Anleitung für das Asymptote Modul zum Erstellen von konstruktiven Zeichnungen und statischen Systemen in LaTeX

Version 0.9

Inhaltsverzeichnis

1 Installation

Zur Installation von Asymptote empfiehlt sich die TeX Live TeX-Distribution. Dies ist die umfangreichste TeX-Distribution und Asymptote lässt sich besonders leicht einbinden. Ist TeX Live noch nicht installiert, kann der Installer unter `https://www.tug.org/texlive/acquire-netinstall.html` heruntergeladen werden. Deutsche Informationen zur Installation von TeX Live sind unter `www.dante.de/tex/tl-install-windows.html` zu finden
Der Asymptote-Installer kann unter `http://www.sourceforge.net/projects/asymptote` heruntergeladen werden. Eine umfangreiche Anleitung zu Asymptote kann über `http://asymptote.sourceforge.net/doc` erreicht werden. Dort finden sich Informationen zum allgemeinen Umgang mit Asymptote und eine Installationsanleitung.
Das Modul besteht aus fünf .asy-Dateien, die in den Asymptote-Installationsordner verschoben werden müssen. Vorhandene Dateien sind für die Funktionsfähigkeit zu überschreiben. Damit ist das Modul einsatzbereit.

2 Umgang mit dem Modul

2.1 Grundlagen

Um Asymptote in Latex nutzen zu können, muss dass notwendige Package eingebunden werden:

```
\usepackage{asymptote}
```

Zum Erstellen einer Asymptote-Grafik muss die asy-Umgebung in Latex begonnen werden:

```
\begin{asy}
:
\end{asy}
```

Um die Schriftgröße der Grafiktexte einzustellen wird folgender Befehl in der asy-Umgebung benutzt. Die Default-Schriftgröße sind 12pt, 10pt werden für die optimale Bildqualität empfohlen.

```
defaultpen(fontsize(WERT pt));
```

Jedes zusätzliche Asymptote-Modul, dass innerhalb der Grafik benutzt werden möchte, muss zuerst eingebunden werden. Dies geschieht mit dem Befehl `include`. Der Dateiname der dem Modul zugehörigen .asy-Datei entstpricht auch dem Modulnamen, der im Befehl verwendet werden muss:

```
include MODUL;
```

Die Größe der Grafik wird mit dem Befehl `size` eingestellt. Als Einheit für die Größe empfiehlt sich "cm":

```
size(BREITE cm);
```

Das Kompilieren der .tex-Datei muss über einen Spezialbefehl geschehen, der in der jeweiligen LaTeX-Benutzeroberfläche selbst erstellt werden muss. Er umfasst die Befehle:

```
txs:///pdflatex | txs:///asy | txs:///pdflatex | txs:///view-pdf
```

Jeder Asymptote-Befehl muss mit einem Semikolon abgeschlossen werden. Bei Fehlermeldungen sollte somit zuerst immer geprüft werden, ob ein Semikolon überall vorhanden ist, oder ein Semikolon zuviel gesetzt wurde.
Kommentare, die nicht ausgeführt werden sollen, sind mit `//` davor zu kennzeichnen:

```
// Dies ist ein Kommentar!  Befehle hinter den Slashes werden nicht ausgeführt!
```

Der grundlegende Befehlskopf für ein seitenbreites statisches System sieht somit wie folgt aus:

```
\begin{asy}
   include statik;
   size(14cm);
   defaultpen(fontsize(10pt));
   :
\end{asy}
```

Die Befehle bestehen aus folgenden Teilen: Dem **Befehlsnamen**, der vor der Klammer steht, und der **Variablendefinition** in den Klammern. In den folgenden Listen steht immer der Datentyp vor dem Variablennamen, dieser muss bei der Benutzung nicht angegeben werden, der eingegebene Wert muss nur das richtige Format für den jeweiligen Datentyp haben. Steht Gleichzeichen und ein Wert hinter der Variable handelt es sich dabei um einen **Default-Wert**, der auch ohne spezielle Definition der Variable durch den Nutzer benutzt wird. Es reicht somit aus, nur die Variablen ohne Default-Werte in den Klammern zu definieren, wenn die Default-Werte nicht geändert werden wollen. In Asymptote gibt es folgende Datentypen:

bool	ein Boolescher Typ, der nur die Werte `true` oder `false` annehmen kann
int	ein Integer Wert, also eine natürliche Zahl
real	eine reelle Zahl, Trennzeichen ist ein Punkt
pair	eine komplexe Zahl als Paar der reellen Komponenten, hier hauptsächlich als Punkt zu verstehen
path	ein Weg, bestehend aus Punkten und dem Verbindungstyp
string	ein Zeichenkette begrent von Anführungszeichen

Zusätzlich können auch Arrays, oder Matrizen von Datentypen erstellt werden:

```
DATENTYP       VARIABLENNAME  =  Wert;
DATENTYP[]         ARRAYNAME  =  {Wert1, Wert2, ...  , Wertn};
DATENTYP[][]    MATRIZENNAME  =  {{Wert11, Wert12, ...  , Wert1n},
                                 {Wert21, Wert22, ...  , Wert2n},
                                 {Wertm1, Wertm2, ...  , Wertmn}};
```

2.2 Statische Befehle (statik.asy)

2.2.1 Balken/Gelenke

Definieren eines **Punktes**:

```
pair PUNKTNAME = (X-KOORDINATE,Y-KOORDINATE);
```

Definieren eines **Weges** (Polygonzug):

```
path WEGNAME = 1.-PUNKT--2.-PUNKT--...--n.-PUNKT;
```

Erstellen eines **Balkens**:

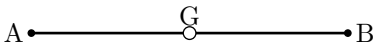
```
balken (path a, pen p=defaultpen+1);

    a  =  Weg des Balkens (Polygonzug)
    p  =  Strichdicke
```

Erstellen eines **unten gestrichelten Balkens**:

```
balkengestrichelt (pair p1, pair p2, real a=0.1, pen pb=defaultpen+1);

    p1  =  Punkt A
    p2  =  Punkt B
     a  =  Abstand der Strichelung vom Balken
    pb  =  Strichdicke des Balkens
```

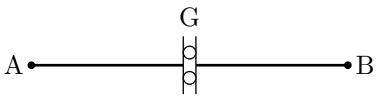
Erstellen eines **Momenten-Gelenks**:

```
gelenk (pair z, pair v=(0,0), real r=0.1, pen p=defaultpen);

    z  =  Zentrum des Gelenks (Mittelpunkt des Gelenks)
    v  =  Verschiebung des Gelenkes vom Zentrum
    r  =  Radius des Gelenks
    p  =  Strichdicke
```



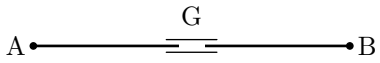
Erstellen eines **Querkraft-Gelenks**:

```
querkraftgelenk (pair z, int a=0, pen p=penlager, real b=0.9);

    z  =  Zentrum des Gelenks (Mittelpunkt des Gelenks)
    a  =  Verschiebung des Gelenkes vom Zentrum
    p  =  Strichdicke
    b  =  Lager-"Größe"
```



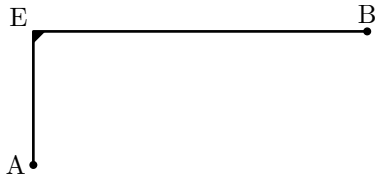
Erstellen eines **Normalkraft-Gelenks**:

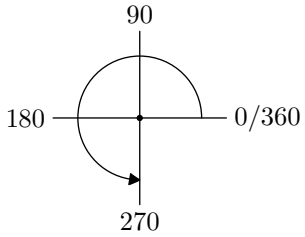
```
normalkraftgelenk (pair z, int a=0, pen pen=penlager, real breite=0.8);

    z  =  Zentrum des Gelenks (Mittelpunkt des Gelenks)
    a  =  Verschiebung des Gelenkes vom Zentrum
    p  =  Strichdicke
    b  =  Lager-"Größe"
```



Erstellen einer **biegesteifen Ecke**:

```
biegesteifeecke (pair z, int w1, int w2, real l=0.18);

     z  =  Punkt der Ecke
    w1  =  Startwinkel
    w2  =  Endwinkel der Ecke (Definition siehe unten)
     b  =  Ecken-"Größe"
```





2.2.2 Auflager

Erstellen eines **festen Lagers**:

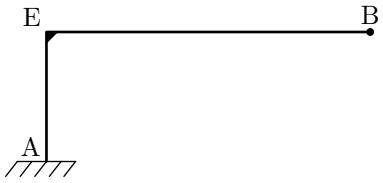
```
lagerfest (pair z, int w=0, bool g=true, pair v=(0,0), pen p=penlager, pen
           pg=penlager, real s=0.5);

     z  =  Punkt des Auflagers
     w  =  Drehwinkel des Auflagers
     g  =  Gelenkkugel anzeigen
     v  =  Verschiebung des Auflagers von z
     p  =  Strichstärke des Lagers
    pg  =  Strichstärke der Gelenkkugel
     s  =  Seitenlänge des Lagers
```

Erstellen eines **verschieblichen Lagers**:

```
lagerverschieblich (pair z, int w=0, bool g=true, pair v=(0,0), pen p=penlager,pen
                    pg=penlager, real s=0.5, real b=0.15);

     z  =  Punkt des Auflagers
     w  =  Drehwinkel des Auflagers
     g  =  Gelenkkugel anzeigen
     v  =  Verschiebung des Auflagers von z
     p  =  Strichstärke des Lagers
    pg  =  Strichstärke der Gelenkkugel
     s  =  Seitenlänge des Lagers
     b  =  Abstand der Gleitlinie
```

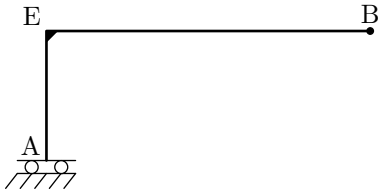
Erstellen einer **Einspannung**:

```
einspannung (pair z, int w=0, pen p=penlager, real b=0.9);

    z  =  Punkt des Auflagers
    w  =  Drehwinkel des Auflagers
    p  =  Strichstärke des Lagers
    b  =  Breite des Lagers
```



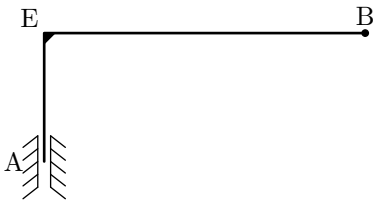
Erstellen eines **Querkraft-Lagers**:

```
querkraftlager (pair z, int w=0, pen p=penlager, real b=0.9);

    z  =  Punkt des Auflagers
    w  =  Drehwinkel des Auflagers
    p  =  Strichstärke des Lagers
    b  =  Breite des Lagers
```



Erstellen eines **Normalkraft-Lagers**:

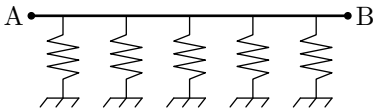
```
normalkraftlager (pair z, int w=0, pen p=penlager, real b=0.9);

    z  =  Punkt des Auflagers
    w  =  Drehwinkel des Auflagers
    p  =  Strichstärke des Lagers
    b  =  Breite des Lagers
```



2.2.3 Federn

Erstellen einer **Stabbettung**:

```
bettung (pair z, real l, int w=0, pen p=penlager, real do=0.3, real du=0.3, real
        df=0.1, real bf=0.5, int n=6, real b=0.5);

     z  =  Anfangspunkt der Bettung
     l  =  Länge der Bettung
     w  =  Drehwinkel der Bettung
     p  =  Strichstärke des Bettung
    do  =  Abstand der Federn vom Balken
    du  =  Abstand der Federn vom Boden
    df  =  Abstand der Federspulen
    bf  =  Breite der Federn
     n  =  Anzahl der Spulen
     b  =  Breite der Festhaltung
```



Erstellen einer **Wegfeder**:

```
wegfeder (pair p, int w=0, pen p=penlager, real do=0.3, real du=0.3, real df=0.1,
         real bf=0.5, int n=6, real b=0.5);

     z  =  Angriffspunkt der Feder
     w  =  Drehwinkel der Feder
     p  =  Strichstärke der Feder
    do  =  Abstand der Feder vom Balken
    du  =  Abstand der Feder vom Boden
    df  =  Abstand der Federspulen
    bf  =  Breite der Federn
     n  =  Anzahl der Spulen
     b  =  Breite der Festhaltung
```

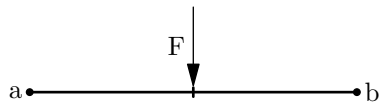
Erstellen einer **Wegfeder**:

```
drehfeder (pair z, int w=0, pen p=penlager);

    z  =  Angriffspunkt der Feder
    w  =  Drehwinkel der Feder
    p  =  Strichstärke der Feder
```

2.2.4 Lasten

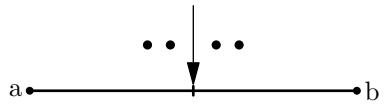
Erstellen einer **Einzellast**:

```
einzellast (pair z, int w=0, bool r=true, string s="F", real a=0.1, real l=1.2,
            pen p=penlast, real as=8, real x=1.0);

     z  =  Angriffspunkt der Last
     w  =  Drehwinkel der Last
     r  =  Richtung der Last
     s  =  Beschriftung der Last
     a  =  Abstand vom Stab
     l  =  Länge des Pfeils
     p  =  Strichstärke
    as  =  Pfeilkopfgröße
     x  =  Maßstabsfaktor
```



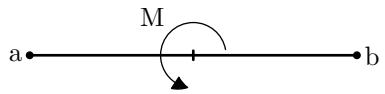
Erstellen einer **Wanderlast**:

```
wanderlast (pair z, int w=0, real a=0.1, real l=1.2, pen p=penlast, real as=8,
            real pa=0.35, real yp=0.7, real r=0.07);

     z  =  Angriffspunkt der Last
     w  =  Drehwinkel der Last
     a  =  Abstand vom Stab
     l  =  Länge des Pfeils
     p  =  Strichstärke
    as  =  Pfeilkopfgröße
    pa  =  Abstand der Punkte voneinander
    yp  =  Abstand der Punkte vom Stab
     r  =  Radius der Punkte
```



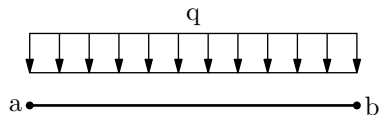
Erstellen eines **Einzelmoments**:

```
einzelmoment (pair z, int w1=0, int w2=270, bool d=true, string s="M", align
              pos=NoAlign, real r=0.5, pen p=penlast, int wa=10, real as=5);

      z  =  Angriffspunkt der Last
     w1  =  Anfangswinkel der Last
     w2  =  Endwinkel der Last
      d  =  Richtung der Last (gegen den Uhrzeigersinn true)
      s  =  Beschriftung
    pos  =  Ausrichtung der Beschriftung
      r  =  Radius des Moments
      p  =  Linienstärke
     wa  =  Winkelabstand vom Anfangs- und Endwinkel
     as  =  Pfeilkopfgröße
```



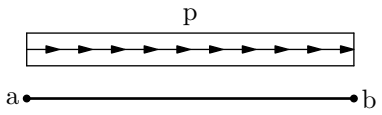
Erstellen einer **Linienlast in z-Richtung**:

```
streckenlast (pair p1, pair p2, bool r=true, string s="q",pen p=penlast, real
              a=0.5, real l=0.6, real b=0.5, real as=5);

    p1  =  Startpunkt
    p2  =  Endpunkt
     r  =  Richtung (zum Stab true)
     s  =  Beschriftung
     p  =  Linienstärke
     a  =  Abstand vom Stab
     l  =  Pfeillänge
     b  =  Abstand der Pfeile untereinander
    as  =  Pfeilkopfgröße
```



Erstellen einer **Linienlast in x-Richtung**:

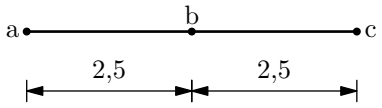
```
streckenlastdehnstab (pair p1, pair p2, bool r=true, string s="p",pen p=penlast,
                      real a=0.5, real l=0.5, real b=0.5, real as=5);

    p1  =  Startpunkt
    p2  =  Endpunkt
     r  =  Richtung (zum Stab true)
     s  =  Beschriftung
     p  =  Linienstärke
     a  =  Abstand vom Stab
     l  =  Pfeillänge
     b  =  Abstand der Pfeile untereinander
    as  =  Pfeilkopfgröße
```



2.2.5 Beschriftung/Bemaßung

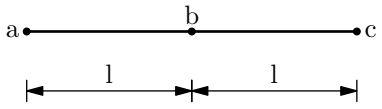
Erstellen einer **horizontalen Bemaßung**:

```
bemassunghorizontal (real[] l, real x=0, real y=-1.5, pen p=penbemassung, arrowbar
                     b=Bars(size=8), arrowbar b2=Arrows(size=6))

     l  =  Längen-Array
     x  =  Anfangs-X-Wert der Bemassung
     y  =  Anfangs-Y-Wert der Bemassung
     p  =  Linienstärke
     b  =  Striche zwischen Maßteilen
    b2  =  Pfeile zwischen den Maßteilen
```



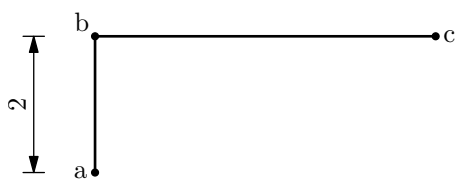
Erstellen einer **horizontalen Bemaßung mit bestimmtem Wert**:

```
bemassunghorizontalwert (real[] l, real x=0, real y=-1.5, pen p=penbemassung,
                         string s="", arrowbar b=Bars(size=8), arrowbar
                         b2=Arrows(size=6))

     l  =  Längen-Array
     x  =  Anfangs-X-Wert der Bemassung
     y  =  Anfangs-Y-Wert der Bemassung
     p  =  Linienstärke
     s  =  Beschriftung
     b  =  Striche zwischen Maßteilen
    b2  =  Pfeile zwischen den Maßteilen
```



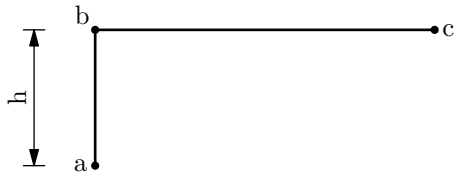
Erstellen einer **vertikalen Bemaßung**:

```
bemassungvertikal (real[] l, real y=0, real x=-1.5, pen p=penbemassung, arrowbar
                   b=Bars(size=8), arrowbar b2=Arrows(size=6))

     l  =  Längen-Array
     y  =  Anfangs-Y-Wert der Bemassung
     x  =  Anfangs-X-Wert der Bemassung
     p  =  Linienstärke
     b  =  Striche zwischen Maßteilen
    b2  =  Pfeile zwischen den Maßteilen
```



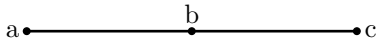
Erstellen einer **vertikalen Bemaßung mit bestimmtem Wert**:

```
bemassungvertikalwert (real[] l, real y=0, real x=-1.5, string s="", pen
                       p=penbemassung, arrowbar b=Bars(size=8), arrowbar
                       b2=Arrows(size=6))

     l   =  Längen-Array
     y   =  Anfangs-Y-Wert der Bemassung
     x   =  Anfangs-X-Wert der Bemassung
     s   =  Beschriftung
     p   =  Linienstärke
     b   =  Striche zwischen Maßteilen
    b2   =  Pfeile zwischen den Maßteilen
```



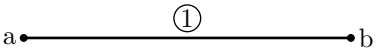
Erstellen einer **Knotenbeschriftung**:

```
knoten (string s, pair p, align r)

    s   =  Beschriftung
    p   =  Knotenpunkt
    r   =  Ausrichtung der Beschriftung
```



Erstellen einer **umkreisten Beschriftung (z.B. Stabnummerierung)**:

```
ustring (pair a, string s, real x=10)

    a   =  Punkt der Beschriftung
    s   =  Beschriftung
    x   =  Schriftgröße
```



Erstellen eines **Textfelds**:

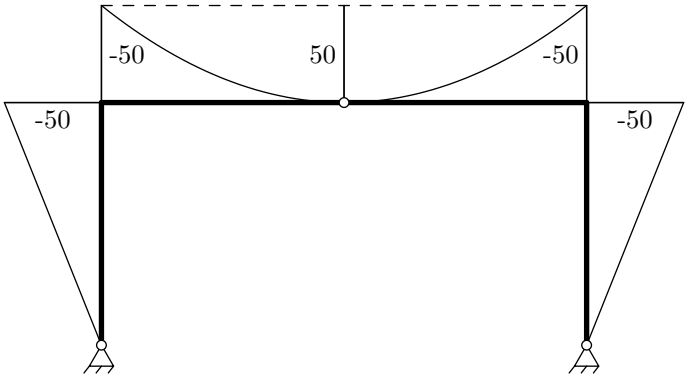
```
textfeld (pair a, string s, real b, real x=1.0)

    a   =  Ort der Nummerierung
    s   =  Text
    b   =  Breite des Textfeldes in cm
    x   =  MaÃ§stabsfaktor
```

Das hier ist sinnloser Text, weil
mir nichts besseres eingefallen ist

2.2.6 Verläufe

Erstellen eines **Schnittgrößenverlaufs**:

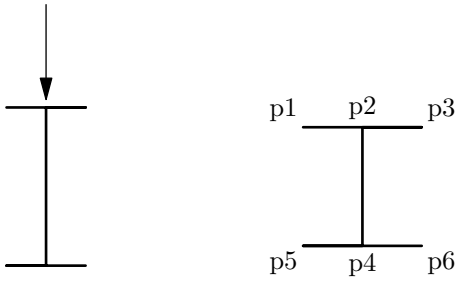
```
verlauf (pair a, pair e, real ma=1, real mp=0, real me=ma, bool par=true,
         bool werte=true, real lges=15, real mmax=ma, align pos=NoAlign, pen
         p=defaultpen)

        a  =  Stababschnittanfang
        e  =  Stababschnittende
       ma  =  Anfangsmoment
       mp  =  Parabelstich
       me  =  Endmoment
      par  =  Parabel?
    werte  =  Werte anzeigen?
     lges  =  Gesamtlänge des Systems
     mmax  =  Maximalmoment im System
      pos  =  Ausrichtung der Beschriftung
        p  =  Strichstärke
```



2.2.7 Zusatz

Erstellen eines **Balkenprofils**:

```
profil (pair a, string s="IPE", real w=0, string s2=" ", real m=0.5, bool
         el=false, string s3="p2", int w2=0, pen p=pensystem+1)

     a  =  Mittelpunkt des Profils
     s  =  Profilart (IPE oder HEA)
     w  =  Drehwinkel
    s2  =  Beschriftung
     m  =  Maßstabsfaktor
    el  =  Kraftangriff am Profil anzeigen?
    s3  =  Ort der Last (Definition siehe unten)
    w2  =  Drehwinkel der Last
     p  =  Strichstärke
```

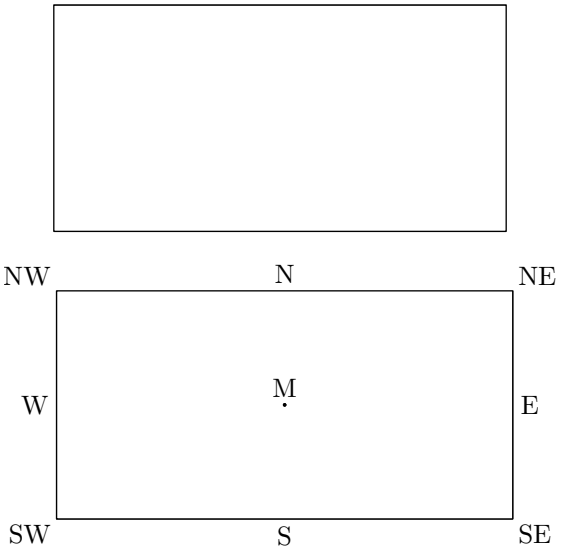


2.3 Technische Befehle (konstruktiv.asy)

2.3.1 Grundformen

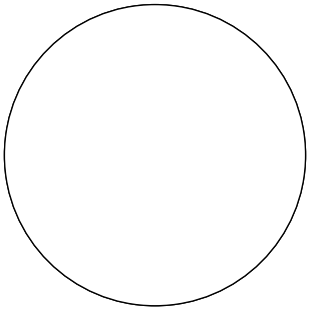
Erstellen eines **Rechtecks**:

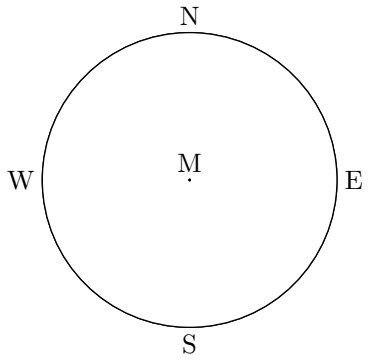
```
rechteck (pair e, real b, real h, string s="white", string ausr="M", pen
        p=defaultpen)

      e  =  Erstellpunkt
      b  =  Breite
      h  =  Höhe
      s  =  Schraffurart
   ausr  =  Ausrichtung des Erstellpunktes (Lage von a, siehe unten)
      p  =  Strichstärke
```



Erstellen eines **Kreises**:

```
kreis(pair e, real r, string s="white", string ausr="M", pen p=defaultpen)

      e  =  Erstellpunkt
      r  =  Radius
      s  =  Schraffurart
   ausr  =  Ausrichtung des Erstellpunktes (Lage von a, siehe unten)
      p  =  Strichstärke
```

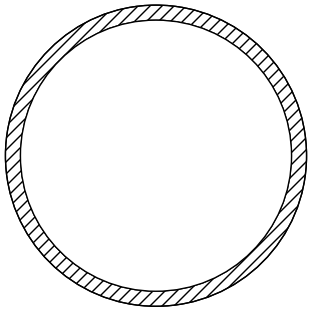




2.3.2 Querschnitte

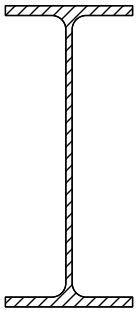
Erstellen eines **Rohrquerschnitts**:

```
rohr (pair e, real d, real t, string s1="stahl", string ausr="M", pen
      p=defaultpen, string s2="white")

       e  =  Erstellpunkt
       d  =  Durchmesser
       t  =  Blechdicke
      s1  =  Schraffurart Rohrquerschnitt
    ausr  =  Ausrichtung des Erstellpunktes (Lage von a, siehe Kreis)
       p  =  Strichstärke
      s2  =  Schraffurart innen
```



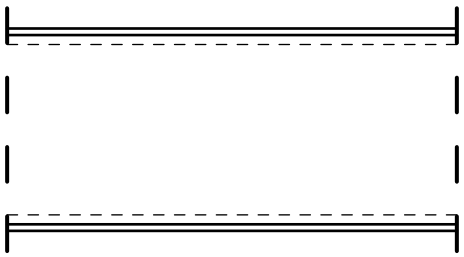
Erstellen eines **Doppel-T-Profils nach DIN EN 10 034**:

```
PROFILKLASSE (pair e, string kl, pen p=defaultpen, real w=0, string ausr="M",
              string s="stahl");

       e  =  Erstellpunkt
      kl  =  Klasse(z.B. "300")
       p  =  Strichstärke
       w  =  Drehwinkel
      s1  =  Schraffurart Rohrquerschnitt
    ausr  =  Ausrichtung des Erstellpunktes (Lage von a, siehe Rechteck)
       s  =  Schraffurart
```



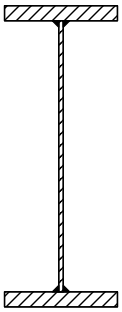
Erstellen einer **Ansicht eines Doppel-T-Profils nach DIN EN 10 034**:

```
Iansicht (pair p0, string tp, real lges=1000, real w=0, bool ra=true, bool
          ls=true, bool rs=true, string ausr="M", pen p=defaultpen);

      p0  =  Erstellpunkt
      tp  =  Profilname (z.B. "HEA300")
    lges  =  Gesamtlänge des Trägerstücks
       w  =  Drehwinkel
      ra  =  Achse des Ausrundungsendes anzeigen?
      ls  =  Schnitt am linken Ufer anzeigen?
      rs  =  Schnitt am rechten Ufer anzeigen?
    ausr  =  Ausrichtung des Erstellpunktes (Lage von a, siehe Rechteck)
       p  =  Strichstärke
```



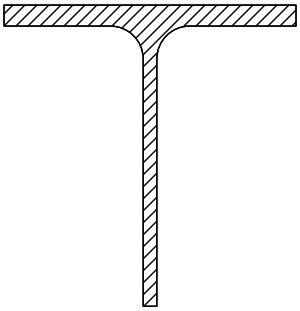
Erstellen eines **geschweißten Doppel-T-Profils**:

```
Iprofil (pair e, real b, real h, real tw, real tf, real a, pen p=defaultpen, real
         w=0, string ausr="M", real b2=b, real tf2=tf, string s="stahl")

       e  =  Erstellpunkt
       b  =  Breite
       h  =  Höhe
      tw  =  Stegdicke
      tf  =  Flanschdicke
       a  =  Schweißnahtdicke
       p  =  Strichstärke
       w  =  Drehwinkel
    ausr  =  Ausrichtung des Erstellpunktes (Lage von a, siehe Rechteck)
      b2  =  Breite des unteren Teils
     tf2  =  Flanschdicke des unteren Flansches
       s  =  Schraffurart
```



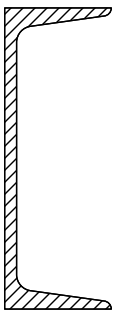
Erstellen eines **T-Profils**:

```
Tprofil (pair e, real b, real h, real tw, real tf, real r, bool rund=true, string
        s="stahl", pen p=defaultpen, real w=0, string ausr="M")

      e  =  Erstellpunkt
      b  =  Breite
      h  =  Höhe
     tw  =  Stegdicke
     tf  =  Flanschdicke
      r  =  Ausrundung/SchweiÃ§nahtdicke
   rund  =  Ausrundung(true) oder SchweiÃ§naht(false)
      s  =  Schraffurart
      p  =  Strichstärke
      w  =  Drehwinkel
   ausr  =  Ausrichtung des Erstellpunktes (Lage von a, siehe Rechteck)
```



Erstellen eines **U-Profils**:

```
U (pair e, string kl, pen p=defaultpen, real w=0, string ausr="M", string
        s="stahl")

      e  =  Erstellpunkt
     kl  =  Klasse(z.B. "300")
      p  =  Strichstärke
      w  =  Drehwinkel
   ausr  =  Ausrichtung des Erstellpunktes (Lage von a, siehe Rechteck)
      s  =  Schraffurart
```



Erstellen eines **Trapezblechs nach DIN 18807**:

```
Trapez (pair e, string kl, real lges=1000, pen p=defaultpen+2)

       e  =  Erstellpunkt
      kl  =  Klasse(z.B. "158/250")
    lges  =  Gesamtlänge
       p  =  Strichstärke
```

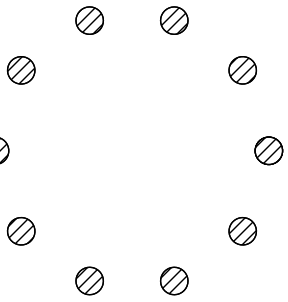
2.3.3 Bewehrung

Erstellen einer **Eck-Bewehrung**:

```
rbewehrung (pair p0, real x, real y, real d, string s="stahl", pen p=defaultpen)

    p0  =  Mittelpunkt
     x  =  Abstand vom Mittelpunkt auf der X-Achse
     y  =  Abstand vom Mittelpunkt auf der Y-Achse
     d  =  Durchmesser der Bewehrung
     s  =  Schraffurart
     p  =  Strichstärke
```

Erstellen einer **Kreis-Bewehrung**:

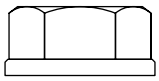
```
kbewehrung (pair p0, real r, real d, int n, string s="stahl", pen p=defaultpen)

    p0  =  Mittelpunkt
     r  =  Radius
     d  =  Durchmesser der Bewehrung
     n  =  Anzahl der StÃďbe
     s  =  Schraffurart
     p  =  Strichstärke
```



2.3.4 Verbindungsmittel

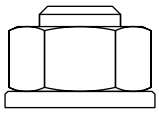
Erstellen einer **Schraubenkopf-Ansicht**:

```
schraubenkopf (pair e, string s, real w=0, pen p=defaultpen+0.5)

    e  =  Mittelpunkt der Schraube auf dem Blech
    s  =  Schraubentyp
    w  =  Drehwinkel
    p  =  Strichstärke
```



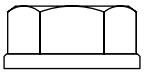
Erstellen einer **Schraubenmutter-Ansicht**:

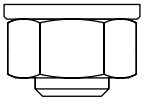
```
schraubenmutter (pair e, string s, real w=0, pen p=defaultpen+0.5)

    e  =  Mittelpunkt der Schraube auf dem Blech
    s  =  Schraubentyp
    w  =  Drehwinkel
    p  =  Strichstärke
```



Erstellen einer **Schrauben-Ansicht**:

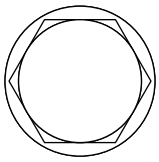
```
schraube (pair e, real t, string s, real w=0, bool sft=false, pen
      p=defaultpen+0.5)

      e  =  Mittelpunkt der Verbindungsbleche auf Schraubenachse
      t  =  Dicke der Verbindungsbleche (Summe)
      s  =  Schraubentyp
      w  =  Drehwinkel
    sft  =  Schaft gestrichelt anzeigen?
      p  =  Strichstärke
```





Erstellen einer **Schrauben-Draufsicht**:

```
schraubend (pair a, string s, pen p=defaultpen+0.5)

    e  =  Mittelpunkt der Verbindungsbleche auf Schraubenachse
    s  =  Schraubentyp
    p  =  Strichstärke
```

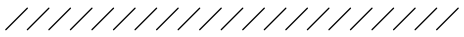


Erstellen einer **Schweißnaht**:

```
naht (pair e, real a, string n, real t, real w=0, bool spiegeln=false)

     e  =  Verbindungsmittelpunkt
     a  =  Schweißnahtstärke
     t  =  Blechdicke
     w  =  Drehwinkel
    sp  =  Spiegeln?
     n  =  Schweißnahttyp:
           "Kehl"  ≙  Kehlnaht  "Doppelkehl"  ≙  Doppelkehlnaht
              "V"  ≙  V-Naht            "DV"  ≙  Doppel-V-Naht
             "HV"  ≙  HV-Naht          "DHV"  ≙  Doppel-HV-Naht
              "Y"  ≙  Y-Naht            "DY"  ≙  Doppel-Y-Naht
             "HY"  ≙  HY-Naht          "DHY"  ≙  Doppel-HY-Naht
```

Erstellen einer **Schweißnaht in der Draufsicht**:

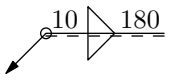
```
nahtd (pair e, real l, real w, real a=6, bool turn=false, real dl=10)

       e  =  Anfangspunkt
       l  =  Schweißnahtlänge
       w  =  Drehwinkel
       a  =  Schweißnahtstärke
    turn  =  Schraffierung umdrehen?  (links und unten von Objekten)
      dl  =  Abstand der Schraffur
```



2.3.5 Beschriftung/Bemaßung

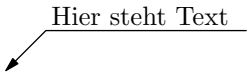
Erstellen einer **Schweißnahtbeschriftung**:

```
nahtbeschr (pair p0, real a, real l, string n, bool um=false, bool ausr=true, bool
            turn=false)

      p0  =  Beschriftungspunkt
       a  =  Schweißnahtstärke
       l  =  Schweißnahtlänge
       n  =  Schweißnahttyp
      um  =  umlaufende Naht?
    ausr  =  Strichelung unten?
    turn  =  Schweißnaht gespiegelt?
```



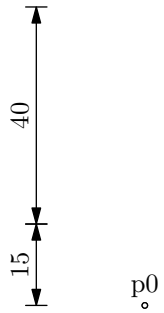
Erstellen einer **Führunglinie**:

```
flinie (pair p0, string b, int f=15)

    p0  =  Beschriftungspunkt
     b  =  Beschriftungstext
     f  =  Länge des schrägen Teils
```



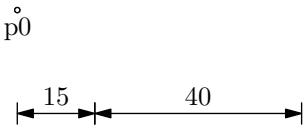
Erstellen einer **vertikalen Bemaßung**:

```
vbemassung (real[] l, pair p0, real x=20, pen p=defaultpen, arrowbar
           bar=Bars(size=8), arrowbar bar2=Arrows(size=6))

      l  =  Längen-Array der Bemaßungskette
     p0  =  Beschriftungspunkt
      x  =  Abstand von p0
      p  =  Schrichstärke
    bar  =  Balken zwischen den Bemaßungen
   bar2  =  Pfeile zwischen den Bemaßungen
```



Erstellen einer **horizontalen Bemaßung**:

```
hbemassung (real[] l, pair p0, real y=20, pen p=defaultpen, arrowbar
           bar=Bars(size=8), arrowbar bar2=Arrows(size=6))

      l  =  Längen-Array der Bemaßungskette
     p0  =  Beschriftungspunkt
      y  =  Abstand von p0
      p  =  Schrichstärke
    bar  =  Balken zwischen den Bemaßungen
   bar2  =  Pfeile zwischen den Bemaßungen
```



2.3.6 Verbundbau

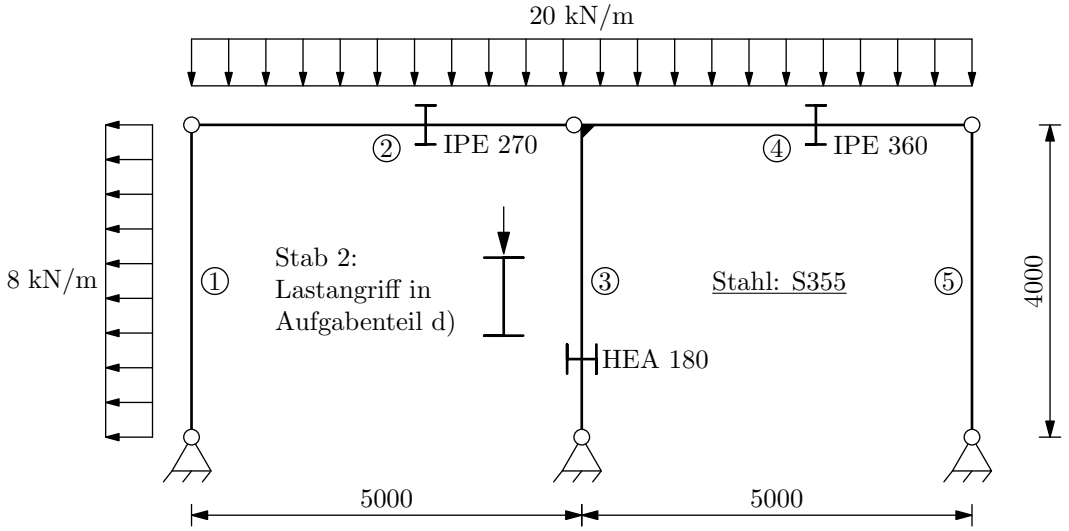
Erstellen eines **Verbundträgers** (Beispiel siehe 3.2):

```
Vtrager (string tp, real dt, real dp, real l=1000, bool bem=true, real x=80, bool
         bew=false, real ds=20, real e=80, real cnom=40)

      tp   =  Stahlprofil (z.B. "HEA300")
      dt   =  Höhe Trapezblech
      dp   =  Höhe Betonplatte
       l   =  Breite des Ausschnitts
     bem   =  Bemaßung anzeigen?
       x   =  Abstand der Bemaßung vom Träger
     bew   =  Bewehrung anzeigen?
      ds   =  Stabdurchmesser der Bewehrung
       e   =  Abstand der Bewehrungsstäbe
    cnom   =  Betondeckung
```

3 Beispiele für den Gebrauch

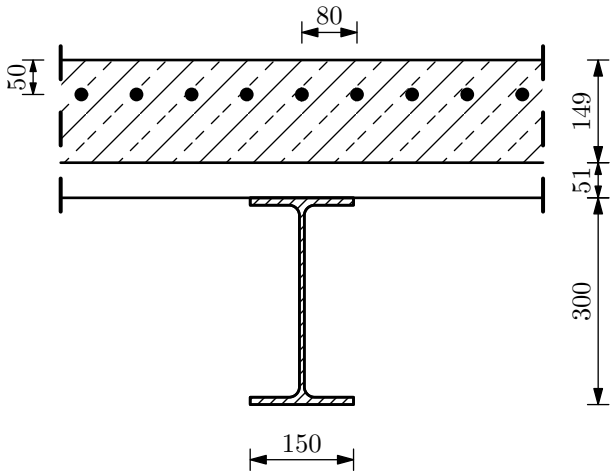
3.1 Rahmensystem Stahlbau II WS2012/13

```
\begin{asy}
   include konstruktiv;
   include statik;
   size(14cm);
   defaultpen(fontsize(10pt));

   // Definition der Punkte
   pair a=(0,0);
   pair b=(0,4.001);
   pair c=(5,0);
   pair d=(5,4);
   pair e=(10,0);
   pair f=(10,4);

   // Balken
   balken(a-b-d-f-e,pensystem);
   balken(c-d,pensystem);

   // Lager
   lagerfest(a);
   lagerfest(c);
   lagerfest(e);

   // Verbindungen
   gelenk(b);
   gelenk(d,(-0.1,0));
   gelenk(f);
   biegesteifeecke(d,0,270);

   // Bemassung
   real[] bh={5,5};
   bemassunghorizontalwert(bh,0,-1,"5000");
   real[] bv={4};
   bemassungvertikalwert(bv,0,11,"4000");

   // Belastung
   streckenlast(b,f,true,"20 kN/m");
   streckenlast(a,b,false,"8 kN/m");

   // Profile
   profil(c+(0,1),"HEA",90,"HEA 180");
   profil(b+(3,0),"IPE",0,"IPE 270");
   profil(d+(3,0),"IPE",0,"IPE 360");

   profil((4,1.8),"IPE",0,"",1.0,true);

   // Beschriftung
   ustring(a+(0.3,2),"1");
   ustring(b+(2.5,-0.3),"2");
   ustring(c+(0.3,2),"3");
   ustring(d+(2.5,-0.3),"4");
   ustring(e+(-.3,2),"5");
   textfeld((1,2.5),"Stab 2:\\ Lastangriff in\\ Aufgabenteil d)",2.5);
   textfeld((6.6,2.2),"\underline{Stahl:  S355}",2.5);
\end{asy}
```



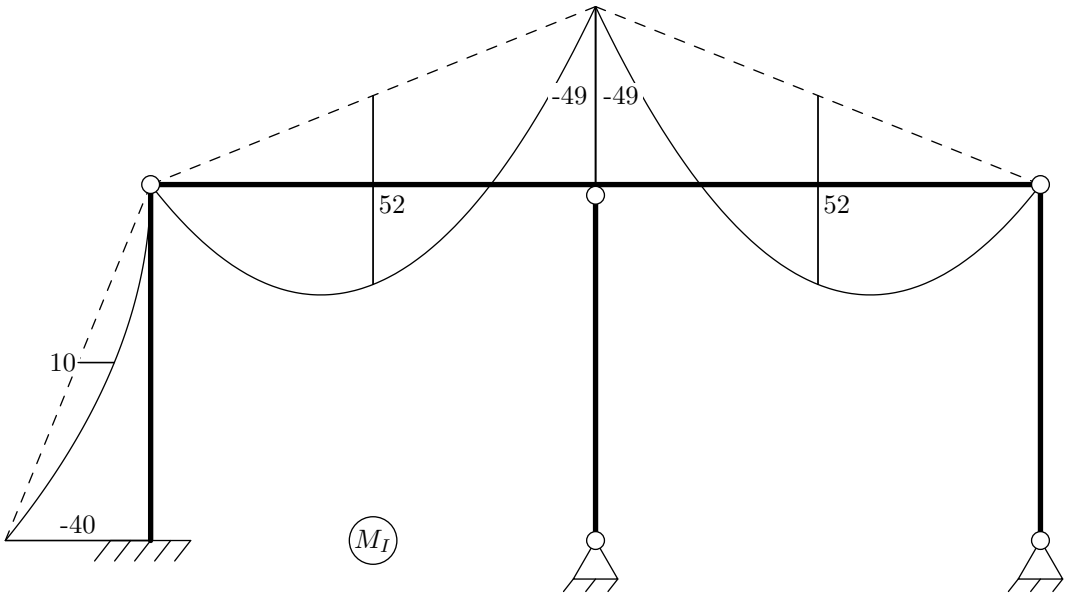
3.2 Verbundträger

```
\begin{asy}
   include konstruktiv;
   size(8cm);
   defaultpen(fontsize(10pt));
   Vtrager("IPE300",51,149,700,true,80,true,20,80,40);
\end{asy}
```



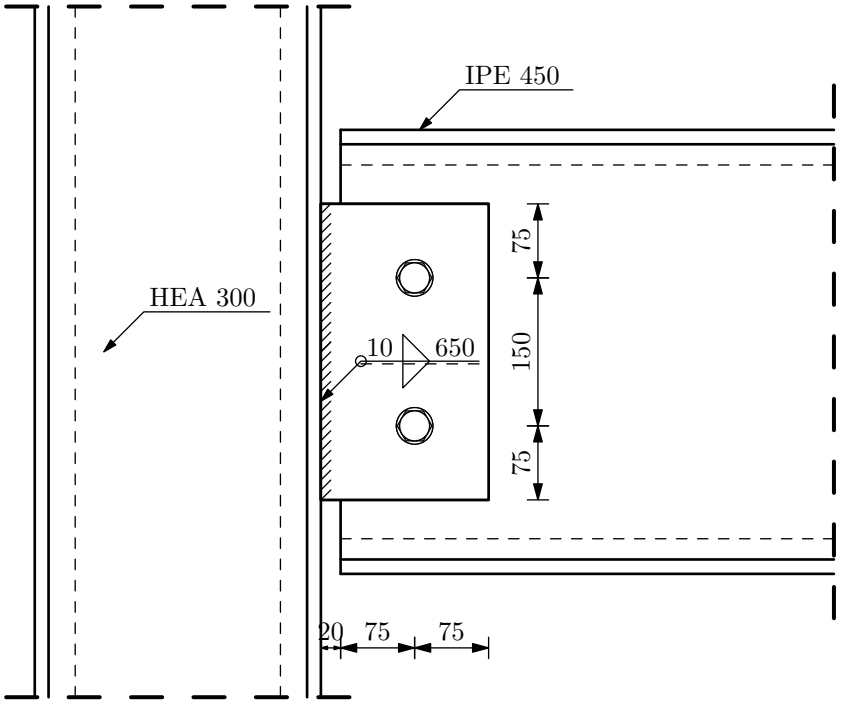
3.3 Momentenlinie

```
\begin{asy}
   include statik;
   size(10cm);
   defaultpen(fontsize(10pt));

   //Definition der Punkte
   pair a=(0,0);
   pair b=(0,4);
   pair c=(5,4);
   pair d=(5,0);
   pair e=(10,4);
   pair f=(10,0);

   //Momentenlinie
   verlauf(a,b,-40,10,0,true,true,10,49,W*2.5);
   verlauf(b,c,0,52,-49,true,true,10,49,SE);
   verlauf(c,e,-49,52,0,true,true,10,49,SE);
   balken(c-d,defaultpen+2);
   balken(e-f,defaultpen+2);
   gelenk(b);
   gelenk(c,(0,-0.12));
   gelenk(e);
   einspannung(a);
   lagerfest(d);
   lagerfest(f);

   //Beschriftung
   ustring((2.5,0),"$M_{I}$");
\end{asy}
```



3.4 Fahnenblech-Anschluss

```
\begin{asy}
   include konstruktiv;
   size(11cm);
   defaultpen(fontsize(10pt));

   //Punkte Definition
   pair p1=(-290/2,0);
   pair p2=(20,0);
   pair p3=(0,0);

   //TrÃďger
   Iansicht(p1,"HEA300",700,90);
   Iansicht(p2,"IPE450",500,true,false,true,"E");

   //Fahnenblech
   rechteck(p3,170,300,"white","E",defaultpen+1);

   //SchweiÃ§naht
   nahtd((p3),140,90,10);
   nahtd((p3),150,270,10);

   //Schrauben
   schraubend((95,75),"M20");
   schraubend((95,-75),"M20");

   //BemaÃ§ung
   real[] l1={75,150,75};
   vbemassung(l1,(0,-150),220);
   real[] l2={20,75,75};
   hbemassung(l2,(0,-150),-150);

   //Beschriftung
   nahtbeschr((0,-50),10,650,"Doppelkehl",true);
   flinie((100,450/2),"IPE 450");
   flinie((-220,0),"HEA 300");
\end{asy}
```


IPE 450
HEA 300
10
650
75
150
75
20
75
75