


# Meclib: Efficient authoring of STACK questions with interactive input and formative feedback

Prof. Dr.-Ing. Martin Kraska

2023 STACK Community Meeting

Give the general formula for [Tidy STACK question tool](#) |  Question is missing tests or variants.  
the length of the hypotenuse  $c$  of a right-angled triangle with sides  $a$  and  $b$ .

✗ Missing variable:  $b$  ( $b$ ).

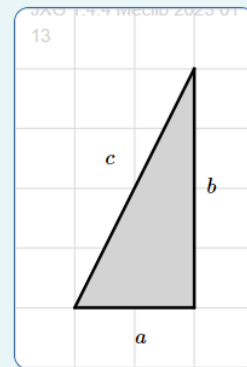
What is the exact length of the hypotenuse if the grid width is  $L$ ?

✗ Missing variable:  $L$  ( $L$ ).

What is the length of the hypotenuse if the grid width is 1 cm?

✗ The absolute value is at least 50% too small.

Check



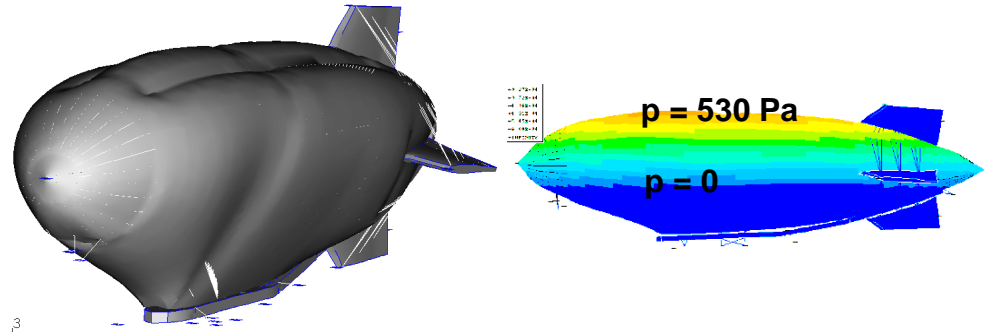


# Outline

- Introduction
- Update on Meclib
  - Concept
  - Examples
- Feedback functions
- Unit test of feedback functions
- Summary



# Introduction



## Education:

- Mechanical Engineering in Moscow, Freiberg (Sa.) and Berlin
- Research and teaching assistant at the Institute of Mechanics at the TU Berlin

## Professional experience:

- Structural analysis at CargoLifter
- Metal forming simulation at INPRO Berlin
- Non-destructive testing and process monitoring



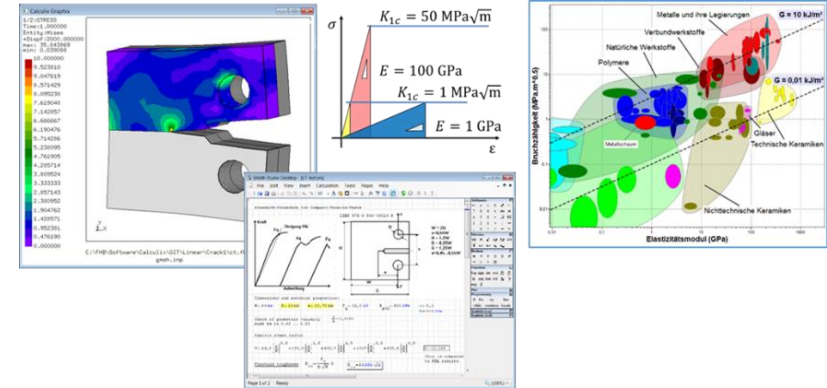
# Introduction

## Teaching at THB:

- Engineering Mechanics
- Materials science (postgrad)
- Finite element analysis (3rd year and postgrad)
- Product development (3rd year)

## Research at THB:

- Mechanics of materials and structures
- Application and extension of free math software and e-learning tools (SMath Studio, CalculiX, STACK)

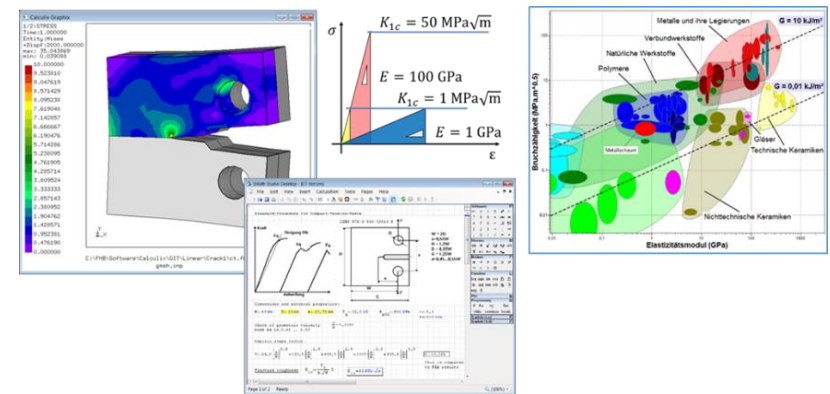




# Introduction

## SMath Studio:

- [Free math software](#) similar to MathCAD
- Contributions
  - [German handbook](#)
  - [Advanced usage examples](#)
  - [Multilingual Interactive Getting Started](#)
  - [Plugin for access to Maxima for extended CAS features](#)
  - Active user support in the [forum](#)

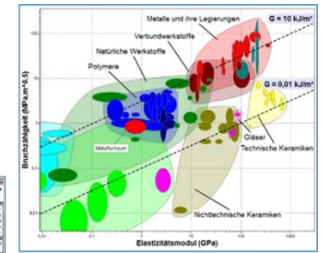
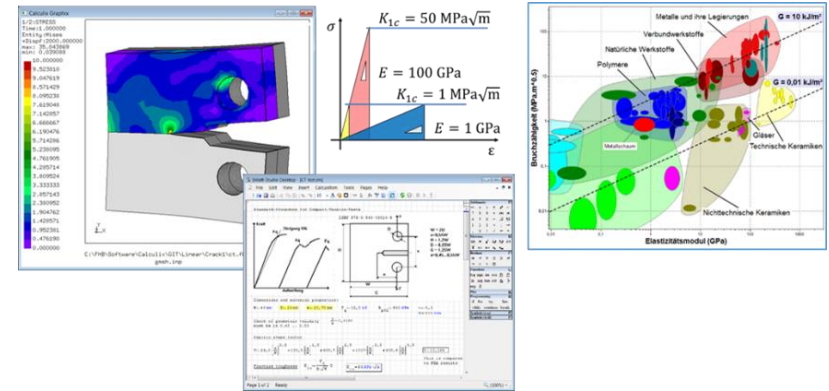




# Introduction

## CalculiX:

- [Free nonlinear FEA software](#)
- Contributions
  - [Public example collection on Github](#)
  - Sponsoring of development of pre/post features (relevant for teaching purposes)





# STACK in Teaching Mechanics

## Engineering Mechanics

- Heterogeneous students – need for asynchronous e-learning materials with instant feedback 24/7
- Most questions involve sketches
- High importance of model building skills (sketching)
- High importance of unit handling

## STACK has all we need

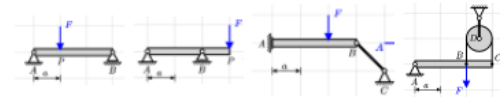
- Power of CAS
- Units
- Graphics via JSXGraph
- Powerful feedback concept

### UE 04 Freischnitt und statische Bestimmtheit (10 min)

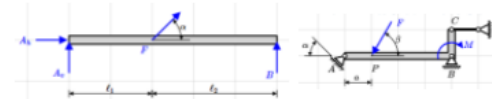
Zu gegebenen Systemen suchen Sie passende Freischnitte heraus und beurteilen, ob die Systeme statisch bestimmt, statisch unbestimmt oder beweglich sind.

### UE 04 Auflagerreaktionen einfach (40 min)

Hier können Sie studieren, wie Systeme freigeschnitten werden. Trainiert wird das Aufstellen der Gleichgewichtsbedingungen und das Auflösen nach den unbekannten Auflagerreaktionen. Die Systeme sind vergleichsweise einfach gehalten.

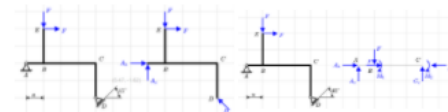


### UE 04 Auflagerreaktionen weniger einfach (40 min)

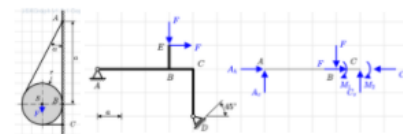


### UE 04 Auflagerreaktionen Rahmen mit Reduktion (60 min)

Hier werden zunächst Auflagerreaktionen berechnet und dann das System auf einen horizontalen Balken reduziert. Dabei sind Versetzungsmomente zu berechnen.



### HA 04



Moodle Quizzes of a typical week



# STACK in Teaching Mechanics

## Challenges

- Efficient implementation of standard formative feedback
  - Missing or unexpected variables
  - Wrong coefficients at certain variables
  - Numeric values off by a power of 10
  - Numeric values off by x%

→ Dedicated feedback functions/answer tests instead of PRTs
- Efficient production of interactive graphics
  - Consistent appearance
  - Reduced complexity of authoring

→ Library of JSXGraph based objects controlled by Maxima variables (**Meclib**)





# What is Meclib?

Collection of JavaScript classes representing **tailored objects for sketches/drawings in engineering mechanics**

- Linked from a website (Github) to the question text
- Authors of questions don't need any knowledge about JavaScript or JSXGraph
- Suited for bulk production of visually consistent sketches in your tests (strong Corporate Identity)
- Sketches entirely defined inside STACK, no external files or editors, easy to re-use and distribute.

Collection of **complex feedback functions in Maxima**

- Dedicated feedback for interactive graphics objects
- General purpose functions for formative feedback on numerical or symbolical answers



# STACK and JSXGraph

## Question variables

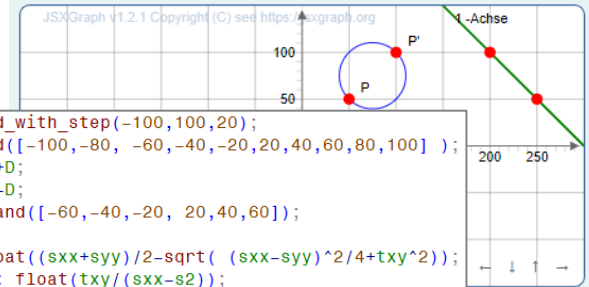
(Randomized) question variables

TM2 03 T02

**Mohrscher Spannungskreis:** Gegeben sind die folgenden Spannungswerte:

$$\sigma_x = 20 \text{ MPa}, \sigma_y = 140 \text{ MPa}, \tau_{xy} = -40 \text{ MPa}$$

Ziehen Sie die Punkte  $P$  und  $P'$  sowie die 1-Achse an die richtige Stelle.



```
M: rand_with_step(-100,100,20);
D: rand([-100,-80, -60,-40,-20,20,40,60,80,100] );
sxx: M+D;
syy: M-D;
txy: rand([-60,-40,-20, 20,40,60]);

s2: float((sxx+syy)/2-sqrt( (sxx-syy)^2/4+txy^2));
slope1: float(txy/(sxx-s2));

sxx_MPa: stackunits(sxx,MPa);
```

## Question text

[[JSXGraph, block with  
**problem specific**  
Javascript code]]

Ordinary question text  
follows here

```
[[[jsxgraph width='500px' height='250px'
input-ref-ans1='ans1Ref'
input-ref-ans2='ans2Ref'
input-ref-ans4='ans4Ref'
input-ref-ans5='ans5Ref']]

var board = JSXGraph.initBoard(divid, {
  boundingbox: [-300, 150, 300, -150], showNavigation:true,
  grid:{gridX:50,gridY:50},axis:true
});

var xaxis = board.create('line', [ [0, 0], [1, 0] ], { visible: false });
var A = board.create('point', [50, 50], { name: 'P', snaptoGrid: true, snap:
var AS = board.create('point', [100, 100], { name: 'P'', snaptoGrid: true,
var MSK1 = board.create('semicircle', [A, AS]);
var MSK2 = board.create('semicircle', [AS, A]);
var int1 = board.create('intersection', [MSK1, xaxis], { visible: true, siz
var int2 = board.create('intersection', [MSK2, xaxis], { visible: true, siz
var g1 = board.create('point', [200, 100], {
  name: "g1", label:{visible:false},snaptoGrid: true, snapsizeX: 10, snap
var g2 = board.create('point', [250, 50], {
  name: "g2",label:{visible:false}, snaptoGrid: true, snapsizeX: 10, snap
var s1 = board.create('line', [g1, g2], {
  strokecolor: 'green', name:'1 -Achse',withLabel:true,label:{offset:[10,
stack_jxg.bind_point(ans1Ref,A);
stack_jxg.bind_point(ans2Ref,AS);
//stack_jxg.bind_point(ans3Ref,int1);
stack_jxg.bind_point(ans4Ref,g1);
stack_jxg.bind_point(ans5Ref,g2);
board.update();
]]/jsxgraph]]</p>
<p hidden="">[[[input:ans1]] [[validation:ans1]]</p>
<p hidden="">[[[input:ans2]] [[validation:ans2]]</p>
<p hidden="">[[[input:ans4]] [[validation:ans4]]</p>
<p hidden="">[[[input:ans5]] [[validation:ans5]]</p>
<p>[[feedback:prtl]]</p>
```



# STACK and JSXGraph with MecLib

## Question variables

(Randomized) question variables

Maxima list of objects

## Question text

[[JSXGraph, block with  
**entirely generic**  
Javascript code]]

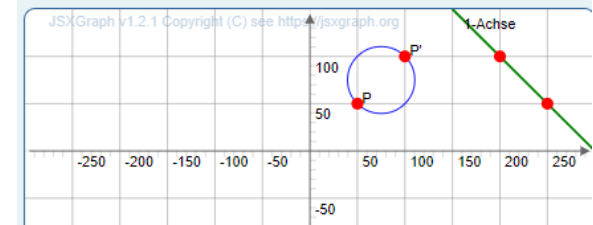
Ordinary question text  
follows here

TM2 03 01-2

**Mohrscher Spannungskreis zur vorigen Aufgabe:** Gegeben sind die folgenden Spannungswerte:

$$\sigma_x = -30 \text{ MPa}, \sigma_y = 90 \text{ MPa}, \tau_{xy} = 10 \text{ MPa}$$

Ziehen Sie die Punkte  $P$  und  $P'$  sowie die 1-Achse an die richtige Stelle.



```
initdata: [  
  [ "grid", " ", " ", -6,6, -3, 3, 40, [f,f] ],  
  [ "circle2p", "P", "P'", [50,50],[100,100], f ],  
  [ "line2p", "1-Achse", [200,100],[250,50], f ]  
];
```

- No Javascript
- Consistent appearance
- Tailored objects



# Meclib on Github

<https://github.com/mkraska/meclib>



## List of Objects

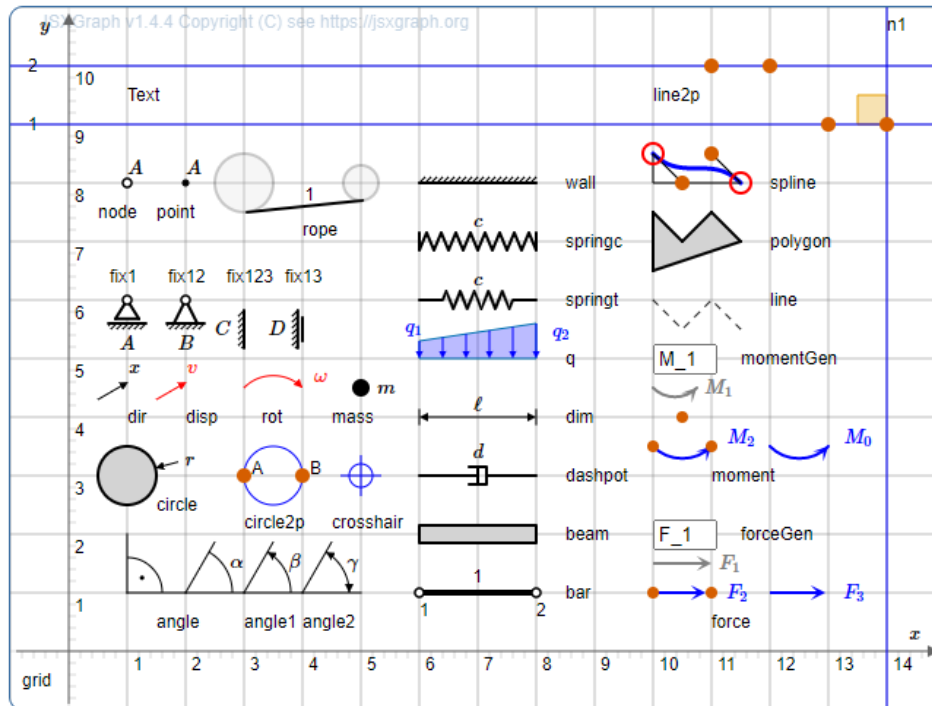
mkraska edited this page now · 13 revisions

Interactivity legend:

- "Switch": object can be activated or deactivated by double-click.
- "Move": object or it's control points can be dragged around with the mouse.
- "Delete": object can be deleted by double-click (if active)
- "Generate": object can generate "force" or "moment" objects by dragging sample

Return value: only relevant for interactive input

Object	Interactivity	Return value in <code>names</code>
"angle", "angle1", "angle2"		label string
"bar"	Switch	"show" or list of load indices if h
"beam"	Switch	state
"circle"	Switch	state
"circle2p"	Move	[x1,y1],[x2,y2]
"crosshair"	Move	0



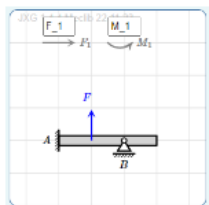


# Demo Moodle Course at THB

<https://extmoodle.th-brandenburg.de/course/view.php?id=138&section=1>

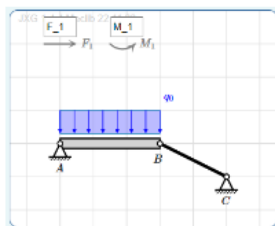
## FBD 01

- Interaktive Freischnittskizze
- Auswahllisten für MC-Fragen
- Musterlösung mit separatem Meclib-Bild



## STACK 22 FBD

- Beispiel für das STACK User Meeting 2022
- Interaktive Freischnittskizze
- Festlager, Pendelstütze und Streckenlast



## Interactive Tryout and Automatic Unit Tests

### fb\_bar\_name() test suite

- Unit tests of the `fb_bar_name()` feedback function

### fb\_unidir() test suite

- Unit tests of the `fb_unidir()` feedback function

### fb\_unit() interactive tryout

- Interactive tryout of the `fb_unit()` feedback function.

### fb\_unit() test suite

- Pre-defined tests of the `fb_unit()` feedback function.

### fb\_vars() test suite

- Pre-defined tests of the `fb_unit()` feedback function.

### fb\_vars() interactive tryout

- Interactive tryout of the `fb_vars()` feedback function.



# Meclib Development since STACK 2022

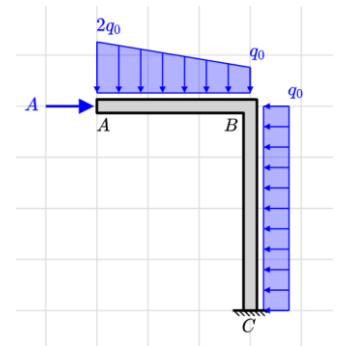
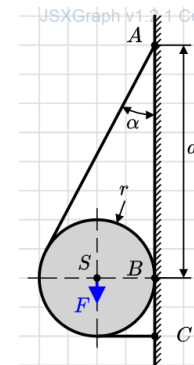
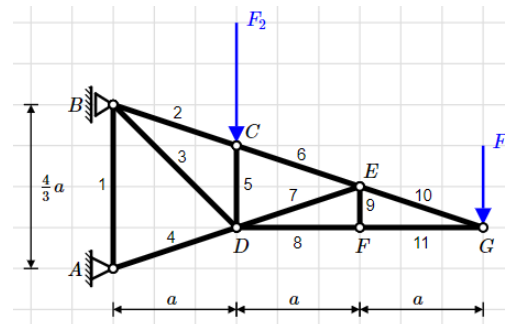
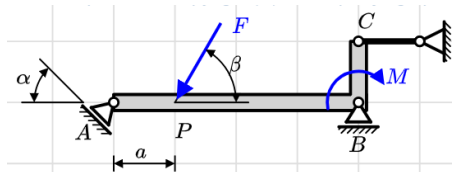
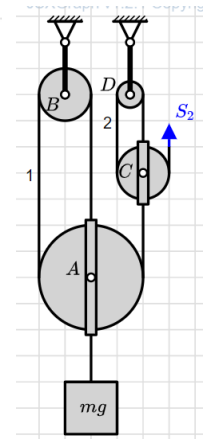
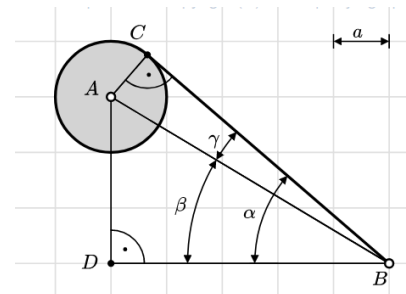
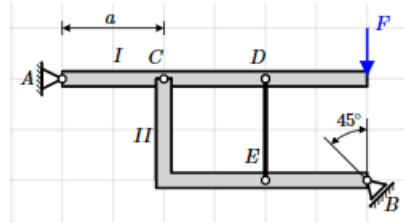
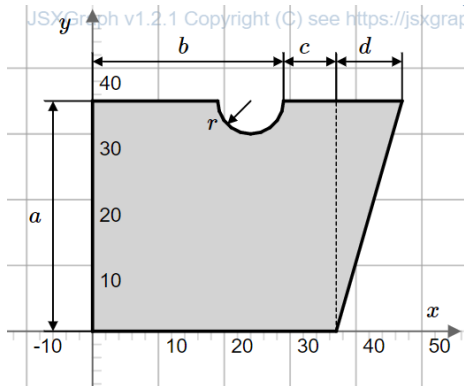
- Transition to STACK 4.4 and JSXGraph 1.4.5
- Complete switch to STACK include feature
- Minor extensions on graphics objects
- Public Demo Moodle Course at THB
- Bulk production of question with interactive graphical input
- Current Focus of development
  - Refactoring of the JS source code
  - Feedback functions for interactive graphics
  - Feedback for numerical and symbolical input
  - Automated unit tests of feedback functions



# Meclib in Action

## Engineering Mechanics (Statics)

- 128 STACK questions in total
- 102 static or randomized meclib questions
- 24 interactive questions

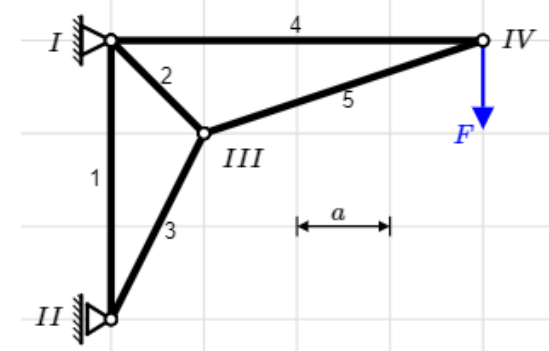
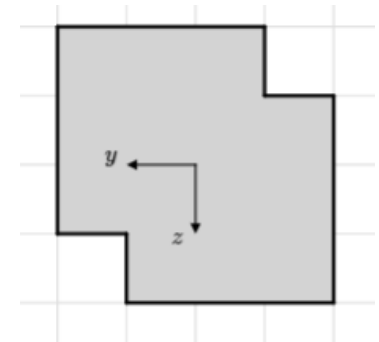
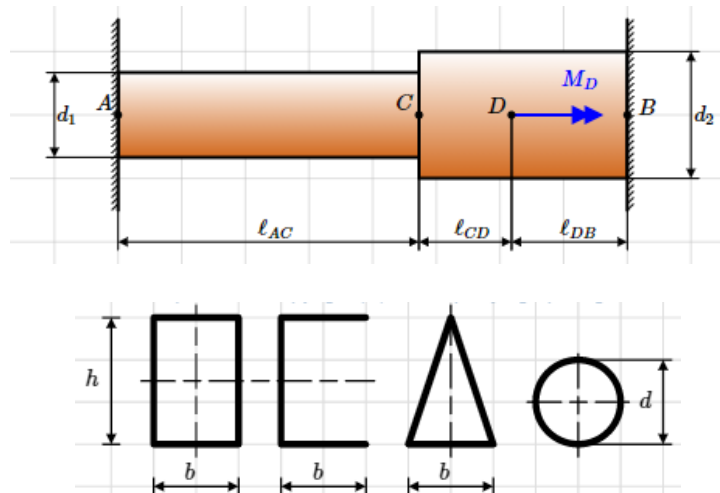
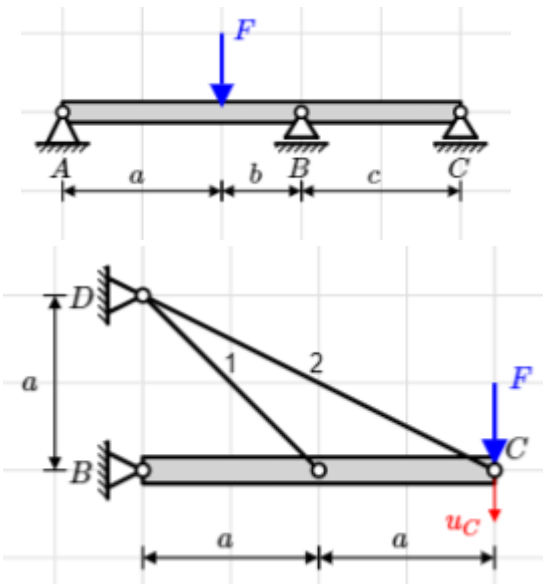
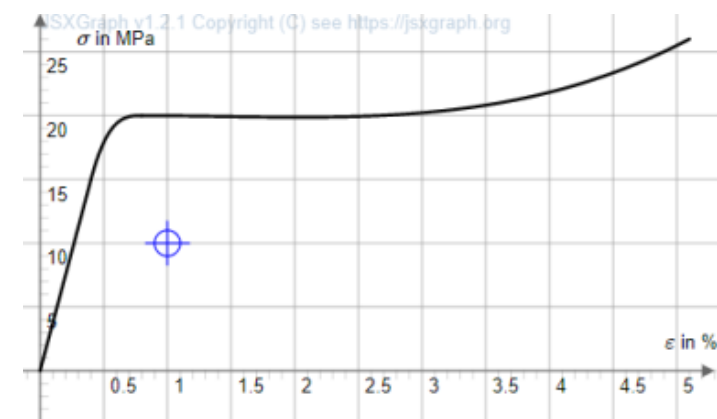




# Meclib in Action

## Strength of Materials

- 80 STACK questions in total
- 52 static or randomized meclib questions
- 8 interactive meclib questions



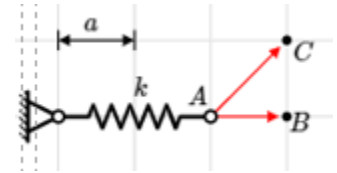
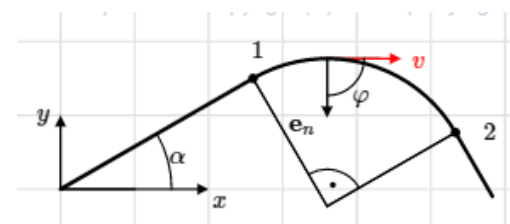
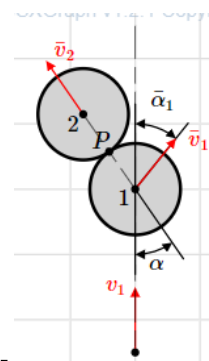
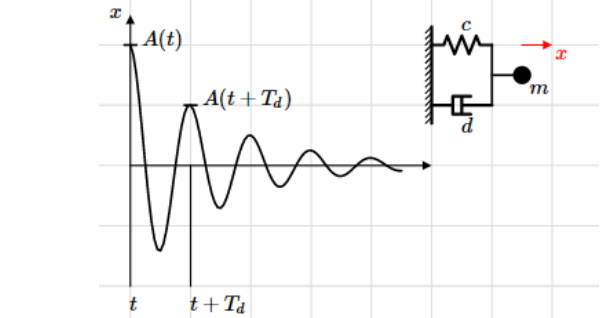
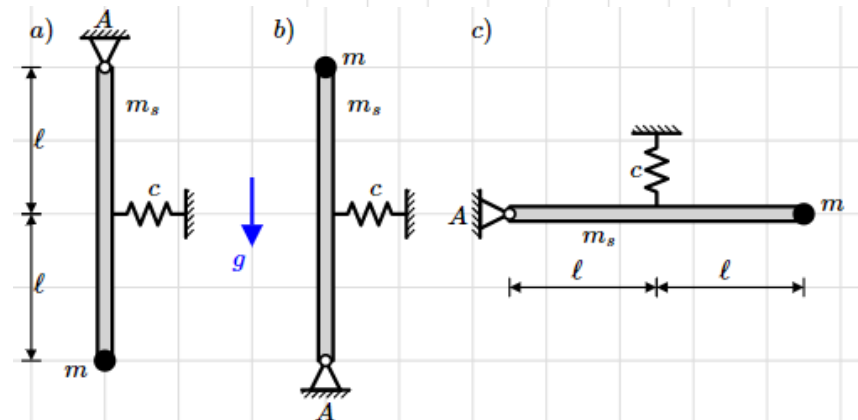
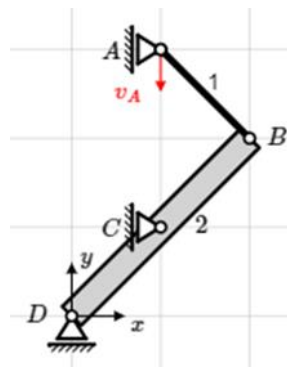
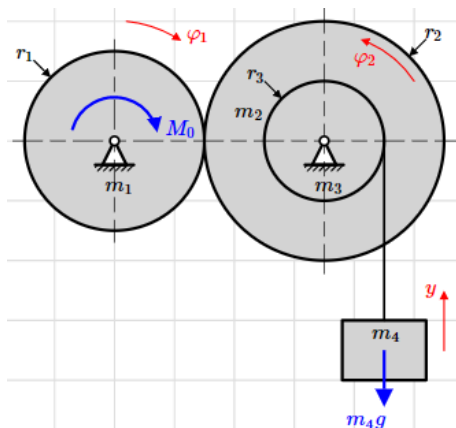
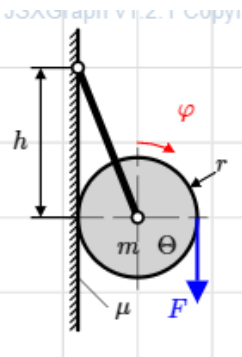




# Meclib in Action

## Dynamics

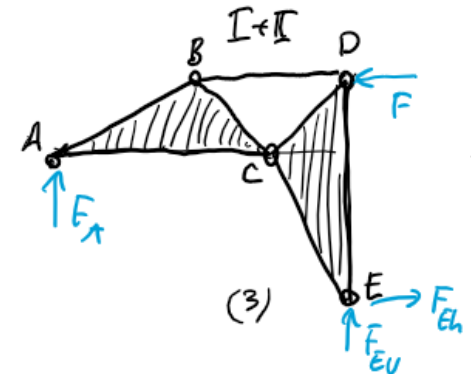
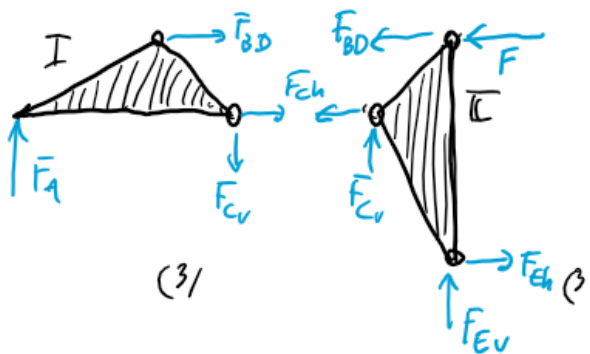
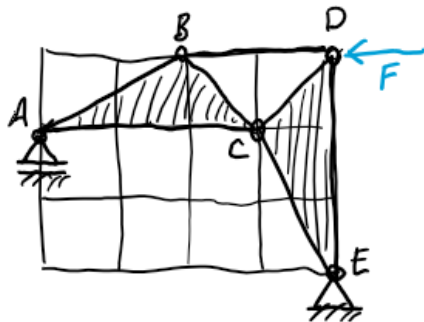
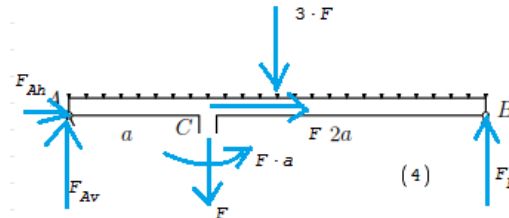
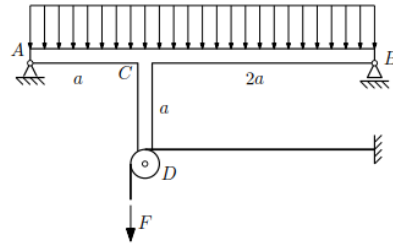
- 62 STACK questions in total
- 43 static or randomized meclib questions
- 5 interactive meclib question





# Free Body Diagrams

- Modelling technique in engineering mechanics
- Isolate the system and replace environment by forces and moments

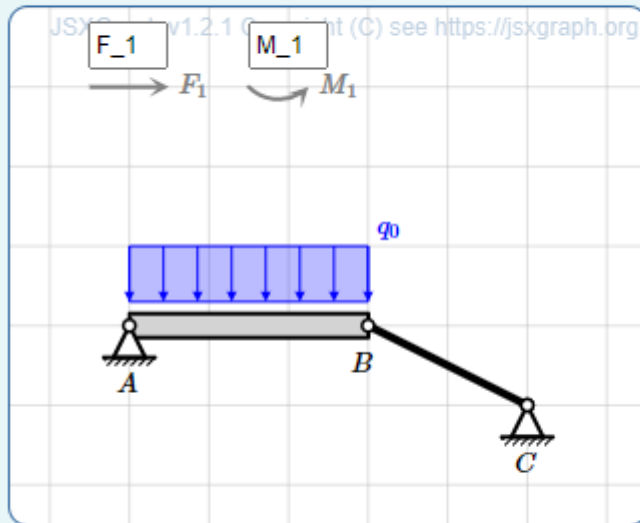




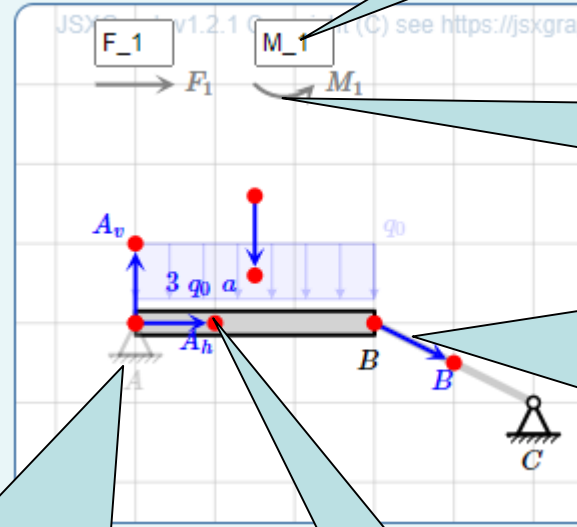
# Free Body Diagrams: Editor

Bestimmen Sie die Auflagerreaktionen für den skizzierten Balken  $AB$ .

Gegeben: Gitterweite  $a$ , Streckenlast  $q_0$ .



Schneiden Sie den Balken frei (ersetzen Sie die Reaktionen) und ersetzen Sie die Streckenlast durch eine resultierende Kraft.



Edit label of new objects

Drag to create new objects

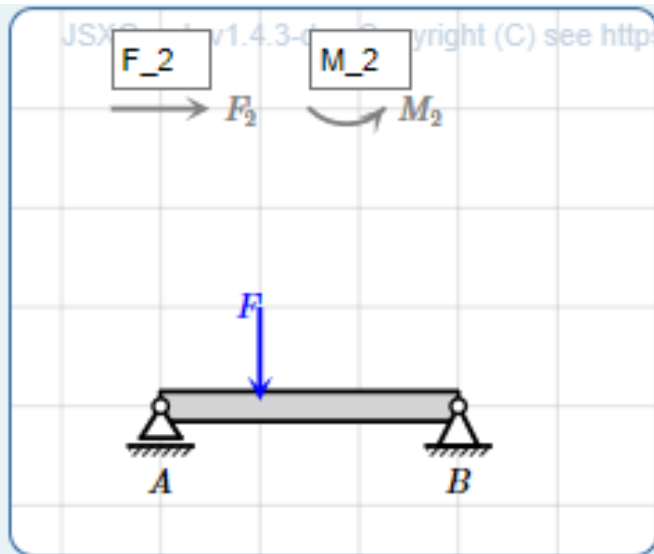
Double-click to delete forces or moments

Double-click to activate/deactivate supports or distributed loads

Drag to move control points



# Free Body Diagrams: Feedback



Nothing done. Feedback:

- Deactivate the supports
- Add 3 reactions!



Loslager A: ist nicht deaktiviert.



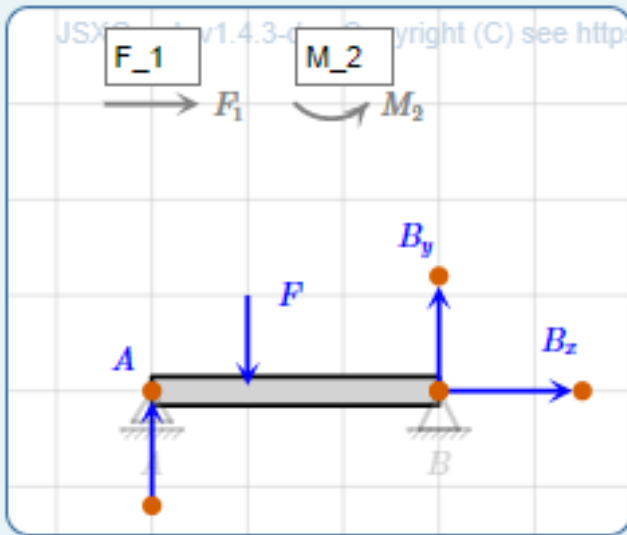
Festlager B: ist nicht deaktiviert.



Sie haben 0 Kräfte und 0 Momente platziert. Erwartet werden 3 Kräfte und kein Moment.



# Free Body Diagrams: Feedback



Loslager A: Reaktion  $A$  gefunden.



Festlager B: Reaktionen  $B_y$ ,  $B_x$  gefunden.



Sie haben wie erforderlich 3 Kräfte und 0 Momente platziert.



# Free Body Diagrams: Feedback

- Input: objects
- Input: names
- Potential response tree: Loslager
- Potential response tree: Festlager
- Potential response tree: FBD

2 nodes

2 nodes

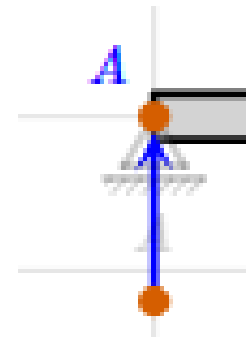
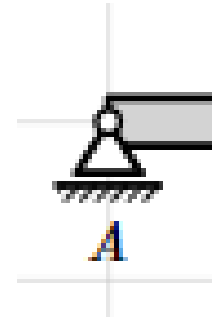
1 node

```
obj: stackjson_parse(objects);  
[text, isOK]: fb_fix1(obj, names, i_fix1, "Loslager A");  
[ntext, nOK]: fb_fix1_name(obj, names, i_fix1);
```



# Free Body Diagrams: Feedback

```
fb_fix1(o, n, i, description):=block(
  [txt, R], txt: sconcat("<br>",description,": "),
  /* Is object i a fixed support */
  if ( o[i][1] # "fix1" ) then
    return ([sconcat(txt, "object ", string(i), " is not a fixed support (fix1)", false)),
  /* Is object i de-activated */
  if ( o[i][length(o[i])] # "hide") then
    return ([sconcat(txt, " ist nicht deaktiviert."), false]),
  /* Any reactions found at i? */
  if not listp(n[i]) or n[i]=[] then
    return([sconcat(txt, "Keine Reaktion gefunden."), false]),
  /* Exactly 1 reaction found? */
  if (length(names[i]) > 1) then
    return( [sconcat(txt, "Mehr als eine Reaktion gefunden."), false]),
  /* Is the reaction a force? */
  if ( o[names[i][1]][1] # "force" ) then
    return ([sconcat(txt, "Die Reaktion muss eine Kraft sein."),false]),
  /* Now ready for examination of the reaction */
  R: o[names[i][1]],
  /* Is the force normal to the support? */
  if not parallelep(R, o[i]) then
    return ([sconcat(txt, "Die Reaktion \\", R[2], "\\) hat nicht die richtige Richtung."),false]),
  /* everything should be ok here */
  return([sconcat(txt, "Reaktion \\", R[2], "\\) gefunden."),true])
);
```





# Free Body Diagrams: Feedback

```
obj: stackjson_parse(objects);  
[text, isOK]: fb_fix1(obj, names, i_fix1, "Loslager A");  
[ntext, nOK]: fb_fix1_name(obj, names, i_fix1);
```

Answer test	AlgEquiv	SAns	isOK	TAns	true	Test			
Mod	=	Score	0.5	Penalty		Next	Node 2	Answer note	Loslager-1-T
<div>{@text@}</div>									
<div>HTML format</div>									
Mod	=	Score	0	Penalty		Next	[stop]	Answer note	Loslager-1-F
<div>{@text@}</div>									
<div>HTML format</div>									





# Unit Tests for Feedback functions

## Challenge:

- How to make sure that the feedback is not broken by new Moodle/STACK/MecLib/JSXGraph versions
- Casttext() behaviour can't be examined in the Maxima sandbox
- Actual casttext() result doesn't display verbatim using `[[debug/]]`

## Solution:

- STACK questions as unit tests and interactive tryout
- Manual construction of reference casttext() expressions
- AlgEquiv test with visual inspection of the result



# Unit Tests for Feedback functions

Question variables



```
stack_include("https://raw.githubusercontent.com/mkraska/meclib/main/Maxima/fb_value.mac" );
```

Question text



```
**T04** `fb_vars(a, a+b+omega)` Missing variables \\(b,\\omega\\).  
  
[[feedback:t04]]
```

Feedback variables



```
R: fb_vars(a, a+b+omega);  
E: casttext("&nbsp;Fehlende Variable: {@b@}, {@omega@} (<code>  
{@stack_disp_comma_separate(listify({b,omega}))@}</code>).");
```

<b>Node 1</b>		Answer test	AlgEquiv	SAns	R	TAns	E
		Test options		Quiet	No		
Node 1 when true		Mod	=	Score	1	Penalty	
Node 1 true feedback						Next	[stop]
		Answer note	t04-1-T				
		<div>{@R@}</div> <div>HTML format  <a href="#">Show syntax</a> <a href="#">Insert image</a> <a href="#">Insert file</a> <a href="#">Preview</a></div>					
Node 1 when false		Mod	=	Score	0	Penalty	
Node 1 false feedback						Next	[stop]
		Answer note	t04-1-F				
		<div>&lt;br&gt;E: {@R#} R: {@E#}</div> <div>HTML format  <a href="#">Show syntax</a> <a href="#">Insert image</a> <a href="#">Insert file</a> <a href="#">Preview</a></div>					

**T04** `fb_vars(a, a+b+omega)` Missing variables  $b, \omega$ .

✓ Fehlende Variable:  $b, \omega$  ( $b, \omega$ ).



# Unit Tests for Feedback functions

- Check them in the Demo Moodle Course <https://extmoodle.th-brandenburg.de/course/view.php?id=138&section=1>
- Work in progress (tests are added whenever modifications are required)

## Interactive Tryout and Automatic Unit Tests



### fb\_bar\_name() test suite

- Unit tests of the `fb_bar_name()` feedback function



### fb\_unidir() test suite

- Unit tests of the `fb_unidir()` feedback function



### fb\_unit() interactive tryout

- Interactive tryout of the `fb_unit()` feedback function.



### fb\_unit() test suite

- Pre-defined tests of the `fb_unit()` feedback function.



### fb\_vars() test suite

- Pre-defined tests of the `fb_unit()` feedback function.



### fb\_vars() interactive tryout

- Interactive tryout of the `fb_vars()` feedback function.



## Summary

- Meclib is a set of JavaScript objects for embedded interactive graphics in STACK questions
- Includes complex feedback functions for graphical, numeric and symbolic input
- Support resources include Wiki on Github and Demo Moodle Courses
- Concept is mature
- Implementation is work in progress
- Application is moving towards more interactivity

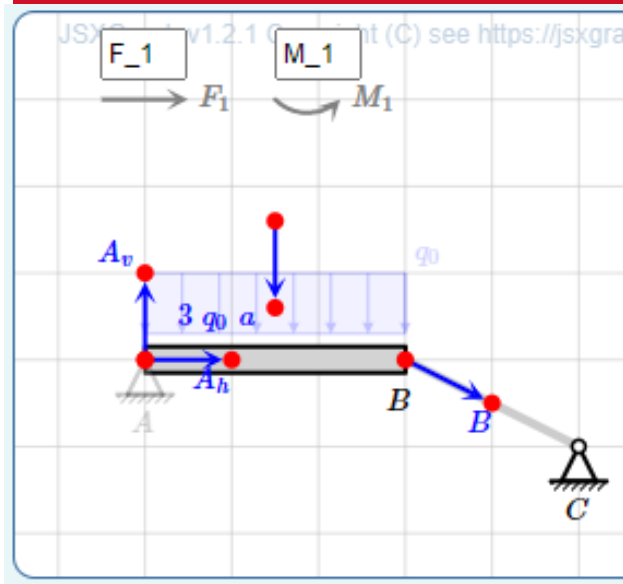


# Acknowledgements

This work was supported by

- **Department of Engineering at THB:** occasional funding of student assistants
- **Moodle and STACK teams:** providing a powerful platform
- **Matti Harjula:** STACK 4.4b test environment and STACK related coaching
- **Antti Rasila:** Curating ABACUS, where first Meclib questions have been uploaded
- **Alfred Wassermann:** Providing JSXGraph library and Meclib-related bugfixes/features

Thank you for your attention!



Prof. Dr.-Ing. Martin Kraska  
Werkstoff- und Strukturmechanik/  
Mechanics of Materials and Structures  
Maschinenbau/Mechanical Engineering  
Fachbereich Technik

Technische Hochschule Brandenburg  
University of Applied Sciences  
Magdeburger Str. 50  
14770 Brandenburg an der Havel  
Raum: 401 IWZ

Postanschrift: Postfach 2132  
14737 Brandenburg an der Havel

T +49 3381 355 356  
F +49 3381 355 66 356  
kraska@th-brandenburg.de  
Web:

[Martin Kraska](#)  
[Offene Werkstatt der THB](#)

[Studiengang Maschinenbau](#)