AXISVM

AUTO M_{cr} GUIDE

Edited by: Inter-CAD Kft.

TABLE OF CONTENTS

PART 1. THEORETICAL BACKGROUND

I. INTRODUCTION

AutoMcr is an application used in the Steel Design module to calculate the elastic critical moment (M_{cr}). M_{cr} is required in the calculation of lateral torsional buckling resistance. AutoMcr creates an individual finite element submodel of each steel design element, for which it determines the M_{cr} value by solving an eigenvalue problem. The submodel is built-up of special beam finite elements only with those degrees of freedom that are relevant for lateral torsional buckling:

- v lateral displacement, in the direction of local y axis;
- θ_x torsion: rotation about beam axis / local x axis;
- θ_z rotation about weak axis / local z axis;
- w warping.

When creating the submodel, the program automatically identifies lateral supports, which can be edited by the user. The rigidity components of the support, indexed according to the local coordinate system of the submodel: R_y, R_{xx}, R_{zz}, R_w.

The AutoMcr is based on the same theory as the LTBeam program, of which further information can be read in the following article: *Yvan Galea: Moment critique de deversement elastique de poutres flechies presentation du logiciel ltbeam* [1].

This Guide has two main goals. In Part 1, examples demonstrate the possibilities and limits of AutoMcr, while helping users to properly use the program. Part 2 is a summary of verification models, in which results of AutoMcr are compared to literature and to other programs. For basics of the AutoMcr method and to learn how to use it, check *AxisVM13 User's Manual: 6.6.2. Steel beam design based on Eurocode*.

The AutoMcr is capable of analysing straight elements with a cross section symmetric at least about the weak axis. Moreover, it can handle:

- elements with variable cross-section;
- cantilevers: no need to define if it is a cantilever or not, as in AxisVM12;
- eccentric load: distance from the weak axis, one value for all load cases analysed at a time;
- eccentric support conditions: defined individually for each support.

The AutoMcr method handles only continuous elements, therefore it splits up design members in the following two cases:

- tapered beam: when part of the beam has variable cross-section, the rest is constant;
- elements with intermediate pin.

II. LATERAL SUPPORTS

With default settings, the Auto Mcr method automatically determines the lateral supports of the designed member; which will be detailed in the following. The program finds not only the supports defined earlier in the main model, but also the elements that are connected to the designed member. These connected elements may be:

- truss, beam of rib elements;
- surface elements;

- rigid elements, node-to-node interface elements.

Based on the properties of these elements, lateral support stiffness values are estimated by the program. This is detailed in Table **Hiba! A hivatkozási forrás nem található.**-**Hiba! A hivatkozási forrás nem található.**.

In the *Design Parameters* window (Fig. 1) the lateral supports may be edited after pressing the [...] button which is below the Auto Mcr setting and next to the *Lateral Supports* caption. The *Lateral supports* window will appear (Fig. 2), in which the assumed lateral supports are visible. These supports are dependent on the settings of the AutoMcr method:

Automatic default setting; see Table 2**Hiba! A hivatkozási forrás nem található.**-3**Hiba! A hivatkozási forrás nem található.**.

Estimated from k_z, k_w Based on the user-defined k_z and k_w parameters, similarly to AxisVM 12, lateral support location and stiffness values are estimated. For details see Table 1.

Fork supports at both ends In the end of the designed member, lateral supports are assumed with rigid R_y and R_{xx} components. If the user-defined cantilever option is checked, then supports appear only on one end with rigid R_y, R_{xx} and R_{zz} components.

User defined Only the user-defined supports are considered defined in the *Lateral supports* window.

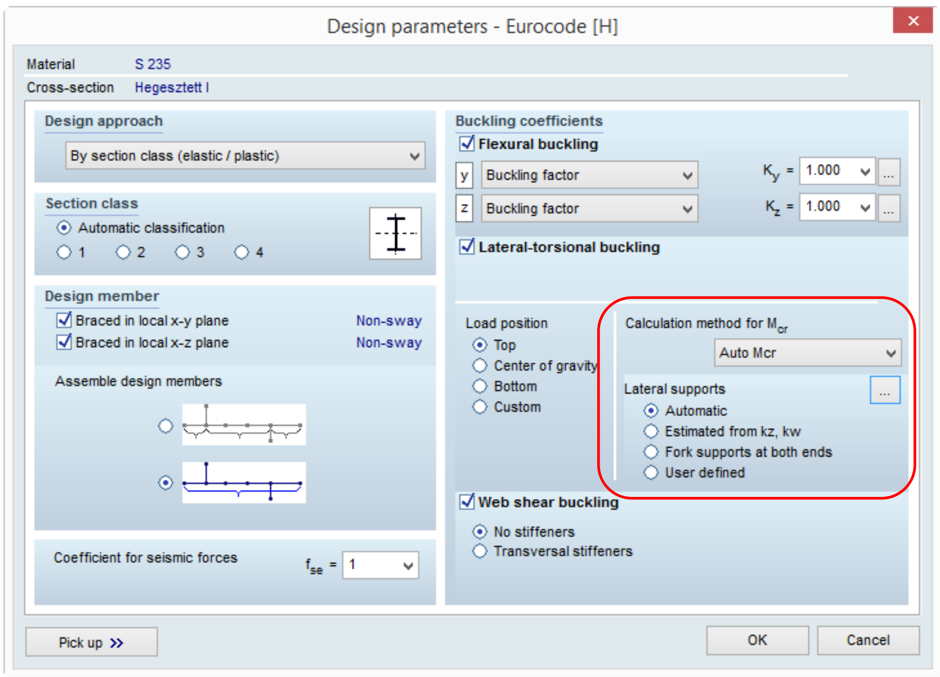


Figure 1: Design Parameters window

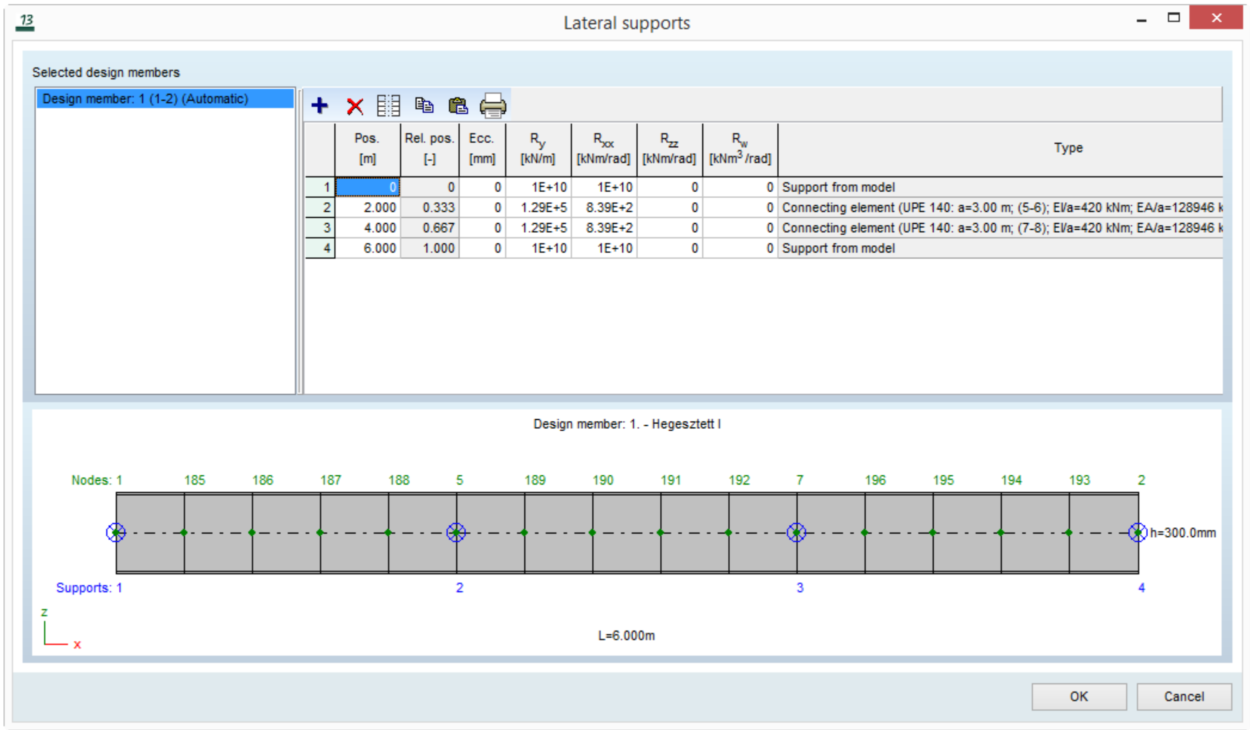
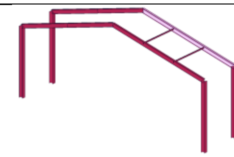
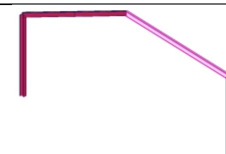
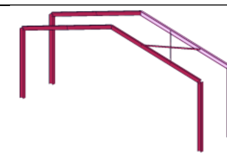
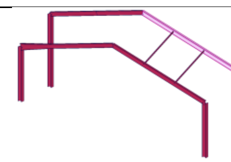


Figure 2: Lateral Supports window

Table 1: Lateral supports determined based on kz and kw

			Support 1					**Support 2**				
	$\boldsymbol{k_z}$	$\boldsymbol{k_w}$	**Rel. pos.**	$\boldsymbol{R_y}$	$\boldsymbol{R_{xx}}$	$\boldsymbol{R_{zz}}$	$\boldsymbol{R_w}$	**Rel. pos.**	$\boldsymbol{R_y}$	$\boldsymbol{R_{xx}}$	$\boldsymbol{R_{zz}}$	$\boldsymbol{R_w}$
	[-]	[-]	[-]	[kN/m]	[kNm]	[kNm]	[kNm^3]	[-]	[kN/m]	[kNm]	[kNm]	[kNm^3]
not cantilever	2<		0	10^{10}		10^{7}		-				
		2<	0		10^{10}		10^{7}	-				
	2		0	10^{10}		10^{10}		-				
		2	0		10^{10}		10^{10}	-				
	$1<k_z<2$		0	10^{10}		10^{10}		1	$10^{5*(2-kz)}$		$10^{5*(2-kz)}$	
		$1<k_w<2$	0		10^{10}		10^{10}	1		$10^{5*(2-kz)}$		$10^{5*(2-kz)}$
	1	1	0	10^{10}	10^{10}	0	0	1	10^{10}	10^{10}	0	0
	0.75		0	10^{10}		10^{7}		1	10^{10}		10^{7}	
		0.75	0		10^{10}		10^{7}	1		10^{10}		10^{7}
	0.5		0	10^{10}		10^{10}		1	10^{10}		10^{30}	
		0.5	0		10^{10}		10^{30}	1		10^{10}		10^{30}
	<0.5		0; 1	10^{10}		0		$1/k_z$; $2/k_z$;...	10^{10}		0	
		<0.5	0; 1		10^{10}		0	$1/k_w$; $2/k_w$;...		10^{10}		0
Cantilever			0 or 1	10^{10}	10^{10}	10^{10}	0					

Table 2: Lateral supports determined by the program automatically – supports and connected line elements

Support or supporting member	**α**	**β**	$\mathbf{R_y}$	$\mathbf{R_{xx}}$	$\mathbf{R_{zz}}$	$\mathbf{R_w}$	**Example**	**Notes**
	[°]	[°]	[kN/m]	[kNm]	[kNm]	[kNm^3]		
nodal support defined in main model	-	-	based on support stiffness			0		when determining R_{zz} the end releases of the designed members are considered
connected truss or pin-connected beam or rib	-	-	EA/a *	0	0	0		
connected beam or rib	90 ±15	0 ±15	EA/a *	2·EI/a	0	0		EI: stiffness of connected member, a: length of connected beam (conservative – it is assumed that the other end of the beam is pinned)
	90 ±15	90 ±15	0	2·EI/a	0	0		
	≠ 90 ±15	0 ±15	0	0	0	0		visible in the table so that the User may edit
	90 ±15	≠ 0 ±15	0	0	0	0		

* if the designed member is not braced in x-y plane; otherwise R_y = 0 kN/m

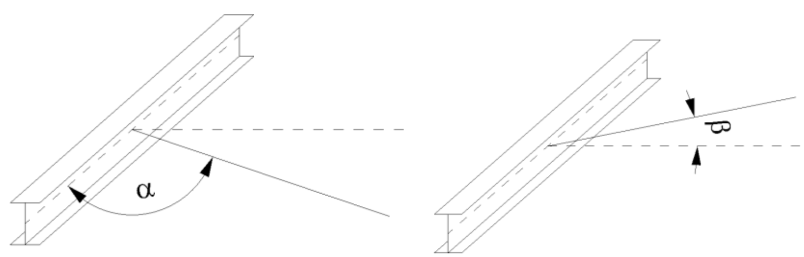
Table 3: Lateral supports determined by the program automatically – further connected elements

Support or supporting member	**α**	**β**	**R_y**	**R_{xx}**	**R_{zz}**	**R_w**	**Example**	**Notes**
	[°]	[°]	[kN/m]	[kNm]	[kNm]	[kNm³]		
surface element or domain (independent of its stiffness and supports)	90 ±15	0 ±15	10^{10} *	10^{10}	10^{10}	0	when designing a column, the slab/slab foundation ensures a fix support	
	0 ±15	90 ±15	0	0	0	0		
	0	⩽ 45	10^{10} *	10^{10}	10^{10}	0	when designing a beam, the slab ensures a continuous support	
Rigid elements or node-to-node interface element – support in the other end	based on support stiffness						when designing a beam, an eccentric support	support eccentricity: length of the rigid element; node-to-node interface element: only those are considered, whose stiffness values (according to the local coordinate system of the designed member): K_y and K_{xx} $\geqslant 10^{10}$
Rigid elements or node-to-node interface element – line element in the other end	same as beam/rod elements						when designing a beam, a connected beam ensures an eccentric support	
Rigid elements or node-to-node interface element – surface element or domain in the other end	same as surface element or domain						when designing a beam, a slab connected by a rigid element	

* if the designed member is not braced in x-y plane; otherwise R_y = 0 kN/m

<u>Notation</u>

α smallest angle between the axis of designed member + the axis of connected member / surface plane (0÷90°)
β smallest angle between the major axis of designed member + the axis of connected member / surface plane (0÷90°)
For example, when designing an I beam these angles for the bracing elements:



PART 2. EXAMPLES

I. GIRDER

In the girders below, lateral torsional buckling is prevented by using fork supports in the ends and by laterally connected beams in two intermediate points of the girder.

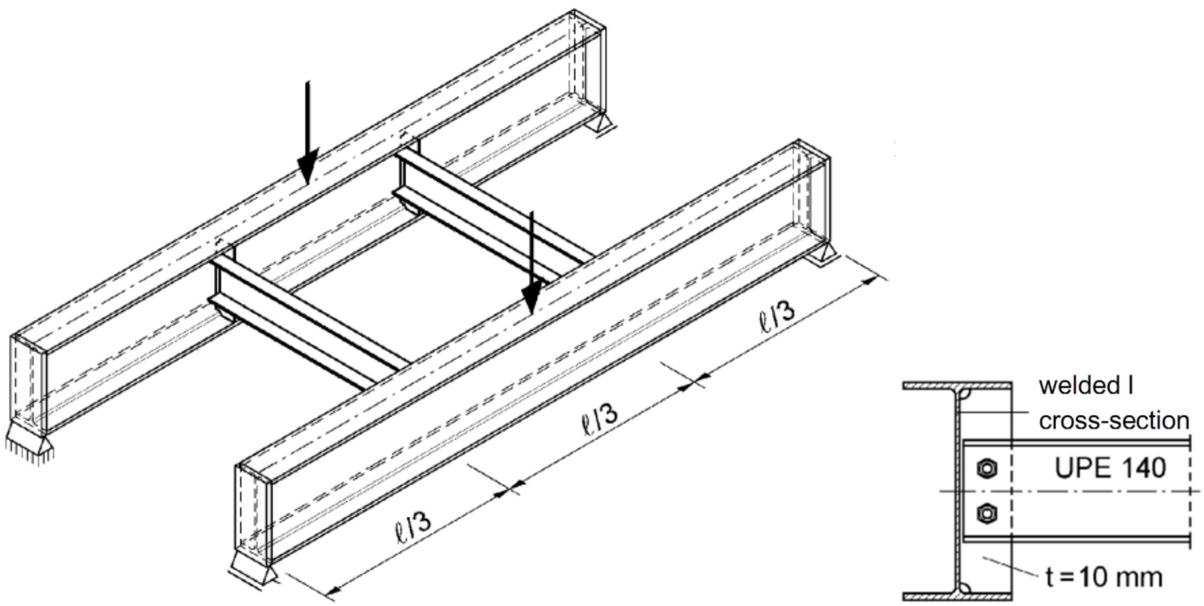


Figure 3: Girders with stiffening beams and connection detail (source: [2])

The goal of this example is to demonstrate:

- how to determine the support stiffness provided by the connected beams;
- comparing M_{cr} obtained by AutoMcr with those of shell models and the LTBeam program.

The structure in the following book served as a basis for this example, which gives guidance in determining the support stiffness provided by adjacent beams: *Teil 2 - Stabilität und Theorie II. Ordnung* [2].

<u>Parameters:</u>

- Cross-section [mm]:
  - girder: in order to be able to compare results with shell finite element models, welded I section similar to IPE 300: web: 300*7, flanges: 150*11 mm;
  - connected beam UPE 140;
- Span:
  - girder: l=6m;
  - connected beam: a=3m;

- Loading: distributed force along the whole girder or point load in the middle of the girder; applied in the geometric centre or on top of the flange;
- Support condition: supports in the ends of the girder according to Figure 3 (either of the two girders may move laterally)

<u>Name of AxisVM models:</u>

- Beam finite element model with AutoMcr: I. Girder - beam finite element model.axs
- Shell finite element model as an eigenvalue problem: I. Girder - shell finite element model.axs

Lateral support stiffness

In the ends of the beams, there are fork supports. In AxisVM13, when creating the AutoMcr submodel, the program automatically adopts the supports defined earlier in *Elements >> Nodal supports*. These supports of the AutoMcr submodel can be seen in the table at *Design Parameters >> Lateral supports*. For the girder, these adopted supports can be seen in Figure 4, of which the lateral R_y and rotational R_{xx} stiffness components are stiff.

	Pos. [m]	Rel. pos. [-]	Ecc. [mm]	R_y [kN/m]	R_{xx} [kNm/rad]	R_{zz} [kNm]	R_w [kNm^3/rad]	Type
1	0	0	0	1E+10	1E+10	0	0	Support from model
2	2.000	0.333	0	1E+10	1E+10	0	0	Connecting element
3	4.000	0.667	0	1E+10	1E+10	0	0	Connecting element
4	6.000	1.000	0	1E+10	1E+10	0	0	Support from model

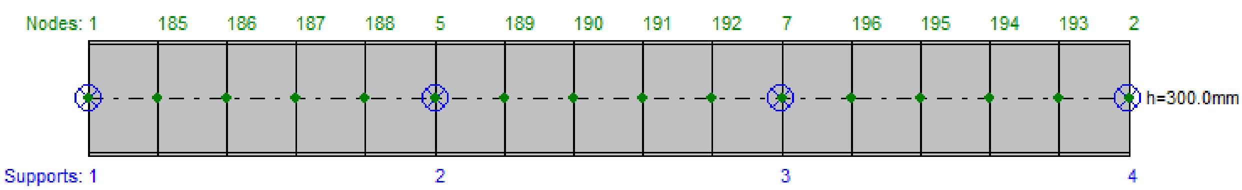


Figure 4: Defining lateral supports in AxisVM13

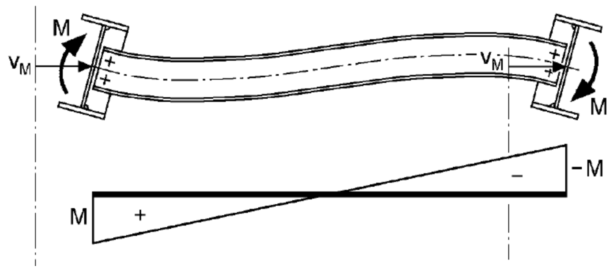
In the table above, additionally to the adopted supports (*Support form model*), the connected beams also provide support (*Connecting element*) against lateral torsional buckling. The program automatically gives approximate values for the R_y and R_{xx} components of such a support:

- $R_y = 10^{10}$ kN/m if the analysed member is braced in local x-y plane; otherwise: $R_y = 0$ kN/m;
- $R_{xx}=2*EI/a$ based on the length (a) and the inertia (I) of the connecting member.

It is the User's responsibility to define this stiffness value accurately, if needed. To calculate the stiffness provided by the connected beams, [2] gives the following recommendation: the rotational support stiffness (R_{xx}) may easily be calculated based on the stiffness of the connected beam (EI/a). The stiffness values may be determined by the following two formulas, based on the deformation of the structure:

Non-symmetric case

Girders exhibit lateral displacements and rotate in the same direction. The connected beams do not provide any lateral support.



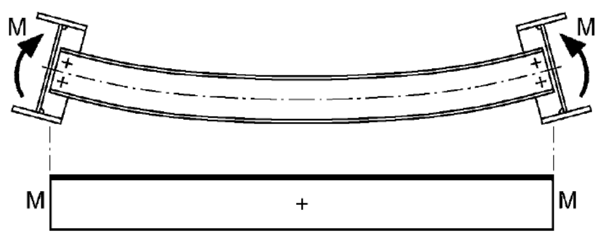
R_{xx} = **6*EI/a** =
= 6*21000kN/cm^2 * 599.6cm^4 / 3m =
= 2520 kNm/rad

$R_y = R_{zz} = R_w = 0$

Figure 5: Possible deformation of the girder structure: non-symmetric case (source: [2])

Symmetric case

Girders do not exhibit lateral displacements and rotate in the opposite direction. The connected beams provide some lateral support.



R_{xx} = **2*EI/a** =
= 6*21000kN/cm^2 * 599.6cm^4 / 3m =
= 840 kNm/rad

$R_y > 0$

$R_{zz} = R_w = 0$

Figure 6: Possible deformation of the girder structure: symmetric case (source: [2])

In reality, semi-rigid connections and the distortions of the girder may lower the above support stiffness values, therefore to stay on the safe side, the program uses the second case. In the following comparison, both cases will be presented, in the second case by neglecting R_y.

Comparison of results

The obtained M_{cr} results are compared to results of shell models created in AxisVM13, and of the LTBeam program, which works on the same basis as AutoMcr. The models created in LTBeam (v1.0.10) have the same settings. The differences in the obtained results are due to the used numerical algorithm and to the differences in the discretisation.

The shell models in AxisVM13 were created with the help of the *Edit >> Convert beams to shell model* function. After defining the load, by solving an eigenvalue problem (*Buckling* tab), a load factor is obtained. M_{cr} can be calculated by multiplying the load factor with the maximal moment along the beam. Compared to beam models, shell models are capable of a more detailed and precise modelling, thus the obtained M_{cr} is more accurate. Another advantage of shell models is that there is no need to create a sub-model, and thus there is no error caused by defining lateral supports. The disadvantage is that the modelling is more complex and more time consuming. The calculation time for AutoMcr is about a 100 times lower that for an appropriate shell model. To avoid local deformations in the shell model, the web of the girder is stiffened by rigid elements at the intersection of the beams (a more accurate modelling of the stiffening plate is neglected). The obtained lowest eigenform is the symmetric case, while the second is the non-symmetric case (Figure 7).

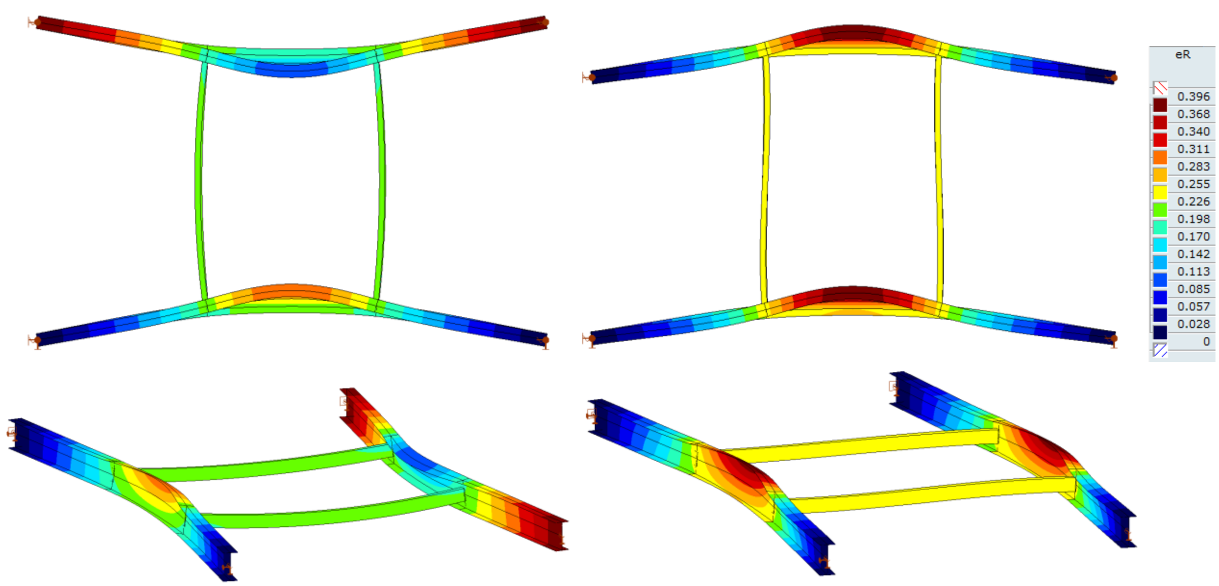


Figure 7: Eigenforms of shell finite element models; left: symmetric case; right: non-symmetric case [mm]

Results [kNm]

In Table 4, the Δ columns show the difference of the AutoMcr results ($M_{AutoMcr}$) compared to either of the other methods (M_{cr}), based on this formula: $\Delta = (M_{AutoMcr} - M_{cr}) / M_{cr}$.

Table 4: Comparison of results

Load type	Load position	Deformation	Auto Mcr	LTBeam	Δ	Shell model	Δ
Distributed	Top flange	Non-symmetric	597	596	0%	644	-8%
		Symmetric	554	554	0%	581	-5%
	Geometric centre	Non-symmetric	625	624	0%	619	1%
		Symmetric	578	577	0%	558	3%
Point load	Top flange	Non-symmetric	628	629	0%	624	1%
		Symmetric	569	569	0%	566	1%
	Geometric centre	Non-symmetric	702	702	0%	669	5%
		Symmetric	639	639	0%	610	5%

Comparing the results to the LTBeam program, the AutoMcr method is accurate. Furthermore, it can be concluded, that the results obtained by the shell finite element model and the beam finite element model with AutoMcr correspond well, thus the applied support stiffness values are accurate enough.

PART 3. VERIFICATION

In this part, the verification of the AutoMcr method is summarized. The calculated M_{cr} values are compared to those of other methods and programs, among which is the LTBeam program that is based on the same theoretical background as AutoMcr. In the first section, the LTBeam and shell models are taken from the verification documentation of LTBeam: *Yvan Galea: LTBeam – Report on Validation Tests* [3]. Afterwards, comparison is made with the ENV [4] analytic formula. Lastly, the differences of the AutoMcr method in AxisVM12 and 13 are summarized.

The error (Δ) of the AutoMcr results ($M_{AutoMcr}$) compared to either of the other methods (M_{cr}) was calculated based on this formula: $\Delta = (M_{AutoMcr} - M_{cr}) / M_{cr}$.

I. Validating with LTBeam program and shell models

<u>Ansys shell models</u>

Based on Chapter 2 of [3].

This section presents simple examples of all the types of models that can be calculated with AutoMcr. Results are compared to those of the LTBeam program and of shell models in Ansys [3] and are presented in Table 5Table 6. M_{cr} values are only -4÷3% different, which is a very good result.

Name of Axis model: LTBeam Validation - Chapter 2 - #.axs (where # is the number of the example)

Table 5: Comparison of results I.

Type of example	Nb. of example #	1x sym. c.s.	Ecc. load	Auto Mcr [kNm]	LTBeam M_{cr} [kNm]	LTBeam Δ [%]	Ansys M_{cr} [kNm]	Ansys Δ [%]	Note
VARIABLE cross-section	**40**			188	186	-0.7	188	0.3	
	41			156	155	-0.7	157	0.4	
MULTI-SPAN BEAM: intermediate lateral support	**50**			275	274	-0.4	274	-0.4	*Assemble design members* parameter: results are more accurate if the beam is modelled as a whole
	51			293	288	-1.6	288	-1.6	
	52			343	338	-1.5	338	-1.5	
	53			254	255	0.3	255	0.3	
	54			212	210	-0.8	210	-0.8	
	55	x		160	160	0.3	160	0.3	
	56	x		130	129	-0.9	129	-0.9	
	57	x		184	184	0.3	184	0.3	
	58	x		157	156	-0.6	156	-0.8	
CANTILEVER: load in varying positions: top flange, shear centre, lower flange	**60**		x	180	184	1.7	185	2.4	
				233	233	0.2	234	0.6	
			x	268	267	-0.3	268	-0.1	
	61		x	292	300	2.6	300	2.6	
				421	424	0.7	422	0.2	
			x	538	536	-0.3	532	-1.1	
	62		x	282	290	2.7	291	3.2	
				424	425	0.4	425	0.3	
			x	529	527	-0.4	525	-0.7	
	65	x	x	119	121	1.4	121	1.8	
		x		132	133	0.2	133	0.5	
		x	x	155	157	1.0	157	0.9	
	66	x	x	190	193	1.8	193	1.6	
		x		223	224	0.7	223	0.3	
		x	x	298	305	2.3	303	1.5	
	67	x	x	184	188	2.3	189	2.5	
		x		220	221	0.4	221	0.3	
		x	x	285	290	1.6	288	1.0	

Table 6: Comparison of results II.

Type of example	Nb. of example #	1x sym. c.s.	Ecc. load	Auto Mcr	LTBeam		Ansys		Note
					M_{cr}	Δ	M_{cr}	Δ	
				[kNm]	[kNm]	[%]	[kNm]	[%]	
SIMPLE BEAM: varying support conditions	**70**			150	149	-0.5	149	-0.7	
	71			530	523	-1.4	523	-1.2	
	72			361	358	-0.9	358	-0.9	
	75	x		105	105	-0.4	105	-0.4	
	76	x		264	264	0.1	263	-0.4	
	77	x		223	222	-0.3	221	-0.7	
SIMPLY SUPPORTED BEAM: intermediate lateral supports	**80**			854	853	-0.1	847	-0.9	Intermediate lateral supports may be defined directly in the AxisVM model - adopted by AutoMcr autoomatically-, or in the Lateral Supports window
	82			625	625	0.1	622	-0.4	
	83			1265	1230	-2.9	1220	-3.7	
	84		x	625	622	-0.4	622	-0.4	
				579	577	-0.3	577	-0.3	
			x	359	363	1.2	363	1.2	
	85			477	478	0.1	476	-0.2	
	86			299	300	0.2	299	-0.1	
	87			344	345	0.1	344	-0.1	
	88		x	432	432	-0.1	431	-0.4	
				403	403	0.1	402	-0.1	
			x	377	378	0.4	378	0.1	
	89			330	324	-1.7	323	-1.9	Continuous support: may only be defined in Auto Mcr as a number of individual supports
	90			319	314	-1.5	313	-1.8	
	91		x	315	309	-1.7	310	-1.6	
	92	x		225	224	-0.4	223	-0.5	
T CROSS-SECTION: simply supported	**100**	x		17.7	17.8	0.6	17.8	0.7	
	101	x		15.1	15.1	-0.3	15.1	-0.2	
	102	x		15.6	15.7	0.7	15.8	1.0	
	103	x		13	13.0	-0.1	13.0	0.3	

Variable cross-section

Based on Chapter 5 of [3].

The analysed beam has variable web height ($h_{w1} \div h_{w2}$), fork supports in the end points, and end moments (M_1 and M_2). The results are generally +2% and maximum -9% different from the results of LTBeam and Finelg [3], the reason of which lies in the different discretisation of the sections. These differences are negligible compared to the general uncertainty of modelling variable cross-sections.

Name of Axis model: LTBeam Validation - Chapter 5 - Variable cross-section.axs

Table 7: Comparison of results of beam with variable cross section

Model Nb.	Span	hw1	hw2	M1	M2	AutoMcr	LTBeam		Finelg	
						M_{cr}	M_{cr}	Δ	M_{cr}	Δ
	[m]	[mm]	[mm]	[kNm]	[kNm]	[kNm]	[kNm]	%	[kNm]	%
P1-1A	5	400	800	200	-800	3498	3591	2.6	3586	2.4
P1-2A					-600	3718	3817	2.6	3811	2.4
P1-3A					400	2012	2064	2.5	2062	2.4
P1-4A					600	2253	2311	2.5	2308	2.4
P1-5A					800	2391	2453	2.5	2450	2.4
P1-6A					200	1501	1541	2.6	1539	2.5
P3-1A	5	200	1000	200	-1200	3599	3365	-6.9	3361	-7.1
P3-4A					1000	2674	2483	-7.7	2480	-7.8
P3-6A					200	1579	1455	-8.5	1454	-8.6
P1-1A	10	400	800	200	-800	1173	1189	1.3	1189	1.3
P1-6A					200	510	520	2.0	521	2.0
P3-1A	10	2000	1000	200	-1200	1169	1137	-2.8	1138	-2.8

II. Basic Cases with the analytic expression in ENV

In order to determine M_{cr} values, AxisVM program has long been using the so-called "3 factor formula", which can be found in the pre-standard of the Eurocode [4] (in the following referred to as ENV). Additionally to the 3 C factors, the formula uses the k_z and k_w effective length factors. Recommended values for all these factors may be found in several literatures for basic cases only, and in some cases giving different results. To calculate the C_1 factor, Lopez et al. proposed a simple analytic formula that AxisVM program implemented. This formula was calibrated by numerical results in several support conditions and load cases.

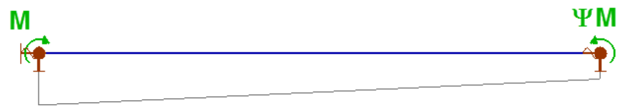
In Table 6, results are summarized and compared for the AutoMcr method and for the ENV formula based on factors of several sources. All the examples are beams supported on the ends, loaded and supported in their shear centre, with a double- or single-symmetric I cross-section and various effective length factors.

In line with [5], in the ENV formula, k_z and k_w are assumed to be equal. Additionally to pinned and fixed beams, [5] provides factors for a third "semi-fixed" support condition: when k values are taken as 0.7. This provides less information about the support condition, than what needs to be defined in AutoMcr. Therefore, in the following, this case was modelled with three different settings. Logically, the k=0.7 corresponds to a beam, that is fully-fixed on one end and pinned on the other; for this setting, the smallest possible M_{cr} value is included in the table. In the other two settings either k_z or k_w is 0.5, the other is 1, which are generally used in practice. Table 5 summarizes these support conditions (the support components not included in the table are assumed to be zero for the AutoMcr method).

Table 8: Lateral support conditions as defined in for the different methods

Support condition	ENV		AutoMcr	
	k_z	k_w	Left support	Right support
Pinned	1	1	$R_y = R_{xx} = 10^{10}$	$R_y = R_{xx} = 10^{10}$
„semi-fixed"	0.7	0.7	$R_y = R_{xx} = R_{zz} = R_w = 10^{10}$	$R_y = R_{xx} = 10^{10}$
	0.5	1	$R_y = R_{xx} = R_{zz} = 10^{10}$	$R_y = R_{xx} = R_{zz} = 10^{10}$
	1	0.5	$R_y = R_{xx} = R_w = 10^{10}$	$R_y = R_{xx} = R_w = 10^{10}$
Fixed	0.5	0.5	$R_y = R_{xx} = R_{zz} = R_w = 10^{10}$	$R_y = R_{xx} = R_{zz} = R_w = 10^{10}$

End moments only



Span: L=8m

Figure 8. End moments only

Cross-section: Symmetric: welded (same plate size as IPE 300)

Name of Axis model: Basic cases – End moments – Symmetric cross-section.axs

Table 9: Comparison of analytic and numerical results, end moments only

Ratio of end moments	Effective length factors		Auto Mcr	ENV analitic formula [4] C factors [5]					ENV formula C_1 factor: Lopez [6]			Access Steel [7]			LTBeam v1.0.10		Abaqus [8]	
Ψ	k_z	k_w	M_{cr}	C_1	C_2	C_3	M_{cr}	Δ	C_1	M_{cr}	Δ	C_1	M_{cr}	Δ	M_{cr}	Δ	M_{cr}	Δ
[-]	[-]	[-]	[kNm]	[-]	[-]	[-]	[kNm]	[%]	[-]	[kNm]	[%]	[-]	[kNm]	[%]	[kNm]	[%]	[kNm]	[%]
	1	1	57	1			57	0	1	57	0	1	57	0	57	0	57	0
	0.7	0.7	91	1			91	0	1	91	0				91	0		
1	0.5	1	126	1	-	-	114	**11**	1	114	**11**				126	0		
	1	0.5	84	1			75	**12**	1	75	**12**				84	0		
	0.5	0.5	150	1			150	0	1	150	0				150	0		
	1	1	75	1.323			75	0	1.301	74	1	1.31	75	0	75	0		
	0.7	0.7	110	1.473			134	**-18**	1.302	119	-8				110	0		
0.5	0.5	1	165	1.473	-	-	168	-2	1.301	148	**11**				165	0		
	1	0.5	111	1.473			111	0	1.301	98	**13**				111	0		
	0.5	0.5	198	1.514			227	**-13**	1.305	196	1				198	0		
	1	1	104	1.879			107	-3	1.78	102	2	1.77	101	3	104	0		
	0.7	0.7	134	2.092			191	**-30**	1.785	163	**-18**				134	0		
0	0.5	1	226	2.092	-	-	239	-5	1.782	203	**11**				226	0		
	1	0.5	157	2.092			157	0	1.782	134	**17**				157	0		
	0.5	0.5	275	2.15			323	**-15**	1.803	271	1				275	0		
	1	1	143	2.704			154	-7	2.397	137	4	2.33	133	8	143	0		
	0.7	0.7	163	3.009			275	**-41**	2.499	228	**-29**				163	0		
-0.5	0.5	1	288	3.009	-	-	343	**-16**	2.472	282	2				289	0		
	1	0.5	227	3.009			226	0	2.472	186	**22**				227	0		
	0.5	0.5	375	3.093			465	**-19**	2.679	402	-7				375	0		
	1	1	154	2.752			157	-2	2.449	140	10	2.55	140	10	154	0	153	1
	0.7	0.7	190	3.063			279	**-32**	2.652	242	**-21**				190	0		
-1	0.5	1	271	3.063	-	-	349	**-22**	2.599	296	-8				271	0		
	1	0.5	268	3.063			230	**17**	2.599	195	**37**				268	0		
	0.5	0.5	378	3.149			473	**-20**	3.024	454	**-17**				378	0		

It can be seen in Table 9, that the various methods give significantly different results. In all cases, the results of AutoMcr and LTBeam are very close.

- For pinned beams, results are always very similar for all methods.
- For fixed beams, the results from the ENV method combined with C_1 factor based on the Lopez formula [6] is closest to the AutoMcr results, mainly if $\Psi > 0$.
- The differences between the methods for the "semi-fixed" cases lie in the different definition of the support condition.

<u>Transverse loading</u>

Name of Axis model: Basic cases – Transverse loading – Symmetric cross-section.axs

Table 10: Comparison of analytic and numerical results, transverse loading

Moment distribution	**Effective length factors**	**Auto Mcr**	**ENV analitic formula [4] C formula [5]**					**ENV formula C_1 factor: Lopez [6]**		
	k [-]	**M_{cr}** [kNm]	**C_1** [-]	**C_2** [-]	**C_3** [-]	**M_{cr}** [kNm]	**Δ** [%]	**C_1** [-]	**M_{cr}** [kNm]	**Δ** [%]
distributed 	1	135	1.132	0.459	0.525	134	-1	1.129	134	-1
	0.5	292	1	0.304	0.478	290	-1	1.014	302	3
distributed 	1	314	2.576	1.562	0.753	305	-3	2.408	285	**-10**
	0.5	524	1.494	0.652	1.07	446	**-17**	1.908	569	8
concentrated 	1	161	1.365	0.553	0.411	162	1	1.247	148	-9
	0.5	318	1.07	0.432	0.338	319	0	1.03	307	-4
concentrated 	1	203	1.565	1.267	2.64	185	-10	1.382	164	**-24**
	0.5	313	0.938	0.715	4.8	280	**-12**	1.037	309	-1
concentrated 	1	130	1.046	0.43	0.562	124	-5	1.124	133	2
	0.5	277	1.01	0.41	0.539	301	8	1.013	302	8

III. DIFFERENCES BETWEEN AXISVM VERSION 12 AND 13

In AxisVM12, when defining the sub-model, the support conditions are assumed based on the user defined k_z and k_w values. The obtained M_{cr} values are very similar to the results in AxisVM13 in the basic cases (k=0.5 or k=1) but are less accurate if $k_z \neq k_w$.

A further important difference is, that in version 13, for a safe design, in case of a simple beam with fixed end-supports, AutoMcr automatically assumes that $R_y=R_{xx}=R_{zz}=10^{10}$, while the user shall determine R_w. In version 12, if $k_z=k_w=0.5$, R_w is also assumed to be rigid.

Table 11: Lateral support conditions

Type of support	**Effective length factor**		**Lateral support stiffness values**	
	k_z	**k_w**	**AxisVM12**	**AxisVM13 basic setting**
pinned	1	1	$R_y = R_{xx} = 10^{10}$	$R_y = R_{xx} = 10^{10}$
fixed	0.5	0.5	$R_y = R_{xx} = R_{zz} = R_w = 10^{10}$	$R_y = R_{xx} = R_{zz} = 10^{10}$

The AutoMcr method of AxisVM13 is numerically more precise in version 12. The M_{cr} results are maximum ±10% different. When first opening a model in version 13, that was created and saved in version 12, the support conditions are the same as they were in version 12, but the M_{cr} values are calculated by the more precise algorithm. In the Steel Design Parameters window, such a model will appear to have the M_{cr} method: „AutoMcr_v12". Conversion of such models are recommended, and the redefinition of lateral support conditions, to facilitate a more accurate design.

References

[1] Yvan Galea: Moment critique de deversement elastique de poutres flechies presentation du logiciel ltbeam, CTICM, www.cticm.com, 2003 (in French)

[2] Stahlbau: Teil 2 - Stabilität und Theorie II. Ordnung, 10.4 Stabilisierung durch behinderung der verdrehungen, szerző: Rolf Kindmann, Ernst&Sohn, 4th edition, pp. 336-338., 2008 (in German)

[3] Yvan Galea: LTBeam – Report on Validation Tests, CTICM, July 2002, www.cticm.com

[4] ENV 1993-1-1: Appendix F

[5] Ádány, Dulácska, Dunai, Fernezely, Horváth: Acélszerkezetek, 1. Általános eljárások, Tervezés az Eurocode alapján, 2006 (in Hungarian)

[6] López, Yong, Serna: Lateral-torsional buckling of steel beams: A general expression for the moment gradient factor. Proceedings of the International Colloquium of Stability and Ductility of Steel Structures, D. Camotim et al. Eds., Lisbon, Portugal, September 6-8, 2006.

[7] Access Steel: NCCI: Elastic critical moment for lateral torsional buckling, SN003a-EN-EU, 2008

[8] Braham M. – "Le déversement élastique des poutres en I à section monosysmétrique soumises à un gradient de moment de flexion" – Revue Construction Métallique n°1-2001 – CTICM (in French)