. "HE300A")
hhc	Height of the haunch	positive no. or blank	Height of the column profile at the haunch in mm (* blank if hot-rolled profile specified)
bc	Width of the profile	positive no. or blank	Width of the column profile at the haunch in mm (* blank if hot-rolled profile specified)
tfc	Flange thickness	positive no. or blank	Thickness of the column flange in mm (* blank if hot-rolled profile specified)
twc	Web thickness	positive no. or blank	Thickness of the column web in mm (* blank if hot-rolled profile specified)
Beam hb	Beam height*	positive number	Height of the rafter profile at the base in mm (* profile name when hot-rolled profile used e.g. "IPE400")
hhb	Height of the haunch	positive no. or blank	Height of the rafter profile at the haunch in mm (* blank if hot-rolled profile specified)
bb	Width of the profile	positive no. or blank	Width of the rafter profile at the haunch in mm (* blank if hot-rolled profile specified)
tfb	Flange thickness	positive no. or blank	Thickness of the rafter flange in mm (* blank if hot-rolled profile specified)
twb	Web thickness	positive no. or blank	Thickness of the rafter web in mm (* blank if hot-rolled profile specified)

Input name		Values accepted	Description
Steel grade	Material model	"S" + 3 digits + letter	Steel grade ("S" + 3 digits) and hardening ("T" for strain hardening, "F" for elastic-plastic behaviour e.g. "S275T")
Fixed base	Support conditions	0,1,T,F	In-plane fixing of the frame base ("0" or "T" for fixed, "0" or "F" for pinned)
purl. beam	Number of purlins on beam	whole number higher than 1	Number of purlins on beam including corner purlins
purl. col.	Number of side-rails on column	whole number higher than 1	Number of side-rails on column including corner rails
L.B. (mm)	Purlin eccentricity	positive number	Distance between purlin center of gravity and purlin contact plane
L.C. (mm)	Side-rail eccentricity	positive number	Distance between side-rail center of gravity and its contact plane
Pdead (N/m2)	Dead load	positive number	Characteristic dead load in N/m2 including the estimated self-weight of the frame and purlins (e.g. 380)
Psnow (N/m2)	Snow load	positive number	Characteristic snow load in N/m2 (or quasi-pernament snow load in case of seismic design situation e.g. 750) Can be empty
Wind load	Wind load	number+"t"+"1","2"," 3"or"4"	Basic wind velocity in m/s and terrain category (e.g. 30t1). Can be empty
Crane load	Crane load	positive number	Hoist load in tons (e.g. 10). Can be empty
Seismic load	Seismic load	"1"or"2"+"A","B","C", "D"or"E"+number	Spectrum type ("1"or"2"), ground type ("A"or"B") and ground acceleration (e.g. "1B0.32") Can be empty
An. type	Analysis type	1,2,GA,GM	Global non-linear analysis ("1"or"GA") or General method ("2"or"GM")
Kill Riks	Calculation termination	0,1,T,F	"0"or"T" sets the calculation to be terminated when the load starts descending
Mesh size	Mesh size	"auto" or number	Size of the shell elements in mm ("auto" sets the automatic recommended size - half of the flange width)
Imp	Bow imperfections	"auto" or number	Out-of-plane imperfections for Global analysis or in-plane imperfections for General method in mm ("auto" sets the imperfections according to EC3)
Plast an.	Plastic analysis	"auto",1,2,F,T,EL,PL	"1","F"or"EL" sets the calculation to output elastic results, "2","T"or"PL" sets plastic analysis, "auto" determines according to profile class
Vert. limit	Vertical SLS limit	"auto", number or "S/"+number	SLS limit in mm or the fraction of span (e.g. "S/200")
Hor. limit	Horizontal SLS limit	"auto", number or "H/"+number	SLS limit in mm or the fraction of height (e.g. "H/150")

3.2.2 Basic outputs, the *"output.txt"* and *"database.txt"* files

Basic calculation results are stored in two files:

- the output file *"output.txt"* contains basic results from the current run
- database file *"database.txt"* where basic results from the current run are appended to results obtained from previous runs (Note: the database file is a repository of all results obtained in all runs.)

The content of these files is presented in detail in Table 2. The two files have identical structure, but while the *"output.txt"* file is rewritten in each run, the *"database.txt"* file is only appended.

Table 2. Detailed content of the AP-Frame output files "output.txt" and "database.txt"

Output file *"output.txt"* and database file *"database.txt"* specification		
Output name	Values	Description (* values copied from the input file *"input.txt"*)
Dat num.	number	Specific number in the database (refers to the database line). The records are sorted only in "database.txt".
Inp. num.		*
H (mm)		*
S (mm)		*
T (mm)		*
Ang (%)		*
Lh/S		*
Column	text	Hot-rolled profile name (e.g. "HE300A") or "height"x"haunch height"x"flange width"x"flange thickness"x"web thickness" (e.g."400x800x300x12x6")
Beam	text	Hot-rolled profile name (e.g. "IPE400") or "height"x"haunch height"x"flange width"x"flange thickness"x"web thickness" (e.g."400x800x300x12x6")
Steel	"S" + number	Steel class (e.g. S275)
H	T,F	Strain hardening ("T" true, "F" false)
Fixed base		*
purl. beam		*
purl. col.		*
L.B. (mm)		*
L.C. (mm)		*
Pdead (N/m2)		*
Psnow (N/m2)		*
Wind load		*
Crane load		*
Seismic load		*
An. type		*
Kill Riks		*
Mesh set		*
Mesh used	number	Mesh size in mm
Imp. set		*
Imp. used	number	Imperfections in mm

Output file “*output.txt*” and database file “*database.txt*” specification		
Output name	Values	Description (* values copied from the input file *“input.txt”*)
Vert. limit		*
Hor. limit		*
Weight (t)		Frame weight in tons
Area (m2)		Surface area in m2
Pdes *N/m2*	Design load	The vertical load in N/m2 in persistent/transient design situations or the horizontal load in kN in seismic design situation
Pchar *N/m2*	Characteristic load	The vertical load in N/m2 in persistent/transient design situations or the horizontal load in kN in seismic design situation
Pult *N/m2*	Ultimate load	The vertical load in N/m2 in persistent/transient design situations or the horizontal load in kN in seismic design situation
Pser *N/m2*	Serviceability load	The vertical load in N/m2 in persistent/transient design situations or the horizontal load in kN in seismic design situation
ULS util.	Utilization in ultimate limit state	*Pdes/Pult.* When frame fails in the pre-loading phase (*Pult=0.0*), the value is set to 1000. (in mixed design situation the value always corresponds to the seismic combination while *Pdes, Pchar, Pult, Pser* are outputs from the standard loading combination).
SLS util.	Utilization in serviceability limit state	*Pchar/Pser.* When frame fails in the pre-loading phase (*Pser=0.0*), the value is set to 1000. (in mixed design situation the value always corresponds to the seismic combination while *Pdes, Pchar, Pult, Pser* are outputs from the standard loading combination).
Plast set		*
Plast used	"EL" or "PL"	Plastic or elastic calculation. This value is needed for comparing records in database.

3.2.3 Calculation statistics, the *“report-mfr.txt”* file

Basic statistics concerning the number of analysed frames, run-time, number of threads used etc. is reported in the *“report-mfr.txt”* file. The information saved in the report file *“report-mfr.txt”* is important in the optimization of frames where several runs are submitted consequently. This file is rewritten when launching a new optimization.

for example:
```
*** OVERALL REPORT ***
Calculation started : 26.10.2009 11:47:38
Calculation stopped : 26.10.2009 11:57:22
Calculation time    : 584 sec
Frames processed    : 19
Frames skipped      : 11
CPUs used           : 4
```

3.2.4 Detailed outputs

As mentioned, the report file *“report*-xxxxxxxxxx*.txt”* is where detailed calculation results are stored for each frame. In the name of the file, *xxxxxxxxxx* stands for the unique identification number from the database line for that frame. The files contain information about the calculation including load-displacement relationship data. At the beginning of the file, the line from the output and database files is copied. Further details are as follows:

A. Load information

This section contains all loads applied on the model including vertical loads (snow and dead load), wind load, crane load and seismic load.

for example:
```
*** VERTICAL LOAD ***
dead load                  : 300.0 N/m2
snow load                  : 750.0 N/m2
design combination         : 1530.0 N/m2
characteristic combination : 1050.0 N/m2
```

B. Calculation limits

Serviceability limits can be specified in the input file. However, when the "auto" value is set, the limits are calculated according to the Eurocode and stored in this section.

for example:
```
*** SERVICEABILITY LIMIT STATE LIMITS ***
vertical deflection   : 144.8525 mm
horizontal deflection : 60.0 mm
```

C. Cross-sectional parameters

Section classification is checked at the largest end of the variable cross-section. Section classes are then used for the buckling curve selection and calculation of imperfections.

for example:
```
*** CROSS-SECTIONAL PARAMETERS ***
- cross-section class is 1
- out-of-plane FB buckling curve is b
- in-plane FB buckling curve is a
- LTB buckling curve is a
- plastic analysis selected
```

D. Calculation method-specific information

This section is adapted to the calculation method used in the design.

for example:
```
*** GENERAL METHOD ***
RUNNING OUT-OF-PLANE LINEAR EIGEN VALUE BUCKLING
ANALYSIS (LBA) WITH SHELL MODEL
- critical multiplier is 0.0543337
RUNNING IN-PLANE LINEAR EIGEN VALUE BUCKLING ANALYSIS
(LBA)
- critical multiplier is 0.89817
- in-plane vertical imperfection is 57.941 mm
- in-plane horizontal imperfection is 0.474341649025 mm
- using imperfection 57.941 mm
RUNNING GEOMETRICALLY AND MATERIALLY NONLINEAR
ANALYSIS WITH IN-PLANE IMPERFECTIONS (GMNIA)
Applying horizontal wind force 47.5671209833 N.
Applying wind load (upwind side) 0.0375565219348 N/m2.
Applying wind load (downwind side) -1.55187659883
N/m2.
- Applying sway imperfection 0.67082039325 mm
- Adding sway imperfection into input file.
```

E. Frame physical properties

for example:
```
- mass properties approximated
- mass of the frame is          : 0.849379868569 tons
- surface area of the frame is: 29.8621285903 m2
- length of short welds is      : 2626.56610071 m
- length of long welds is       : 6920.04728336 m
- total length of welds is      : 9546.61338407 m
```

E. Load-displacement data

for example:
```
*** PUSHOVER CALCULATION RESULTS ***
frame,   load (N), amplifier,   stress (MPa),eq.plast.strain,vert.deflection
    0,      207.4,    0.00078,        4.93439,        0.00000,        4.71662
    1,     6231.1,    0.02343,      104.17348,        0.00000,      118.22704
    2,    12263.3,    0.04611,      208.89291,        0.00000,      240.59586
    3,    15485.6,    0.05823,      266.43594,        0.00000,      309.51572
    4,    18183.9,    0.06837,      281.30084,        0.00604,      409.44717
    5,    19132.9,    0.07194,      288.80084,        0.01324,      506.02548
    6,    19946.8,    0.07500,      294.11713,        0.01834,      602.40869
    7,    21053.2,    0.07916,      300.50980,        0.02447,      746.88879
```

F. Calculation results

In the last section the specific load levels are stored when limit states were reached (or the limit states predictions were calculated).

for example:
```
*** LIMIT POINTS ***
 id., load (N), amplifier, description
MAX:,  21053.2,   0.07916, - maximum load
UEL:,  15965.2,   0.06003, - elastic limit (275.0 MPa)
UPL:,  19113.9,   0.07187, - plastic limit (0.0130952380952)
SLV:,   7543.6,   0.02836, - vertical serviceability limit (144.8525 mm)
SLH:,  19900.1,   0.07483, - horizontal serviceability limit (60.0 mm)
SEV:,   7413.2,   0.02787, - vertical serviceability prediction (elastic)
SEH:,  23412.9,   0.08804, - horizontal serviceability prediction (elastic)
SPV:,   7340.1,   0.02760, - vertical serviceability prediction (plastic)
SPH:,  23574.0,   0.08864, - horizontal serviceability prediction (plastic)

*** FEASIBILITY PARAMETERS ***
 id.,  load (N), amplifier, description
DES:,  265949.2,   1.00000, - design load
ULS:,   15965.2,   0.06003, - ULS limit (elastic)
CHR:,  182514.1,   0.68627, - characteristic load
SLS:,    7413.2,   0.02787, - SLS limit (vertical)

*** ELASTIC RESULTS ***
- reduction factor is 0.987011893788
- general method check is 0.0592511723229
- MAXIMUM ULS LOAD IS 90.6542936541 N/m2

*** PLASTIC RESULTS ***
- reduction factor is 0.981462763014
- general method check is 0.0705383876245
- MAXIMUM ULS LOAD IS 107.923733066 N/m2
```

The creation of report files is switched off by default in UNIX based systems to save the disk space in long analyses. It can be turned on by calling **ABQ_Multiframe(…)** method with parameter **reports=1**.

3.2.5 Frame drawings

Drawing file *"drawing*-x.*svg"*, where the frame basic drawings (shape, cross-sections) are stored for all the frames from the current run (x is the number of input line), is also created during the calculation. This file can provide basic shapes for the designers. The open-source vector format (SVG) is used to make the drawing compatible with the most of vector graphics editors.

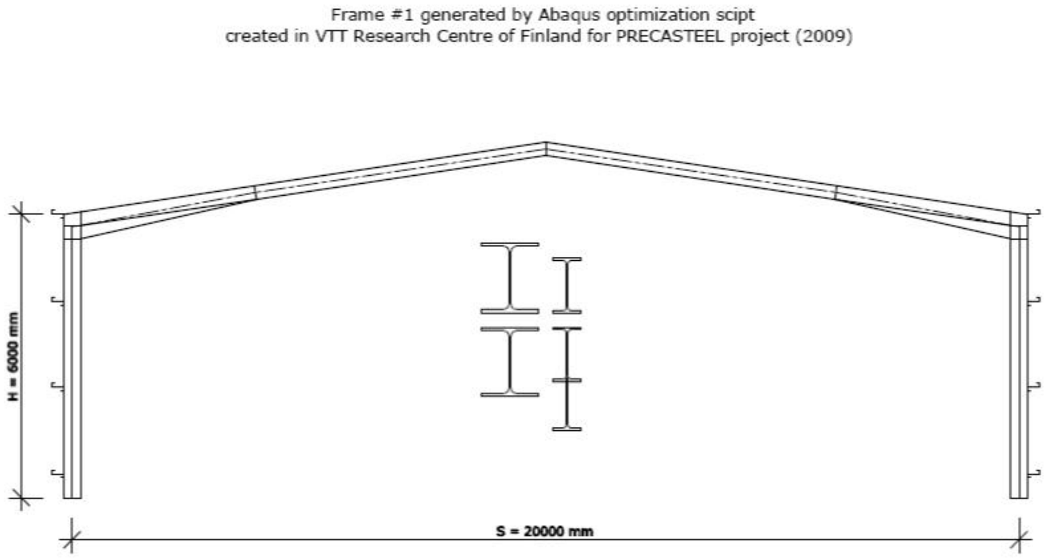


Figure 2. Example drawing automatically created by the python script.

The creation of drawings is switched off by default in UNIX based systems to save the disk space in long analyses. It can be turned on by calling **ABQ_Multiframe(…)** method with parameter **drawings=1**.

3.3 Load combinations

According to the load settings the script automatically selects if vertical or horizontal loads are to be increased gradually. If snow is the leading load, then the structure is preloaded with other accompanying loads (e.g. wind, crane load), and the snow load is gradually increased until failure. If e.g. earthquake is the leading load, then the structure is preloaded with snow, and the earthquake load is increased until failure.

In the fundamental load combination, snow load is considered to be the leading variable action and a combination of actions for persistent or transient design situation is calculated as follows [4]:

$$\gamma_{G,\sup} G_k + \gamma_S S_k + \gamma_W \psi_{0,W} W_k + \gamma_C \psi_{0,C} C_k \text{ (EN 1990 §6.10)} \quad 1$$

Where: G_k is the characteristic dead load ($\gamma_{G,\sup} = 1.35$)
S_k is the characteristic snow load ($\gamma_S = 1.5$)
W_k is the characteristic wind load ($\gamma_W = 1.5, \psi_{0,W} = 0.6$)
C_k is the characteristic crane load ($\gamma_C = 1.5, \psi_{0,C} = 1.0$).

For the horizontal pushover analysis with constant vertical load the seismic design situation applies [4]:

$$G_k + A_{Ed} + \psi_{2,S} S_k + \psi_{2,C} C_k \quad \text{(EN 1990 §6.12)} \qquad 2$$

Where: A_{Ed} is the design seismic load
and the combination factors are $\psi_{2,W} = 0.0, \psi_{2,S} = 0.2, \psi_{2,C} = 0.8$

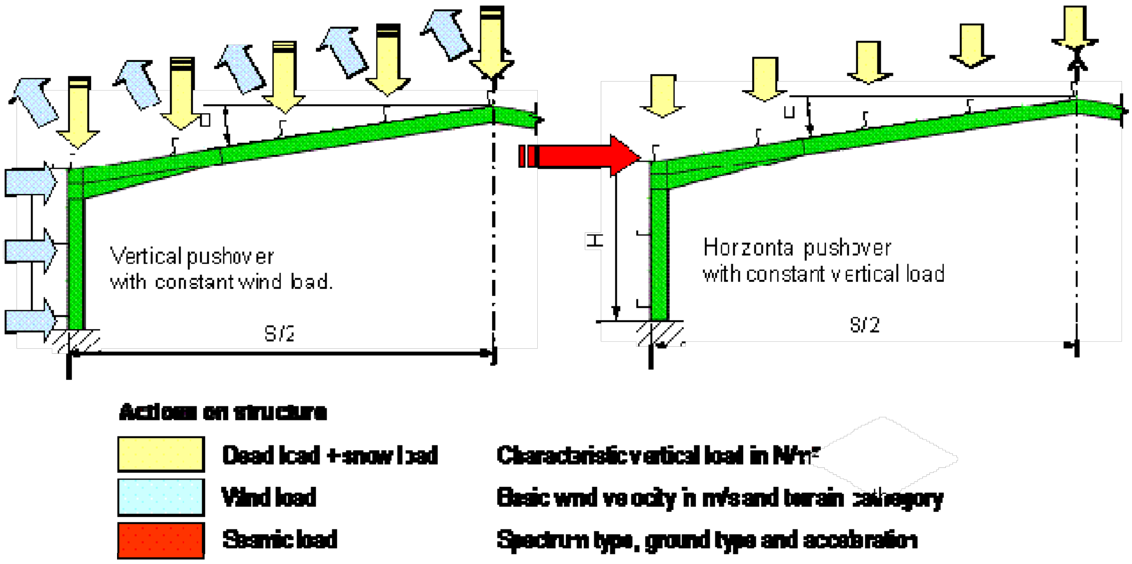


Figure 3. Loading scenarios.

A. Dead load and snow load

The design value of vertical snow and dead loads can be inserted in the input file in kN/m^2 (as distributed load). In case of dead load the value should also contain the estimation of the self-weight of the frame, including purlins and side-rails. This is necessary because as the mass of the frame is not transformed into weight in the model (i.e. gravity is 0). The snow load has to be decreased by combination factor $\psi_{2,S}$ when only seismic design situation is requested (e.g. no wind load is specified). For mixed load input (both wind and seismic load) the program instructions are automatically expanded into two situations: standard combination with wind and full snow load and seismic combination with seismic load and snow reduced automatically with combination factor $\psi_{2,S} = 0.2$.

B. Wind load

Calculation of wind actions follows EN 1991-1-4 [5]. The script uses simplified calculation of external wind pressure based on the exposure factor. The wind load spanning different zones is averaged and each face is loaded with constant pressure.

C. Seismic load

The seismic design loads are calculated out according to EN1998-1:2004 [8]. The masses, from vertical loads from Eq. 2 acting on the frame during earthquake, are concentrated in a single point at roof level (M_{eq}). I.e. the frame is transformed in a single degree of freedom (SDOF) oscillator.

Earthquake action is calculated using the lateral force method of analysis (§4.3.3.2 from [8]). The seismic base shear force F_b is calculated as:

$$F_b = S_d(T) \cdot M_{eq} \cdot \lambda \qquad 3$$

Where: $S_d(T)$ is the ordinate of the design spectrum
M_{eq} is the earthquake mass
λ is a correction factor for multi-storey buildings (λ = 1 is used in the script).

D. Crane load

The crane load is calculated according to EN 1991-3 [6]. User specifies only payload of the crane in the input file. The rest of crane parameters are automatically selected from crane specification file *"cranes.txt"* with the following structure:

Input name		Accepted values	Description
Crane num.	Number of the input	any value	Usually number of the input line.
Load (t)	Payload	positive number	If the requested load in tons doesn't exist in "*crane.txt*" the script uses the first higher load.
S (m)	Girder span (l in EC1)	positive number	Rail-to-rail distance in meters. Script searches for the best-fitted frame using the girder span and safety distance compared to the building interior clearance.
A1 (mm)	Safety distances	positive number	Crane safety distances in mm (see the drawing).
K1 (mm)		positive number	
C1 (mm)		negative number	
L1 (mm)		positive number	
L2 (mm)		positive number	
Zmin (mm)		positive number	Rail-to-column distance in mm.
Hmax (mm)		positive number	Maximum height of the hook in mm.
R (mm)	Wheel span (a in EC1)	positive number	Distance between crane wheels in mm.
Lk (mm)	Crane half-width	positive number	Distance from crane centre to bumpers in mm.
Rmax (kN)	Wheel loads	positive number	Maximum wheel load in kN. ($Q_{r,\max}$ in EC1)
Rmin (kN)		positive number	Minimum wheel load in kN. ($Q_{r,\min}$ in EC1)
Girders	Number of girders	whole number	Number of crane girders.

Self-weight Q_C of the crane without lifting attachment is approximately:

$$Q_c \approx 4Q_{r,\min} \qquad 4$$

Hoist load Q_H includes the masses of the payload, the lifting attachment and a portion of the suspended hoist ropes or chains moved by the crane structure:

$$Q_H \approx 2Q_{r,\max} + 2Q_{r,\min} - Q_C \tag{5}$$

Drive force calculation assumes steel-to-steel contact:

$$K = \mu \sum Q_{r,\min} = 0{,}2 \cdot 2 \cdot Q_{r,\min} \tag{6}$$

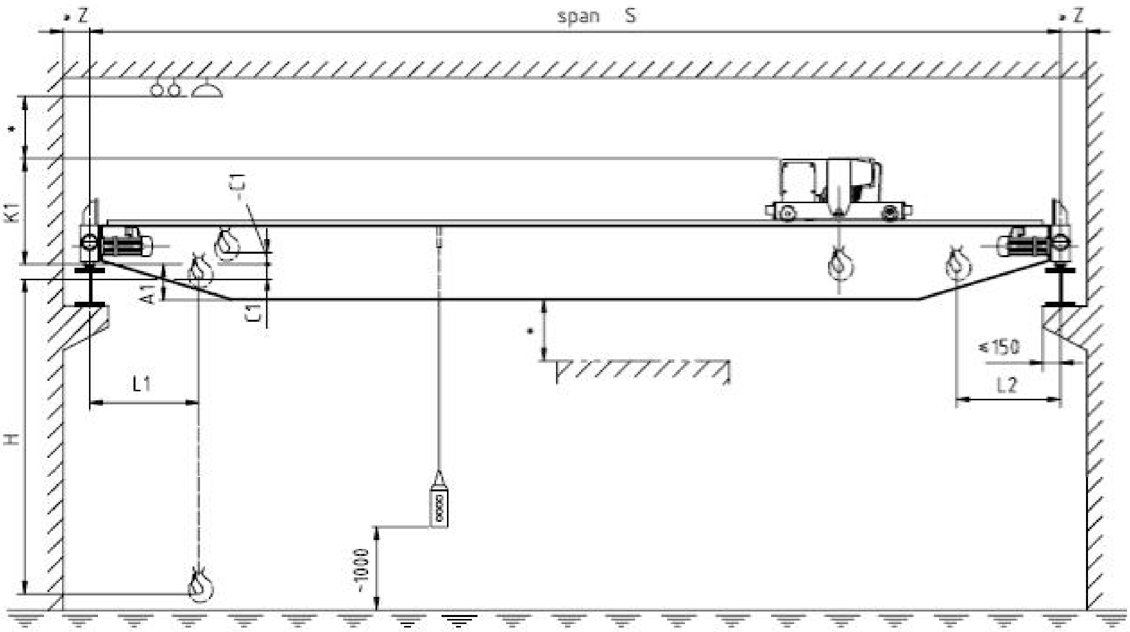


Figure 4. Safety distances used in crane specification file.

Transverse loads are calculated as:

$$H_T = \max(\xi_1,\xi_2)\frac{M}{a} = \max(\xi_1,\xi_2)\frac{Kl_S}{a} = \max(\xi_1,\xi_2)\frac{K(\xi_1 - 0{,}5)l}{a},$$

where $\xi_1 = \dfrac{Q_{r,\max}}{Q_{r,\max} + Q_{r,\min}}$ and $\xi_2 = 1 - \xi_1$.

Skew loads are calculated as:

$$H_S = \lambda_S f \cdot Q_{r,\max} = \max(\xi_1,\xi_2)\left(1 - \frac{a}{2L_k}\right) f \cdot Q_{r,\max},$$

where $f = 0{,}3$.

Design vertical and horizontal load is calculated using following simplified equations:

$P_{vert,\max} = \varphi_1 Q_C + \varphi_2 Q_H$ maximum vertical load,

$P_{vert,\min} = \varphi_1 Q_C$ minimum vertical load,

$P_{hor} = \max(\varphi_5 H_T ; H_S)$ horizontal load, where dynamic factors are $\varphi_1 = 1{,}1$, $\varphi_2 = 1{,}99$, $\varphi_5 = 1{,}5$.

3.4 Numerical models

The wire model of the frame is created using Abaqus/CAE objects and methods in Python script. Methods "**Shapes()**" and "**ShapesMid()**" create the centre line where each straight segment representing tapered member is divided into *nd* parts (see Script structure - Global variables). Additional points are inserted in positions of purlins and crane cantilevers.

Object/Method	Parameters	Description
Model	dimensionality=TWO_D_PLANAR type=DEFORMABLE_BODY	Model properties
Material	Elastic(table=((210000.0, 0.3),))	Material properties
BeamSection	integration=DURING_ANALYSIS	Section properties
Tie	positionToleranceMethod=SPECIFIED positionTolerance=5.0	Connection of two parts
DisplacementBC	u1=SET, u2=SET	Pinned support
DisplacementBC	u1=SET, u2=SET, ur3=SET	Fixed support
ConcentratedForce	cf2=vertical, cf1=horizontal	Point loads
seedPart	deviationFactor=0.1	Seeds for meshing

The shell model of the frame is also created using Abaqus/CAE. Methods "**Shapes()**" and "**ShapesMid()**" create the shape of the web. Flanges extrude from the outline after the web is completed. Spring/dashpots elements support upper flange laterally at each purlin where connections between purlin supports and flanges are modelled as short wires. We use the same points also for loading of the frame. Base supports and crane cantilever beams are modelled as rigid bodies.

Object/Method	Parameters	Description
Model	dimensionality=THREE_D type=DEFORMABLE_BODY	Model properties
Material	Elastic(table=((210000.0, 0.3),))	Material properties
BeamSection	integration=DURING_ANALYSIS	Section properties
Tie	positionToleranceMethod=COMPUTED	Connection of parts
DisplacementBC	u1=0.0, u2=0.0, u3=0.0, ur1=UNSET, ur2=0.0, ur3=UNSET	Pinned support
DisplacementBC	u1=0.0, u2=0.0, u3=0.0, ur1=UNSET, ur2=0.0, ur3=0.0	Fixed support
ConcentratedForce	cf2=vertical, cf1=horizontal	Point loads
SpringDashpotToGround	dof=3 springBehavior=ON springStiffness=100000000000.0	Lateral support
seedPart	deviationFactor=0.1	Seeds for meshing
ElemType	elemCode=S4	Shell elements
setMeshControls	elemShape=QUAD technique=STRUCTURED	Mesh of rectangular web and flange faces
setMeshControls	elemShape=QUAD_DOMINATED technique=FREE	Mesh of triangular haunch face

3.5 Parallel processing and monitoring of multiple calculations

In order to achieve the best performance of a calculation of multiple frames, several jobs are submitted and evaluated at the same time. Each independent thread where calculations are submitted is called PROCESS. Inputs submitted to a single process are called FRAMES and are divided into several PHASES of calculation (Figure 5). The computer memory is redistributed according to the number of currently opened processes. When all jobs are submitted, script switches to a CONTROL LOOP where is monitoring the progress of calculation and terminating jobs if necessary.

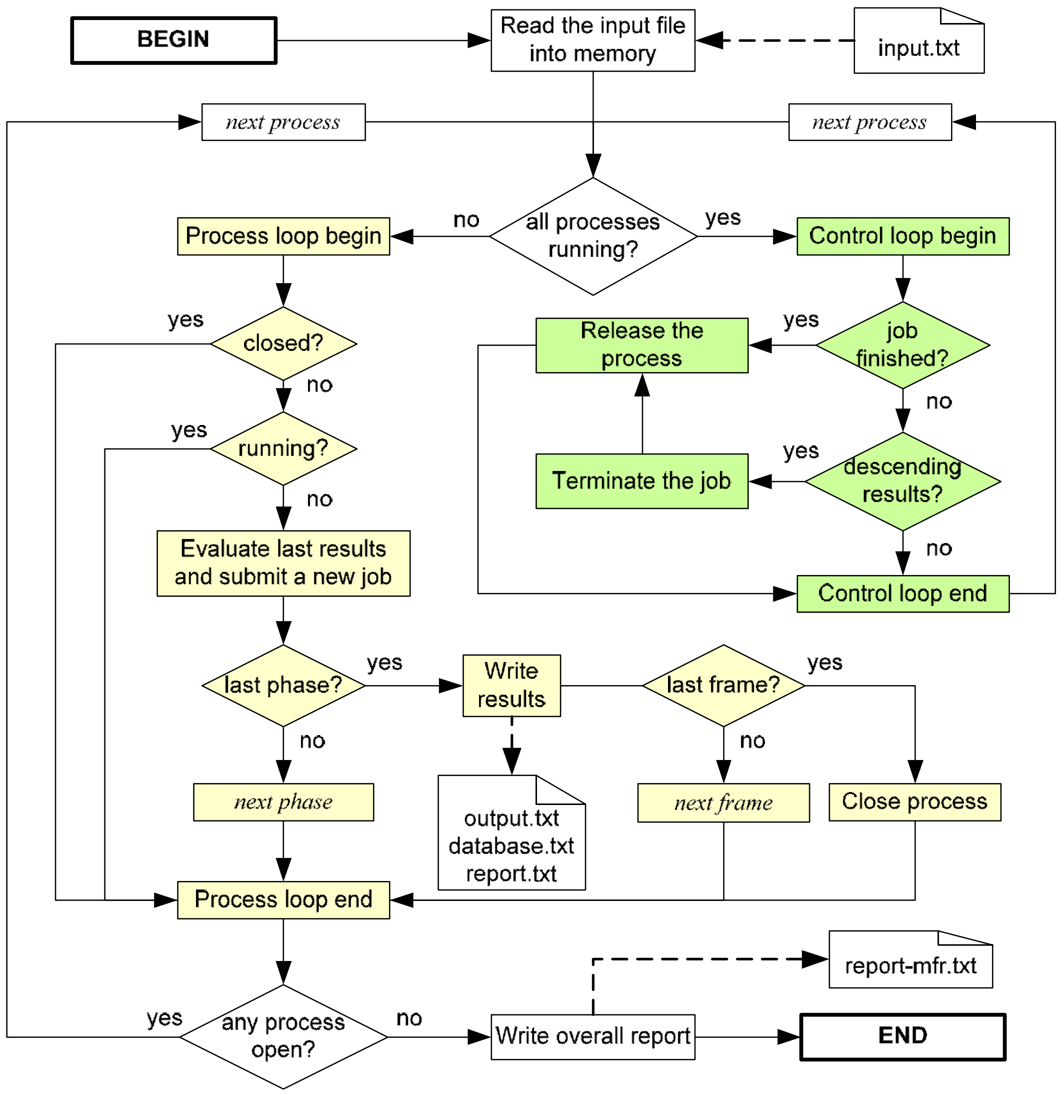


Figure 5. Abaqus script parallel processing.

In order to increase the speed of calculation script includes a method of terminating the Riks analysis. The point where we have already enough data collected is the maximum load and it is usually recognized by three consequent descending steps. Standard monitoring of running jobs implemented in Abaqus/CAE cannot be used because it is not stable when monitoring several jobs at the same time.

The control method is checking for existence of status file (e.g. *"Job-1.sta"*) and lock file (e.g. *"Job-1.lck"*) which indicates that the calculation is still running. Load step increments are obtained by reading the contents of status file. In UNIX based systems simple deleting of all job files terminates the running analysis immediately. On the other hand, the running *"standard.exe"* process has to be terminated in Windows. Both methods proved to be stable for most of the calculations. In case of unexpected problems (licence errors, database is corrupted and so on), the calculation is automatically restarted. When no results are obtained within a given time-out limit, the whole analytical part is restarted and frame variables are re-read from the input file to memory.

3.6 Script structure

The following paragraphs describe briefly the structure of *"mframe*xxx*.py"* script. The script is importing modules from external libraries as "abaqus", "abaqusConstants", "caeModules", "time", "os", "sys" and "socket". These libraries have to be available in order to compile or execute the script.

A. Parameters

Several parameters can be passed to the calculation when calling the main method (e.g. "**ABQ_MultiProcess(inp='input.txt', parallel=1**)". If they are not set, the default value is used.

Identifier	Name	Default value	Comments
inp	Input file name	"input.txt"	
out	Output file name	"output.txt"	
dat	Database file name	"database.txt"	The file must be in the same folder as the script. It has to contain at least the heading of a standard database.
parallel	Number of parallel threads	1	By default, the script is called with parallel=1 in Windows based systems. In UNIX based systems it is suggested to put the number of CPU cores. User needs also one Abaqus/Standard licence for each thread.
reports	Create report files	0	0 – off, 1 – on
drawings	Create SVG drawings	0	0 – off, 1 – on

B. Global variables

Global variables are not changed during the analysis and they are set in the beginning of the script. The most important settings are in the following table.

Identifier	Type	Value	Comments
memMax	integer	90	percent of memory to be redistributed
maxDecreasingRiks	integer	3	number of decreasing steps in Riks analysis before termination
delay	integer	5	delay (in seconds) of the control loop when all processes are busy
timeout	integer	2000	seconds to restart the analysis if no results are produced

offPEEQ	float	5.0	the equivalent plastic strain offset (the multiplier of elastic strain to reach the plastic hinge in wire model)
nd	integer	20	number of divisions in tapered parts of wire model

C. Classes

In object-oriented programming, classes are particularly important because they define variables and methods specific for a given object.

Name	Description
ProcessClass	The object created for each parallel thread carries information of the actual phase of calculation and links to the other objects associated with the current frame assigned to the ProcessClass. "**submit()**" is the internal method where sway imperfections are applied, models are saved and jobs are submitted.
InputClass	Input objects are created instantly when the scripts starts by reading and interpreting the input file. The internal method "**readLine()**" is important because the load calculation and geometry interpretation is included in this method. For every line of the input file, one object is constructed.
DataClass	As the complementary object to InputClass, data object carries containers for information to be stored in *"output.txt"* and *"database.txt"*. The containers are filled during the calculation. The "**isInDatabase()**" method compares input object with the database records and decides whether the calculation can be skipped. All outputs are written into files using "**writeDatabase()**" method.

D. Process-handling methods

Name	Description
ABQ_MultiProcess	Reads the input file and creates objects. Starts the main program loop. Assigns frames to processes. Writes the output files.
ControlP	Loops the "**Control()**" method for all open processes.
Control	Determines if the process can be released. Terminates the job if necessary.
anyRunning	Returns True if there are any processes open (False stops the main loop).
allSubmitted	Returns True if there are all processes waiting for results.

E. Analysis methods

Name	Description
ABQ_EQ1	**Seismic design situation (Phase 1)** Loads the frame with unit (1kN) horizontal load and starts static analysis in order to determine the column stiffness.
ABQ_EQ2	**Seismic design situation (Phase 2)** Calculates the column stiffness and seismic load.
ABQ_GA1	**Global nonlinear analysis (Phase 1)** Starts buckling analysis with shell model. Repeats the task until the first positive eigenvalue is reached.
ABQ_GA2	**Global nonlinear analysis (Phase 2)** Starts pushover analysis (3D shell model) with perturbed shape

	obtained form the previous phase.
ABQ_GA3	**Global nonlinear analysis (Phase 3)** Evaluates pushover results.
ABQ_GM1	**General method (Phase 1)** Starts out-of-plane buckling analysis with shell model. Repeats the task until the first positive eigenvalue is reached.
ABQ_GM2	**General method (Phase 2)** Starts in-plane buckling analysis with wire model.
ABQ_GM3	**General method (Phase 3)** Starts pushover analysis (2D wire model) with perturbed shape obtained form the previous phase.
ABQ_GM4	**General method (Phase 4)** Evaluates pushover results. Calculates resistances using critical multiplier from the first phase.
readRiks	Reads the database and interpolates for the limit points. The method is called by both (wire and shell) pushover methods.
redFact	Calculates reduction factor based on the European buckling curves.

F. Modelling methods

Name	Description
ABQ_WireModel	Creates the wire model
ABQ_ShellModel	Creates the shell model
Wire_Load	Loads the wire model
Shell_Load	Loads the shell model
Shapes	Calculates the shape of frame outer part and its centre line
ShapesMid	Calculates the shape of frame central part (in double-span frames) and its centre line
Drawing	Outputs a drawing in SVG format

G. Input/Output handling methods

Name	Description
IOdecompose	Expands lines in input.txt file where the mixed design situation is requested. (the first line is the same - script executes in seismic mode and ignores the wind load, the second line is without seismic load)
IOcompose	Shrinks double outputs in *output.txt* to match the number of original input file *input.txt*.

3.7 The optimization script using Python

An optimization process was developed around AP-Frame script to optimize portal frames using genetic algorithms (GA). Genetic algorithm is population based and it tries to find the optimal configuration favouring good solutions against bad ones. Optimization starts with a random population of portal frame configurations, which is evaluated and then modified according to genetic algorithm procedure. This new modified population is evaluated and genetic operations are performed again. The algorithm proceeds iteratively towards optimal solution while always storing the best solution found. Each population of portal frames is sent to evaluation by writing the "*input.txt*" file described earlier, and after the completion of the structural analysis part, the output is read from *“output.txt”* file. Therefore, form the point of view of the genetic solver, the file *“output.txt”* contains the input for the GA, while in the *“input.txt”* the output is written after each GA step.

3.7.1 Executing the optimization script

The optimization script needs different files for optimization depending if frames are manufactured of welded tapered or hot-rolled sections. Optimization script for hot-rolled sections can be initialized, as follows:

- from the system command line - write “**abaqus cae noGUI=GA_XX.py“.**
- from the Abaqus/CAE GUI - select “**File-Run Script…**” (**Alt+F**, **Alt+R**) and choose *“GA_XX.py”*.

Frames with welded-tapered sections can be optimized by replacing *GA_XX.py* with *GA_W_XX.py*. The script files *GA_XX.py* and *GA_W_XX.py* contain the genetic algorithm procedures, but analysis script *mframeXXX.py* is needed for design verification of portal frames.

3.7.2 Inputs

Configuration files needed for hot-rolled frames:

- *GA_start_file.txt* - File containing genetic algorithm parameters and frame configurations (1…99 supported) for optimization
- *EU_profiles.txt* - File containing hot-rolled profiles available in the EU
- *GA_output0.txt* - File needs to contain the header information for GA output
- *database.txt* - File for storing calculated frame configurations

Configuration files needed for welded-tapered frames:

- *GA_W_start_file.txt* - File containing genetic algorithm parameters and frame configurations (1…99 supported) for optimization
- *GA_W_output0.txt* - File needs to contain the header information for GA output
- *database.txt* - File for storing calculated frame configurations.

The main configuration file for hot-rolled frame optimization is *GA_ start_file.txt*; which contains the genetic algorithm parameters and a list of frame configurations to be optimised. This start file contains GA parameters on the first five lines and from lines 10-14 there is the header in of the input file, which is used to create the *input.txt* needed for the analysis script. After the header user can input the desired frame configurations with a text editor. The GA script supports up to 99 configurations. These frame configurations are optimized with the same GA settings defined in the first part of the file.

Input file "*GA_start_file.txt*" specification			
Input name		Values accepted	Description
Gen. size.	Generation size	Higher than 1	Amount of frames in each generation, i.e. population size
Generations	Maximum generations	positive number	Amount of generations run before the algorithm is terminated
Sel. met.	Selection method	*tour* or *RRWS*	Type of selection operator, tournament selection recommended (*tour*)
Xover prob	Probability of crossover	0...1.0	Probability of a frame participating in crossover operation, 0.8 recommended
Xover met.	Crossover method	*SBX* or *Laplace*	Type of crossover operator, simulated binary crossover (*SBX*) recommended
SBX n_c	SBX parameter	positive number	Parameter for determining the spread of SBX operator
Xover var	Variable crossover probability	0.5…1.0	Probability of a single variable participating in crossover
Mut prob	Mutation probability	0.001…0.2	Probability of a single variable mutation
Mut. met	Mutation method	*Poly* or *Power*	Type of mutation operator, polynomial (*Poly*) recommended
Mut. n_r	*Poly* mutation parameter	positive number	Parameter for determining the spread of polynomial mutation operator
H. ratio	Height ratio	higher or equal to 2	Maximum ratio between beam and column profile in initial population
Initial pop.	Pre-defined initial population	*True* or *False*	Default is *False*, which means that initial population is random
Beam rest.	Beam profile restriction	*IPE, IPE_BASIC, HE_AB, BASIC, ALL*	Restricts profile selection for the beam
Column rest.	Column profile restriction	*IPE, IPE_BASIC, HE_AB, BASIC, ALL*	Restricts profile selection for the column
Profile order	Profile order	*A, Wy, Iy*	Order of the profile database
Pop. r. size	Population randomize size	< Population size	Amount of random frames inserted in to the population, if frequency condition met
Pop. r. freq	Population randomize frequency	Higher than 10	Amount of generations elite value has to stay the same for randomizing to occur

GA start file for the welded-tapered frames has only a few differences to the hot-rolled version. The options related to the profile selection are removed, but otherwise the table above applies.

Possible variables for hot-rolled frames are:
- Beam profile
- Column profile
- Haunch ratio
- Roof angle.

Variables for optimization can be chosen by leaving the corresponding column on the GA start file empty.

Variables for welded-tapered frames:
- Height of the haunch
- Height of the column at the bottom
- Height of the beam at the top
- Width of the profiles
- Flange plate thickness
- Web plate thickness
- Haunch ratio.

Haunch ratio can also be set to a constant value by defining it in the start file.

3.7.3 Outputs

Files created during the optimization for hot-rolled frames:
- *EU_profiles_sorted.txt* - Database of EU profiles sorted by desired criteria, e.g. profile area
- *EU_profiles_sorted_beams.txt* - Database of profiles used for beams, can be restricted, for example, to most common sections
- *EU_profiles_sorted_columns.txt* - Database of profiles used for columns, can be restricted, for example, to most common sections
- *report-mfr.txt* - File containing information about progress of the algorithm
- *GA_outputX.txt* - Only the header from initial *GA_output0.txt* remains and *GA_outputX.txt* (X = 0,1,2,3…) refers to the optimization output of the frame configuration sent to optimization.

Files created during the optimization for welded-tapered frames:
- *GA_W_outputX.txt* - Only the header from initial *GA_output0.txt* remains and *GA_W_outputX.txt* (X = 0,1,2,3…) refers to the optimization output the frame configuration sent to optimization
- *report-mfr.txt* - File containing information about progress of the algorithm.

Output files for hot-rolled and welded-tapered frames are quite similar. They contain variable data of each frame configuration calculated during the optimization run. Weight and feasibility of the frames are also visible, where negative number in feasibility column means that the frame is failing. Feasibility values near zero meaning that the frame capacity is almost fully utilized.

4 EV-Frame - the VBA script using Microsoft Excel

Microsoft Excel optimization tool (EV-Frame) was developed to cover the same inputs as the Abaqus tool (AP-Frame). EV-Frame uses 1st order elastic calculation to determine internal forces, and buckling reduction factors for performing design checks. The frame is modelled with elastic beam elements only (even though sway imperfections are taken into account). For this reason it is much faster calculation tool then AP-Frame, which is based on 2nd order nonlinear calculations.

In our study, EV-Frame was mainly used for benchmarking and testing convergence of optimization algorithms. Its level of conservativeness in the design is higher than that of AP-Frame. Since it is conservative in design (i.e. it predicts lower load bearing capacity & produces larger frames), it is considered not a perfect tool use to optimize frames.

The structural analysis part of EV-Frame is modification of open-source software [9], and under the terms of its license the following statement should be published with the source code of the program:

This program is free software: it can be redistributed and/or modified under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version. This program is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for particular purpose (See the GNU General Public License for more details http://www.gnu.org/licenses/).

4.1 Loading and imperfections

In case of the beam model of the frame, the rafter is loaded with uniformly distributed vertical load (e.g. dead load, snow), and the frame corners can be loaded with concentrated horizontal forces (e.g. wind and earthquake). Load combinations are according to EN-1990 [4], as presented in Chapter 3.3.

In order to take into account sway imperfection, an initial inclination of the frame is considered by moving the each node's x-coordinate to:

$$x_{imp} = x + y \cdot \tan\phi \,, \qquad 7$$

Where: x is the horizontal coordinate of the node before imperfection is applied

x_{imp} is the horizontal coordinate of the node after imperfection is applied

y is the vertical coordinate of the node

ϕ inclination of the frame from vertical ($\phi = \frac{\alpha_h \cdot \alpha_m}{200}$, EN 1993-1-1 [6], §5.3.2).

4.2 Single frame calculation

For the calculation of one particular frame, the "**FRAME**" worksheet is prepared. Frame parameters defined by user can be entered in white cells in the following form. Input variables correspond to those used in Abaqus script.

	user input	program choice
Type of the frame	**welded**	1
Column is	**pinned**	1
Steel grade	**S275**	275
Span (m)	**20**	20300
Height of the frame (m)	**6**	6000
Distance between frames (m)	**6**	6000
Number of purlins on beam	**9**	9
Number of purlins on column	**5**	5
Haunch ratio (S/L_h)	**5.5**	5.5
Roof angle	**15%**	0.15
Dead load (kN/m^2)	**0.38**	0.38
Snow load (kN/m^2)	**0.75**	0.75
Horizontal load on frame (kN)	**0**	0
For hot-rolled frames specify		
Hot-rolled beam	IPE 500 A	93
Hot-rolled column	IPE 360	68
For welded frames specify		
Height of the haunch (mm)	**800**	800
		800
Height of the column at the base (mm)	**300**	300
Height of the beam at the top (mm)	**600**	600
Width of the frame (mm)	**240**	240
		240
Flange thickness (mm)	**16**	16
		16
Web thickness (mm)	**6**	6
		6

Weight of the frame	**2.950 t**	
Weight of the steel structure	**29.53 kg/m2**	(frame and purlins)
Surface area of the frame	**70.59 m2**	
Cost of the frame	**7 552 EUR**	
Recalculate		
	ULS	**Utilization**
Design load - ULS	**1.64 kN/m2**	**0.79**
Characteristic load - SLS	**1.13 kN/m2**	**0.62**
Maximum design load from ULS	**2.07 kN/m2**	
Maximum design load from SLS	**1.82 kN/m2**	

Figure 6. Single frame calculation.

Evaluation of a frame can be started by pressing "**Recalculate**" and the results in yellow area will be refreshed. The frame is considered to be feasible (satisfying all limit states conditions) when both utilization rations are smaller than 1.

Looking at the "**Input0**" worksheet user can find detailed information about the calculation and a tool for purlin design (right). The purlin design uses Eurocode 3 method for calculation of cold-formed purlin. Purlin span is defined in the "FORM" sheet as the distance between two frames. Database of available cold-formed profiles is stored in "**Purlins**" worksheet.

Purlin design (EC3 1-3)			purlins in catalogue	29	
t	1.5	mm	overlapping	10.00%	
h	150	mm			
b	41	max. 90 mm			
c	18	max. 75 mm	**Z150/1.5**		
A	401	mm^2	weight	3.149	kg/m
I_y	1347492	mm^4			
$I_{eff,y}$	1256367	mm^4			
$W_{eff,y}$	16752	mm^3			
A_{fz}	133	mm^2			
I_{fz}	79091	mm^4			
i_{fz}	24	mm			
W_{fz}	1929	mm^3			
k_{h0}	0.124			figure 10.3	
k_h	0.124			figure 10.3	
$q_{h,Ed}$	0.342	kN/m (N/mm)		(10.4)	
K	1000			estimation	
R	801049			(10.6)	
Π_1	0.694			table 10.2a	
Π_2	5.450			table 10.2a	
Π_3	1.270			table 10.2a	
Π_4	-0.168			table 10.2a	
l_{fz}	172	mm	buckling lengt	(10.9)	
Π_1	70.25				
Π_{fz}	0.101			(10.8)	
Π_{LT}	0.340		buckling c. b	10.1.4.2	
$\Pi_{LT,0}$	0.400			10.1.4.2	
Π	0.750			10.1.4.2	
Π_{LT}	0.453				
Π_{LT}	1.000				
	middle	edge			
Π_R	-0.063	0.093		table 10.1	
$M_{0,fz,Ed}$	512588	-1025176	Nmm	table 10.1	
$M_{fz,Ed}$	32357	95546	Nmm	(10.5)	
$M_{y,Ed}$	3483320	8957109	Nmm	(10.3) (10.7)	
ULS	**0.535**	**0.696**	**TRUE**	(10.3)	
SLS	**0.559**				

Figure 7. Purlin design.

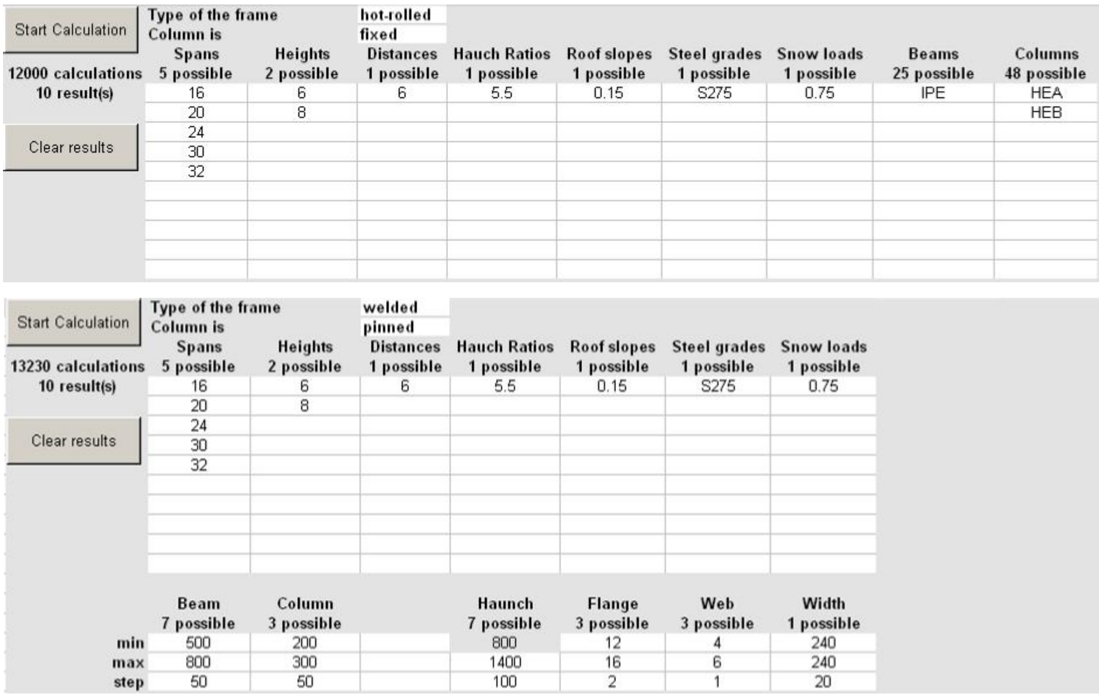
4.3 Multiple frame calculation

The method used in EV-Frame script is very fast, so it is also possible to find an optimal configuration by calculating all possible beam and column combinations for a given frame configuration. In this case, the user defines a few parameters, e.g. the span, height and load, and the script checks one-by-one frames made with each beam and column cross section. The lightest frame which satisfies the design is returned as optimum.

As this procedure does not use any algorithms to search for an optimum, but it evaluates all the possible search space (i.e. all possible configurations are checked), it can be very costly in terms of run time. This feature can be useful, and it has been used mostly, as basis of comparison for the speed and efficiency of more sophisticated optimization algorithms.

The user can enter required variables into white cells in "**CALC**" worksheet and start the script by pressing "**Start Calculation**" button. The total number of calculation executed is displayed under the button.

The result is written in the output part of the sheet for all of variable combinations. When hot-rolled frame selected, the script is using database of EU profiles stored in "**Profiles**" sheet. For welded frames some additional information has to be entered (see picture). User specifies all profile's dimensions by three variables - minimum, maximum and step size.



Start Calculation

Type of the frame: hot-rolled
Column is: fixed

12000 calculations
10 result(s)

Clear results

	Spans	Heights	Distances	Hauch Ratios	Roof slopes	Steel grades	Snow loads	Beams	Columns
	5 possible	2 possible	1 possible	1 possible	1 possible	1 possible	1 possible	25 possible	48 possible
	16	6	6	5.5	0.15	S275	0.75	IPE	HEA
	20	8							HEB
	24								
	30								
	32								

Start Calculation

Type of the frame: welded
Column is: pinned

13230 calculations
10 result(s)

Clear results

	Spans	Heights	Distances	Hauch Ratios	Roof slopes	Steel grades	Snow loads
	5 possible	2 possible	1 possible	1 possible	1 possible	1 possible	1 possible
	16	6	6	5.5	0.15	S275	0.75
	20	8					
	24						
	30						
	32						

	Beam	Column		Haunch	Flange	Web	Width
	7 possible	3 possible		7 possible	3 possible	3 possible	1 possible
min	500	200		800	12	4	240
max	800	300		1400	16	6	240
step	50	50		100	2	1	20

Figure 8. Multiple frame calculation form for hot-rolled frames (top) and welded frames (bottom).

4.4 Optimization

Optimization using genetic algorithms is implemented in Visual Basic for Applications, and objective function is calculated using the frame calculation sheets as described earlier chapters.

The optimization for hot-rolled frames can be started from "**GA**" sheet, while welded-tapered sections are covered in "**GAW**" sheet. Most important outputs of each frame analyzed each generation are stored in "**GA_output**" and "**GA_W_output**" sheets, for hot-rolled and welded-tapered frames, respectively. Multiple runs of the genetic algorithm with the same parameters can be performed by setting the cell next "*Runs (for benchmarking)*" to desired number of runs. The elite value found in each consecutive run will be saved in "**GA_runs**" or "**GA_W_runs**" sheet.

Figure 9 and Figure 10 illustrate the basic functionality available in "**GA**" sheet. "**GAW**" sheet has the same functionality with minor visual changes, as the beam and column have a different format. Default settings for the genetic algorithm parameters can be forced by pressing the “Set default parameters” button in “**GA**” or “**GAW**” sheet.

The most important genetic algorithm parameters are the population (generation) size and maximum number of generations. These two parameters affect the optimization time and performance. In general, longer runs of the algorithm, i.e. larger population size and/or maximum number of generations, are more likely to find good results. Other genetic algorithm parameters also affect the optimization and some gains can be achieved by adjusting these parameters. However, default GA parameters (Figure 9) should give satisfactory performance.

Objective	Weight
Genetic algorithm parameters	
General parameters	
Population size	30
Maximum generations	50
Randomize seed (0=time)	0
Population randomize freq.	50
Population randomize size	15
Selection operator	
Selection type	Tournament
Tournament size	2
Penalty parameter	50
Crossover operator	
Crossover probability	0.8
Crossover method	SBX
SBX parameter (spread)	0.1
Crossover diversify	0.5
Laplace parameter small b2	0.1
Xover var. prob.	0.5
Mutation operator	
Mutation probability	0.05
Mutation type	Polynomial
Polyn. Mut. Param.	5
Power mut. Param.	4

Function calls	1500
Time for function evaluation (s)	0.15
Total time (h)	0.06
Frame dimensions	
Column is pinned=1 / fixed=2	1
Span	24000
Height of the frame	6000
Distance between frames	6000
Number of purlins on beam	7
Number of purlins on column	5
Minimum haunch ratio	4
Maximum haunch ratio	11
Minimum roof angle (%)	15
Maximum roof angle (%)	15
Steel grade	275
Runs (for benchmarking)	1
Run counter	1
Set default parameters	
Start optimization	
Generation count	
	50

Figure 9. GA sheet containing genetic algorithm parameters and frame dimensions.

Frame dimensions can be modified from the genetic algorithm optimization sheets. If constant roof angle or haunch ratio is desired, the maximum and minimum value of the corresponding variable should be set equal. This will transform the variable into a constant. In addition, number of objective function evaluations, i.e. frames calculated, is presented in the sheet. By changing the approximation of time needed for the analyzing a single frame, the total time for one optimization run can be estimated.

During the optimization current population and elite value are displayed in the sheet, as shown in Figure 10.

Elite individual				
Fitness	**Beam**	**Column**	**Haunch ratio**	**Roof angle**
3.8754882	86	87	5	15
	IPE 500	IPE 550 A		
	Weight	**Cost**	**kg/m2**	
	3.8754882	9656.80276	30.45191797	
Current generation				
Fitness	**Beam**	**Column**	**Haunch ratio**	**Roof angle**
3.8754882	86	87	5	15
3.8754882	86	87	5	15
5.071101304	86	91	10	15
5.868014522	96	24	10	15
5.737464729	86	38	5	15
5.747951813	75	117	5	15
5.314238002	61	87	5	15

Figure 10. GA sheet containing the elite value and population members.

The optimization process can be observed by following the development of the elite value, which is displayed as a function of generation (Figure 11).

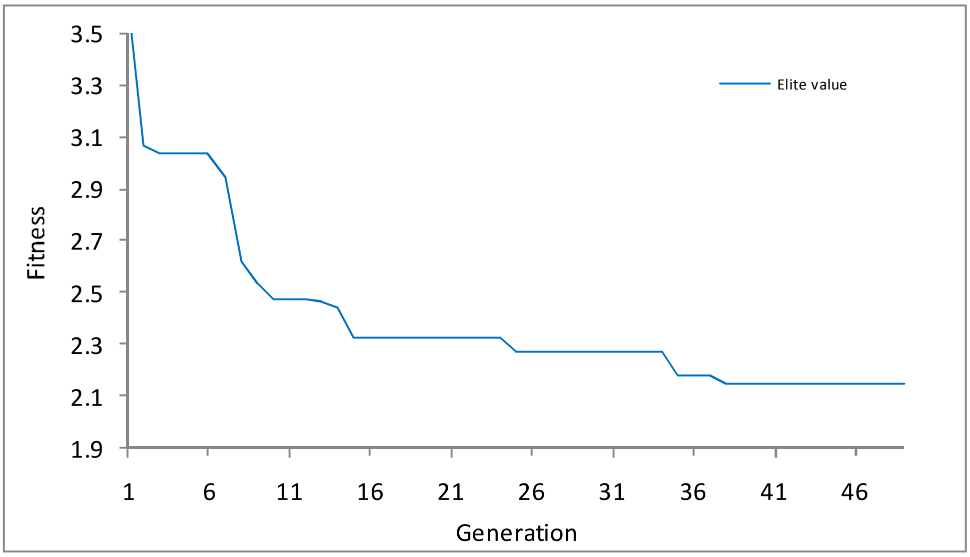


Figure 11. Typical development of the elite value during the optimization run.

5 Summary

The user side of two optimization tools (AP-Frame and EV Frame) were described in this document. The description is fairly detailed, and should enable a future user to: set-up inputs, run the scripts and understand the output produced by the scripts.

Some insides on the structure of the scripts was also given in order to enable the understanding of the programming features used. This should enable advanced users or programmers to review and modify the scripts.

References

[1] ŠAPALAS, V., SAMOFALOV, M., VIAČESLAVAS, Š FEM stability analysis of tapered beam-columns, *Journal of Civil Engineering and Management*, 2005, Volume XI., No. 3, 211-216.

[2] G. VARELIS, D. VASILIKIS, S. KARAMANOS, P. TSINTZOS: WP2: Performance analysis of selected steel types for one-storey industrial buildings, Research report.

[3] FÜLÖP, L., BEAUCAIRE, P., Advanced analysis of the performance of steel frames, *VTT Research report*, Espoo, Finland, 2009 (VTT-R-10493-07).

[4] EN 1990:2002, Eurocode: Basis of structural design.

[5] EN 1991-1-4:2005, Eurocode 1: Actions on structures - Part 1-4: General actions - Wind actions.

[6] EN 1991-3:2006, Eurocode 1: Actions on structures - Part 3: Actions induced by cranes and machinery.

[7] EN 1993-1-1:2005, Eurocode 3: Design of steel structures - Part 1-1: General rules and rules for buildings.

[8] EN 1998-1:2004, Eurocode 8: Design of structures for earthquake resistance – Part 1: General rules, seismic actions and rules for buildings.

[9] JENKINS, D., *Frame analysis with Excel*, http://newtonexcelbach.wordpress.com/, Interactive Design Services Pty Ltd., Sydney, Australia, 2009.