TECHNISCHE UNIVERSITÄT
IN DER KULTURHAUPTSTADT EUROPAS
CHEMNITZ

MTH

Leichtgewichtige prototypische Mechanismensimulation im Web-Kontext

Lightweight prototypal mechanism simulation on the web

M. Sc. Pascal Schnabel

Prof. Stefan Gössner

Professur Montage- und Handhabungstechnik
Prof. Dr.-Ing. Maik Berger

Email: mht@mb.tu-chemnitz.de
Tel: +49 (0) 371 531 – 32841
Fax: +49 (0) 371 531 – 23739

www.tu-chemnitz.de/mb/mht

Technische Universität Chemnitz
Fakultät für Maschinenbau
Professur Montage- und Handhabungstechnik
Reichenhainer Straße 70
D-09126 Chemnitz

Agenda

1) Einleitung

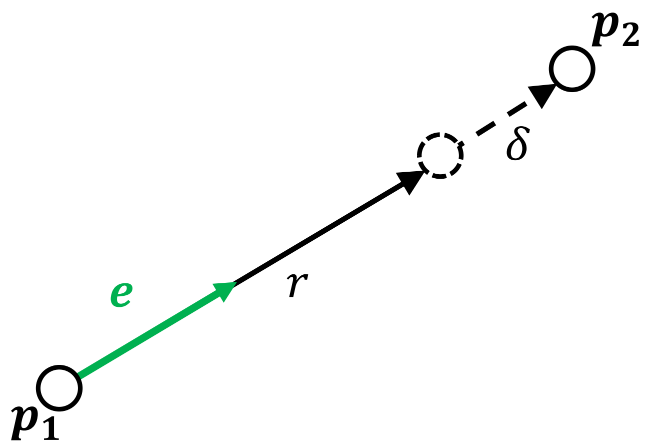
- Leichtgewichtige Modellierung ebener Mechanismen als Partikelsystem

  → *Nodes + Constraints*
- Ansatz in webbasierter Umgebung implementiert

  → Einfache Implementierung

- Mechanismenmodellierung unter Verzicht auf kinematische Größen

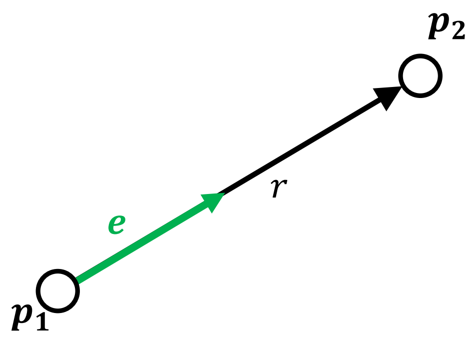
  → Beschränkt auf Lagenkorrektur



- Vektoren **fett** geschrieben

2) Modellierung mit *Nodes* und *Constraints*

- Zweidimensional, basierend auf Lagrange Formalismus
- *Nodes*
  - Position
  - Ausdehnungslos
  - Massebehaftet



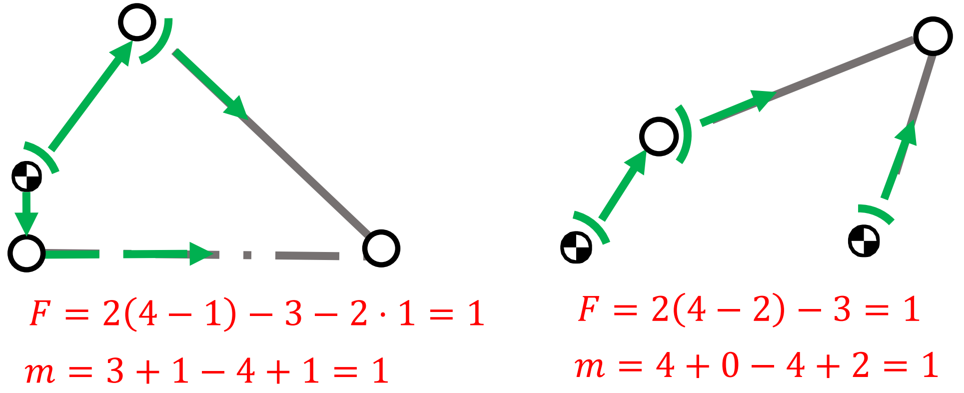
- Bindungen/ *Constraints*
  - Interpretierbar als polare Vektoren
  - Beschränken Beweglichkeit zwischen Nodes (Länge, Orientierung)
  - null- bis zweiwertige Bindung

- Gesamtfreiheitsgrad im $\mathbb{R}^2$

$$F = 2(N - N_{base}) - C_1 - 2C_2 \qquad (1)$$

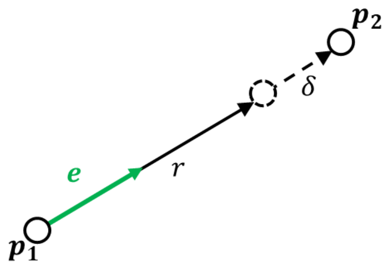
- Maschenanzahl:

$$m = C_1 + C_2 - N + N_{base} \qquad (2)$$



3) Herleitung

- Vektorielle Bindungsgleichung zwischen zwei Nodes



$$\boldsymbol{C}(\boldsymbol{q}) \equiv (\boldsymbol{p}_2 - \boldsymbol{p}_1) - r(t) \cdot \boldsymbol{e} = \boldsymbol{0} \qquad \boldsymbol{q} = \begin{pmatrix} \boldsymbol{p_1} \\ \boldsymbol{p_2} \end{pmatrix} \tag{3}$$

- Ableitung nach der Zeit mit Jacobimatrix $\boldsymbol{J} = \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{q}}$ und zeitabhängigem Anteil C_t

$$\dot{\boldsymbol{C}}(\boldsymbol{q}) = \boldsymbol{J}\dot{\boldsymbol{q}} + \boldsymbol{C_t} = \boldsymbol{0} \tag{4}$$

- Dynamische Grundgesetz mit eingeprägten Kräften $\boldsymbol{Q}$, Bindungskräften $\boldsymbol{J^T \lambda}$ und Lagrange Multiplikatoren $\boldsymbol{\lambda}$

$$\boldsymbol{M}\,\ddot{\boldsymbol{q}} = \boldsymbol{J}^T \boldsymbol{\lambda} + \boldsymbol{Q} \tag{5}$$

- Geschwindigkeitsbilanz nicht erfüllt => Einführung Korrekturvektors $\delta\dot{\boldsymbol{q}}$**:**

$$\dot{\boldsymbol{C}} = \boldsymbol{J}(\dot{\boldsymbol{q}} + \delta\dot{\boldsymbol{q}}) + \boldsymbol{C_t} = \boldsymbol{0} \tag{6}$$

- Vernachlässigung der eingeprägten Kräfte $\boldsymbol{Q} = 0$:

$$\boldsymbol{M}\frac{\delta\dot{\boldsymbol{q}}}{\delta t} = \boldsymbol{J}^T \boldsymbol{\lambda} \tag{7}$$

- Multiplikation mit δt und inverser Massenmatrix $\boldsymbol{W} = \boldsymbol{M^{-1}}$

$$\delta\dot{\boldsymbol{q}} = \boldsymbol{W}\,\boldsymbol{J^T}\boldsymbol{\lambda}\,\delta t \tag{8}$$

3) Herleitung

- Linksseitige Multiplikation mit Jacobimatrix $\boldsymbol{J}$

$$\boldsymbol{J}\delta\dot{\boldsymbol{q}} = \boldsymbol{J}\boldsymbol{W}\,\boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{\lambda}\,\delta t$$

- Umstellen der Gleichung (6): $\dot{\boldsymbol{C}} = \boldsymbol{J}(\dot{\boldsymbol{q}} + \delta\dot{\boldsymbol{q}}) + \boldsymbol{C_t} = \boldsymbol{0}$

$$\boldsymbol{J\delta\dot{q}} = -(\boldsymbol{J\dot{q}} + \boldsymbol{C_t})$$

- Multiplikation mit $\boldsymbol{m_C} = \left(\boldsymbol{J}\,\boldsymbol{W}\,\boldsymbol{J^T}\right)^{\boldsymbol{-1}}$ und einsetzen von $\boldsymbol{J\delta\dot{q}}$ ergibt korrektiven Impulsvektor $\boldsymbol{\lambda\delta t}$
  $\boldsymbol{m_C}$ als Masse der Bindung interpretierbar

$$\boldsymbol{\lambda\delta t} = -\left(\boldsymbol{JWJ^T}\right)^{\boldsymbol{-1}}(\boldsymbol{J\dot{q}} + \boldsymbol{C_t}) = -\boldsymbol{m_C}\cdot\dot{\boldsymbol{C}}(\dot{\boldsymbol{q}}) \qquad (9)$$

- Geschwindigkeitskorrektur $\boldsymbol{\delta\dot{q}}$:

$$\delta\dot{\boldsymbol{q}} = \boldsymbol{W}\,\boldsymbol{J^T}\boldsymbol{\lambda}\delta t = -\boldsymbol{WJ^T}\boldsymbol{m_C}\dot{\boldsymbol{C}}(\dot{\boldsymbol{q}}) \qquad (10)$$

- Integration

$$\boldsymbol{\delta q} = -\boldsymbol{WJ^T}\boldsymbol{m_C}\boldsymbol{C}(\boldsymbol{q}) \qquad (11)$$

3) Herleitung

Distanzerhaltende Bindung

- Einführung des Längen-Korrekturvektors δ

$$\mathbf{C}(\boldsymbol{q}+\delta) \equiv (\boldsymbol{p_2}-\boldsymbol{p_1})-(r+\delta)\boldsymbol{e}=\mathbf{0} \qquad (12)$$

- Multiplikation mit Einheitsvektor $\boldsymbol{e} = \begin{pmatrix} cos\ \varphi \\ sin\ \varphi \end{pmatrix}$

$$\delta = (\boldsymbol{p_2}-\boldsymbol{p_1})\boldsymbol{e}-r \qquad (\boldsymbol{e}\cdot\boldsymbol{e}=1)$$

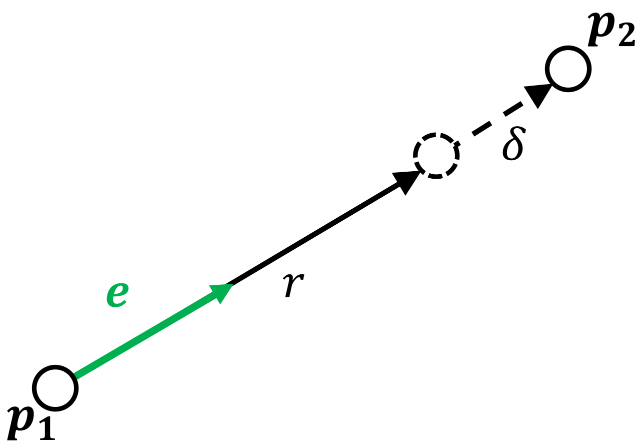
- Aufteilung gemäß *Node*massen

$$\delta = \delta_1+\delta_2 = \frac{m_C}{m_1}\delta + \frac{m_C}{m_2}\delta \qquad (13)$$

- Einsetzen in Bindungsgleichung

$$(\boldsymbol{p_2}-\boldsymbol{p_1})-(r+\delta_1+\delta_2)\boldsymbol{e} = (\boldsymbol{p_2}-\delta_2\boldsymbol{e})-(\boldsymbol{p_1}+\delta_1\boldsymbol{e})-r\,\boldsymbol{e}=\mathbf{0}$$

$$\boxed{\left(\boldsymbol{p_2}-\frac{m_C}{m_2}\delta\,\boldsymbol{e}\right)-\left(\boldsymbol{p_1}+\frac{m_C}{m_1}\delta\,\boldsymbol{e}\right)-r\,\boldsymbol{e}=\mathbf{0}} \qquad (14)$$



3) Herleitung

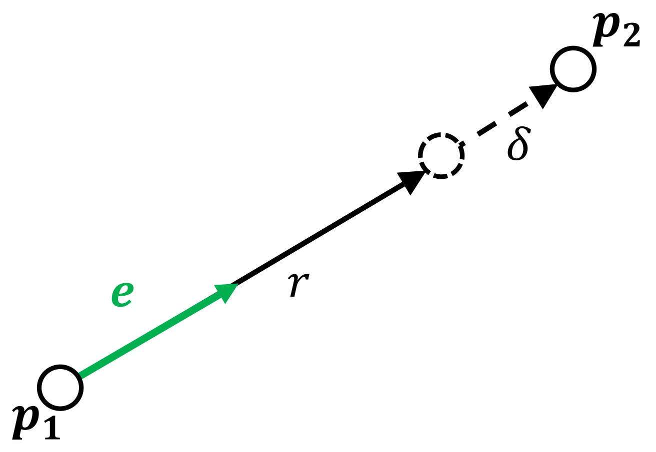
Distanzerhaltende Bindung – alternative Herleitung

- Skalare Bindungsgleichung

$$\boldsymbol{C}(\boldsymbol{q}) \equiv (\boldsymbol{p_2} - \boldsymbol{p_1})\boldsymbol{e} - (r) = \boldsymbol{0}$$

$$\mathbf{C}(\boldsymbol{q} + \delta) \equiv (\boldsymbol{p_2} - \boldsymbol{p_1})\boldsymbol{e} - (r + \delta) = \boldsymbol{0} \rightarrow \boldsymbol{C}(\boldsymbol{q}) = \boldsymbol{\delta}$$

$$\boldsymbol{q} = \begin{pmatrix} \boldsymbol{p_1} \\ \boldsymbol{p_2} \end{pmatrix}$$



- Ableitung nach den generalisierten Koordinaten ergibt

$$\boldsymbol{J} = \frac{\partial \boldsymbol{C}(\boldsymbol{q})}{\partial \boldsymbol{q}} = \begin{pmatrix} -\boldsymbol{e} & \boldsymbol{e} \end{pmatrix} \text{ mit } \qquad \boldsymbol{W} = \begin{pmatrix} 1/m_1 & 0 \\ 0 & 1/m_2 \end{pmatrix}$$

- Es resultiert die Bindungsmasse

$$m_C = \left(\boldsymbol{J}\,\boldsymbol{W}\,\boldsymbol{J}^T\right)^{-1} = \left[\begin{pmatrix} -\boldsymbol{e} & \boldsymbol{e} \end{pmatrix} \begin{pmatrix} \frac{1}{m_1} & 0 \\ 0 & \frac{1}{m_2} \end{pmatrix} \begin{pmatrix} -\boldsymbol{e} \\ \boldsymbol{e} \end{pmatrix}\right]^{-1} = \frac{1}{\frac{1}{m_1} + \frac{1}{m_2}}$$

- Einsetzen in (11)

$$\boxed{\delta \boldsymbol{q} = -\begin{pmatrix} \frac{1}{m_1} & 0 \\ 0 & \frac{1}{m_2} \end{pmatrix} \begin{pmatrix} -\boldsymbol{e} \\ \boldsymbol{e} \end{pmatrix} m_C \boldsymbol{C} = -\begin{pmatrix} \frac{1}{m_1} & 0 \\ 0 & \frac{1}{m_2} \end{pmatrix} \begin{pmatrix} -\boldsymbol{e} \\ \boldsymbol{e} \end{pmatrix} m_C \delta = \begin{pmatrix} \frac{m_C}{m_1} \delta \boldsymbol{e} \\ -\frac{m_C}{m_2} \delta \boldsymbol{e} \end{pmatrix}}$$

3) Herleitung

Orientierungserhaltende Bindung

- Winkelkorrektur über eine zur Zielrichtung $\boldsymbol{e}$ orthogonale Komponente $\tilde{\boldsymbol{e}}$

$$\mathbf{C}(\boldsymbol{q}+\delta) \equiv (\boldsymbol{p_2}-\boldsymbol{p_1})-(\rho\boldsymbol{e}+\delta\tilde{\boldsymbol{e}})=\mathbf{0} \qquad (15)$$

- Multiplikation mit $\tilde{\boldsymbol{e}}$

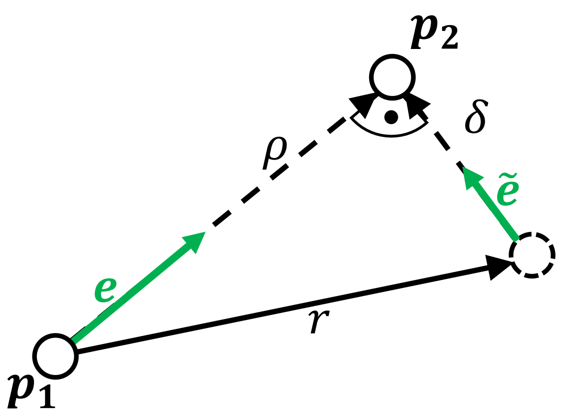
$$\delta=(\boldsymbol{p_2}-\boldsymbol{p_1})\tilde{\boldsymbol{e}} \qquad (\boldsymbol{e}\cdot\tilde{\boldsymbol{e}}=0)$$

- Aufteilung gemäß *Node*massen

$$\delta=\delta_1+\delta_2=\frac{m_C}{m_1}\delta+\frac{m_C}{m_2}\delta \qquad (13)$$

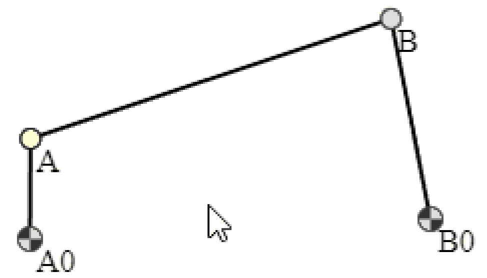
- Einsetzen in Bindungsgleichung

$$\boxed{\left(\boldsymbol{p_2}-\frac{m_C}{m_2}\delta\,\tilde{\boldsymbol{e}}\right)-\left(\boldsymbol{p_1}+\frac{m_C}{m_1}\delta\tilde{\boldsymbol{e}}\right)-\rho\,\boldsymbol{e}=\mathbf{0}} \qquad (16)$$



4) Implementierung

- HTML, CSS, Javascript
- Bibliotheken: *g2.js; cstr.js*



1) HTML–Dokumentkopf: <head>
2) <body> des HTML-Dokumentes
   - Zeichenfläche: <canvas>
   - Verweis auf Bibliotheken
   - Programmlogik: <script>

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Titel des Dokuments</title>
</head>
```

```
<body>
    <canvas id="c" width="900" height="800"></canvas><!--Zeichenfläche-->
    <script src="./g2.js"></script>    <!--Verweis auf Grafikbibliothek-->
    <script src="./cstr.js"></script>    <!--Verweis auf Constraintbibliothek-->
```

4) Implementierung

- Zugriff auf <canvas>
- Interaktionsobjekt
- Selektionsobjekt

```
<script>
        //Zugriff auf canvas-Element
        const ctx = document.getElementById('c').getContext('2d');
        //Interaktionsobjekt erstellen
        const interactor = canvasInteractor.create(ctx, { x: 140, y: 140, cartesian: true });
        // Selektionsobjekt
        const selector = g2.selectorHdl(interactor.evt);
```

4) Implementierung

- *Nodes*

Attribut	Bedeutung	Verwendung
x,y	x- und y- Koordinate	*x: 100, y: 50*
m	Masse	*m: 1*
base	Gestellfest? => m: ∞	*base: true*
label	Beschriftung	*label: „A0“* *label: „A\u2080“*
r	Radius	*r:10*

```
//Definition der Nodes
const A0 = { x: 0, y: 0, base: true, label: 'A0' },
    A = { x: 0, y: 50, m: 2, label: 'A' },
    B = { x: 180, y: 110, label: 'B' },
    B0 = { x: 200, y: 10, base: true, label: 'B0' };
```

- *Constraints*
  - *n2(), n3()*

Attribut	Bedeutung	Verwendung
n1,n2,n3	Node 1,2,3	*n1: A, n2:B*
len	Länge des Constraints	*len: „const“* *len: 10*
ang	Winkel des Constraints	*ang: „const“,* *ang: 3.14*
id	„ID“ => ermöglicht spätere Referenzierung	*id: „AB“*

```
//Definition der Constraints
const c = cstr().n2({ id: 'A0A', n1: A0, n2: A, len: 'const' })
    .n2({ n1: A, n2: B, len: 'const' })
    .n2({ n1: B0, n2: B, len: 'const' })
```

4) Implementierung

- Zeichnen des Getriebes => *g2.js* – Bibliothek
- Funktionen:
  - *lin(), ply(), nod(), gnd(), …*

```
//Zeichnen des Getriebe
       const g = g2().clr()
           .view(interactor.view)
           .ply({ pts: [A0, A, B,B0], lw: 2}) //Polylinie
           .gnd(A0) //Symbol für gestellfesten Node
           .gnd(B0)
           .hdl(A) //"Handle" - Symbol
           .nod(B)//"Node" - Symbol
       g.exe(ctx);
```

ausführliche Dokumentation: https://goessner.github.io/g2/

4) Implementierung

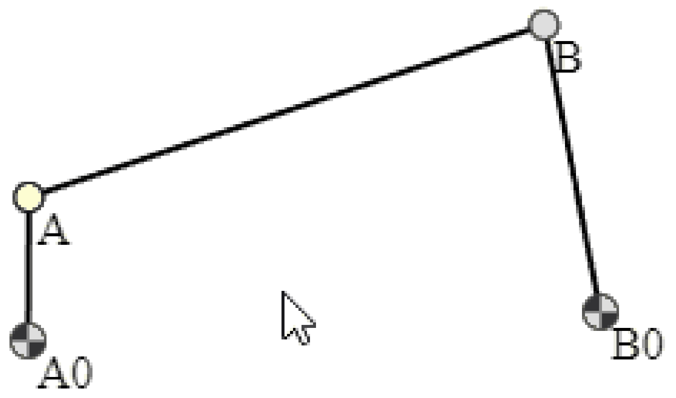
- Interaktions- und Animationsfunktionen => *canvasInteractor*
- Events*: tick, pan, drag, …*
- *startTimer()*

```
        //Definition der Interaktions- und Animationsmethoden
        interactor
            .on('tick', (e) => {
                c.correct();
                g.exe(selector).exe(ctx);
            })
            .on('drag', (e) => {
                if (selector.selection && selector.selection.drag) {
                    selector.selection.drag({x:e.xusr,y:e.yusr,
                         dx:e.dxusr,dy:e.dyusr,mode:'drag' });
                }
            })
            .startTimer();
    </script>
</body>
</html>
```

Weitere Informationen: https://goessner.github.io/canvasInteractor/

4) Implementierung

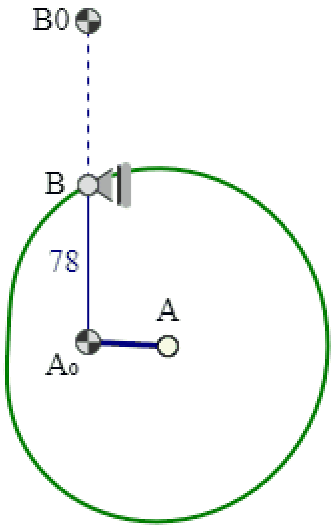
- Vorteile
  - Einfache Programmierung
  - Geringe Dateigröße:
    - *.js*: 104*kb*
    - *.ggb: 54.5Mb*
  - Keine zusätzliche Software
  - Endlose Gestaltungsmöglichkeiten
  - erweiterbar

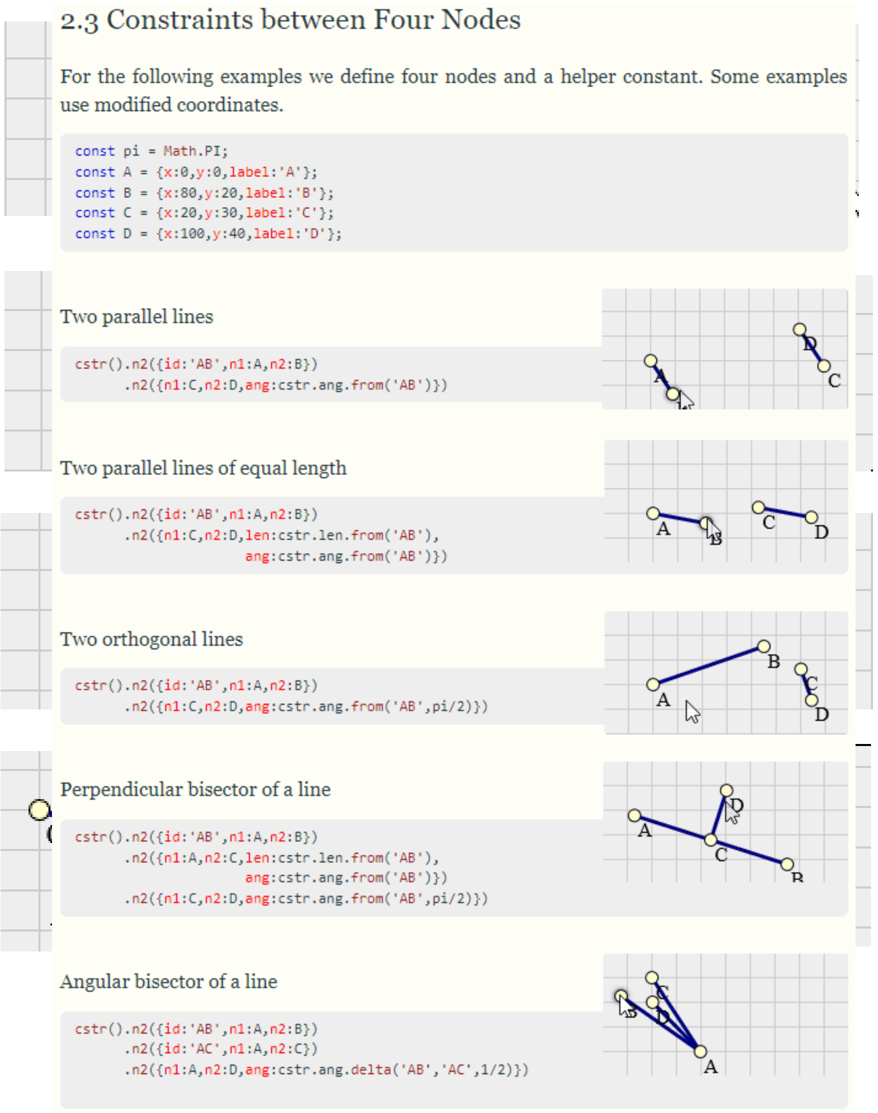


4) Implementierung

Sonstiges

- Zusätzliche Bindungen
  - Längen-/ Winkelbeschränkungen: `len: cstr.len.range(1,10)`
  - Mittelpunkt zweier Nodes, Kollinearität…
  - Abhängigkeiten zu anderen Bindungen, Schnittpunkte, …
  - Funktionen





2.3 Constraints between Four Nodes

For the following examples we define four nodes and a helper constant. Some examples use modified coordinates.

```
const pi = Math.PI;
const A = {x:0,y:0,label:'A'};
const B = {x:80,y:20,label:'B'};
const C = {x:20,y:30,label:'C'};
const D = {x:100,y:40,label:'D'};
```

Two parallel lines

```
cstr().n2({id:'AB',n1:A,n2:B})
      .n2({n1:C,n2:D,ang:cstr.ang.from('AB')})
```

Two parallel lines of equal length

```
cstr().n2({id:'AB',n1:A,n2:B})
      .n2({n1:C,n2:D,len:cstr.len.from('AB'),
                     ang:cstr.ang.from('AB')})
```

Two orthogonal lines

```
cstr().n2({id:'AB',n1:A,n2:B})
      .n2({n1:C,n2:D,ang:cstr.ang.from('AB',pi/2)})
```

Perpendicular bisector of a line

```
cstr().n2({id:'AB',n1:A,n2:B})
      .n2({n1:A,n2:C,len:cstr.len.from('AB'),
                     ang:cstr.ang.from('AB')})
      .n2({n1:C,n2:D,ang:cstr.ang.from('AB',pi/2)})
```

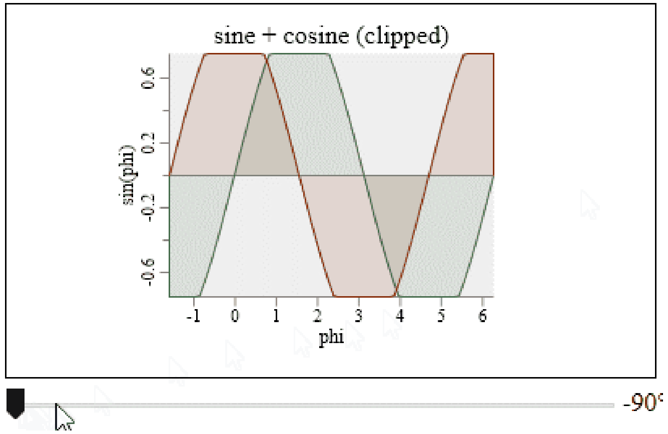
Angular bisector of a line

```
cstr().n2({id:'AB',n1:A,n2:B})
      .n2({id:'AC',n1:A,n2:C})
      .n2({n1:A,n2:D,ang:cstr.ang.delta('AB','AC',1/2)})
```

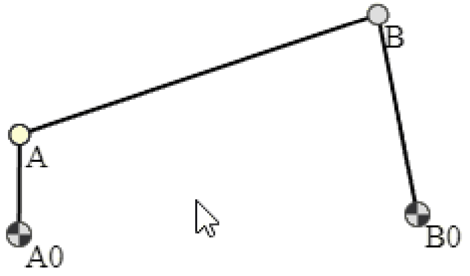
4) Implementierung

Sonstiges

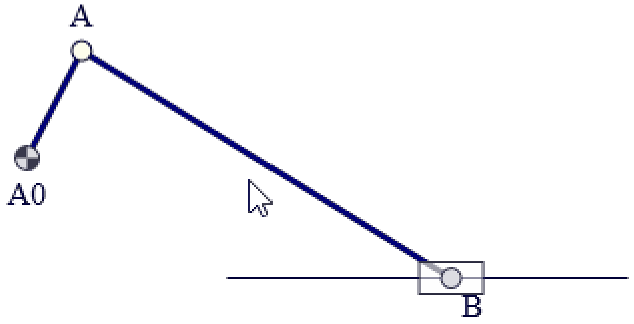
- Konstanten:
  - Standardmäßige Längentoleranz: `cstr.lentol=0.1`
  - Standardmäßige maximale Iterationszahl: `cstr.itrmax=256`
- Berechnung des Freiheitsgrades:
  - `cstr().dof`
- Grafikbibliothekerweiterung g2-Chart
  - Dokumentation: https://goessner.github.io/g2/microjam.md/g2.chart.html



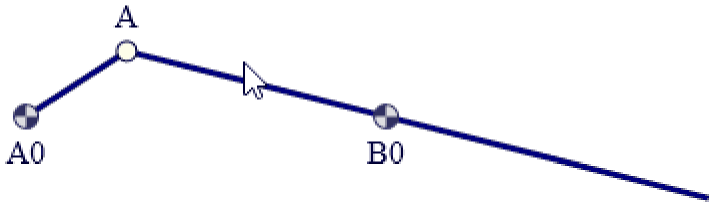
5) Beispiele



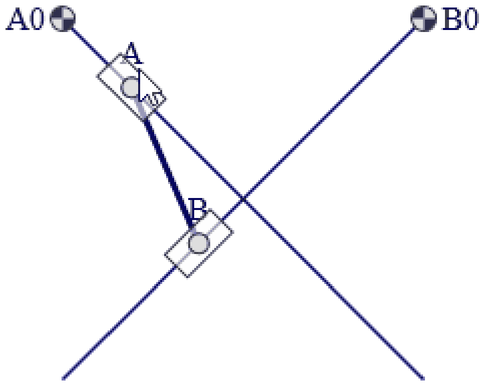
Kurbelschwinge



Schubkurbel



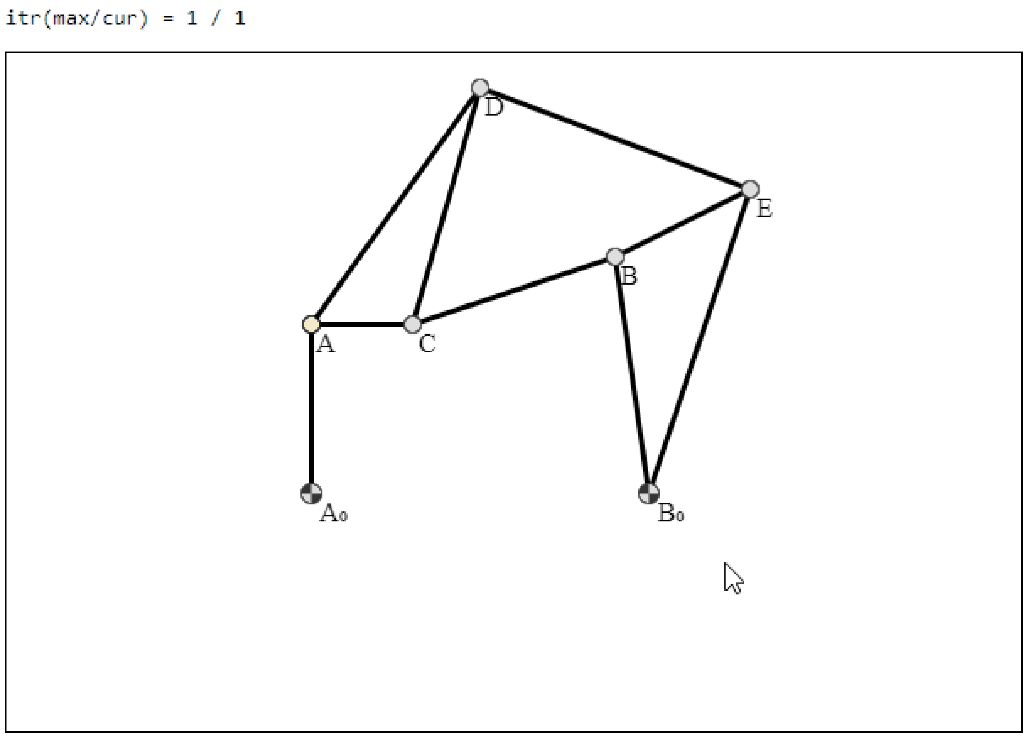
Schubschleife



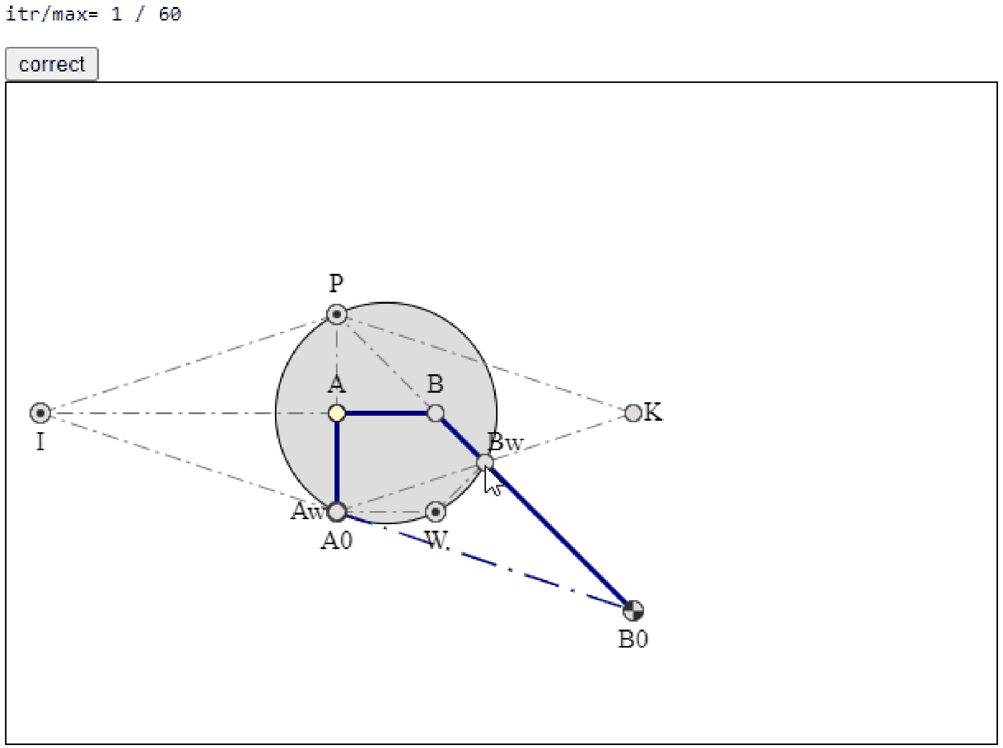
Kreuzschubkurbel

5) Beispiele

- Stephenson II Mechanismus
  - keine analytische Lösung

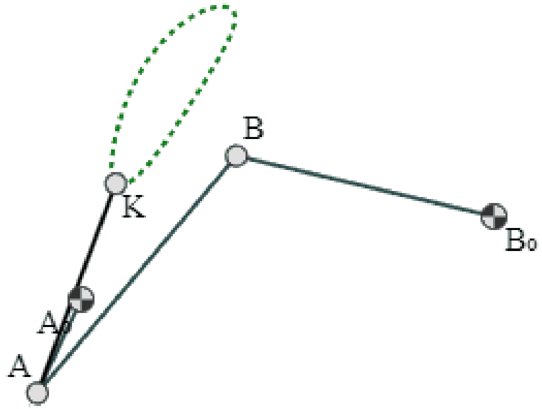


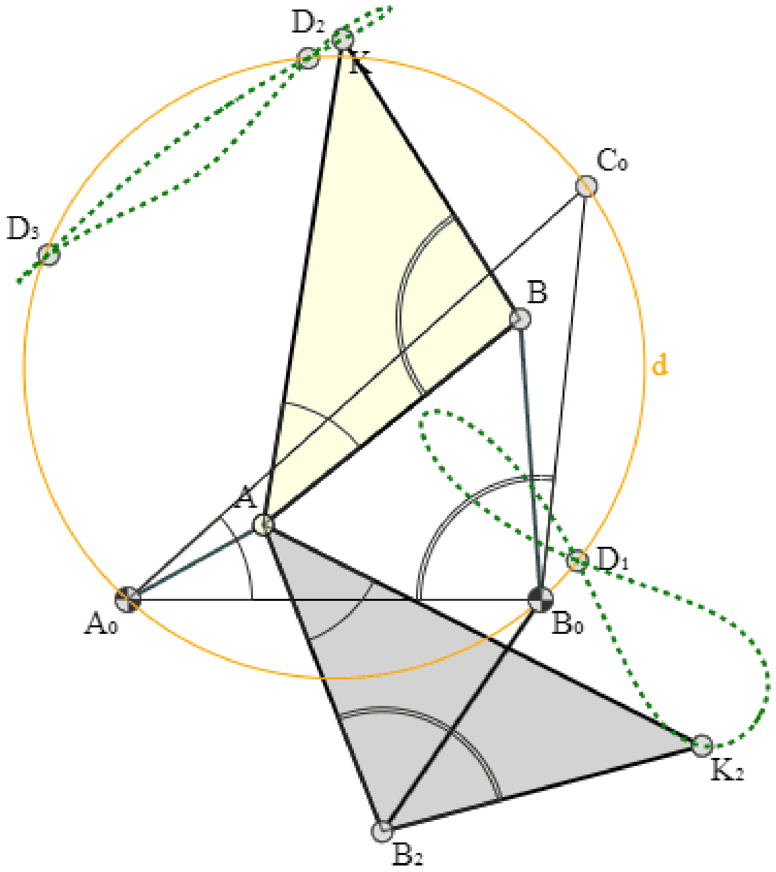
- Satz von Bobillier



5) Beispiele

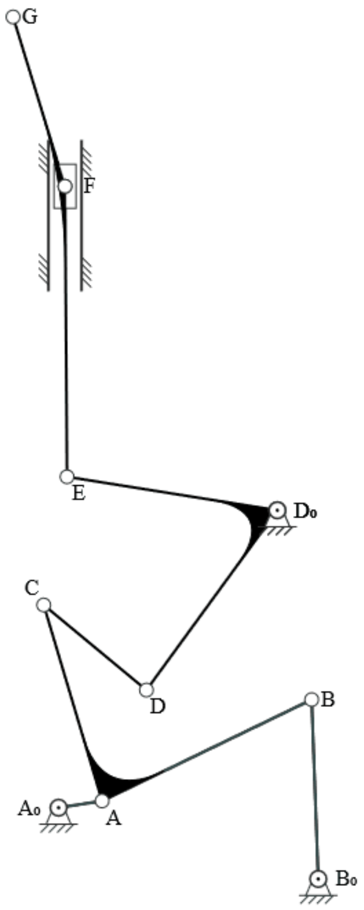
Koppelkurven





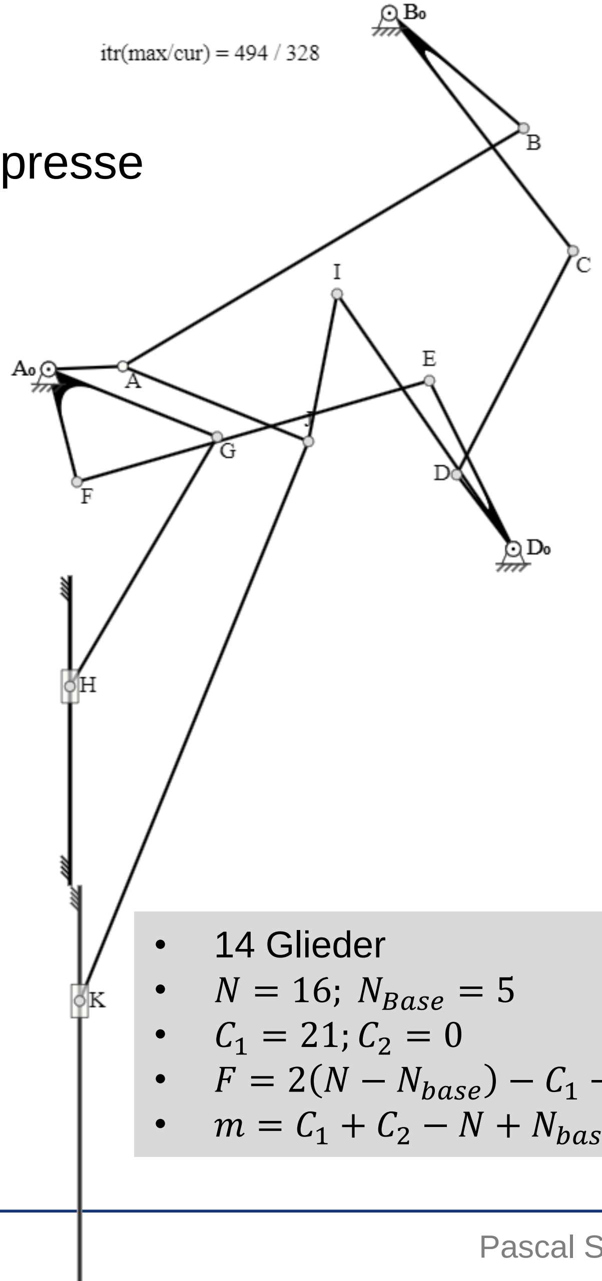
5) Beispiele

- Nadelbarrenantrieb einer Textilmaschine



- 8 Glieder
- $N = 10\,;\ N_{Base} = 4$ (G nicht beachtet)
- $C_1 = 11; C_2 = 0$
- $F = 2(N - N_{base}) - C_1 - 2C_2 = 1$
- $m = C_1 + C_2 - N + N_{base} = 5$

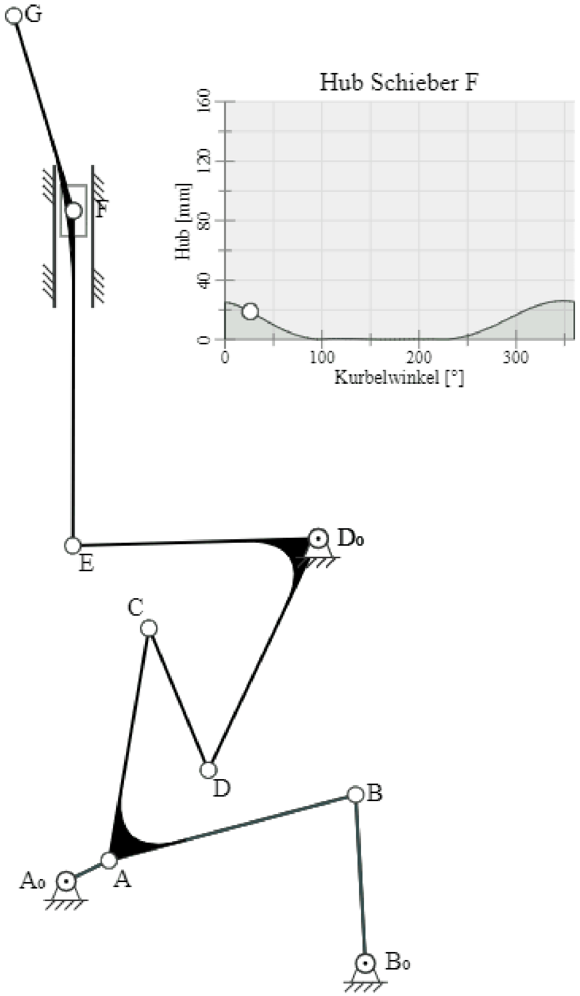
- Tiefziehkurbelpresse



- 14 Glieder
- $N = 16;\ N_{Base} = 5$
- $C_1 = 21; C_2 = 0$
- $F = 2(N - N_{base}) - C_1 - 2 * C_2 = 1$
- $m = C_1 + C_2 - N + N_{base} = 10$

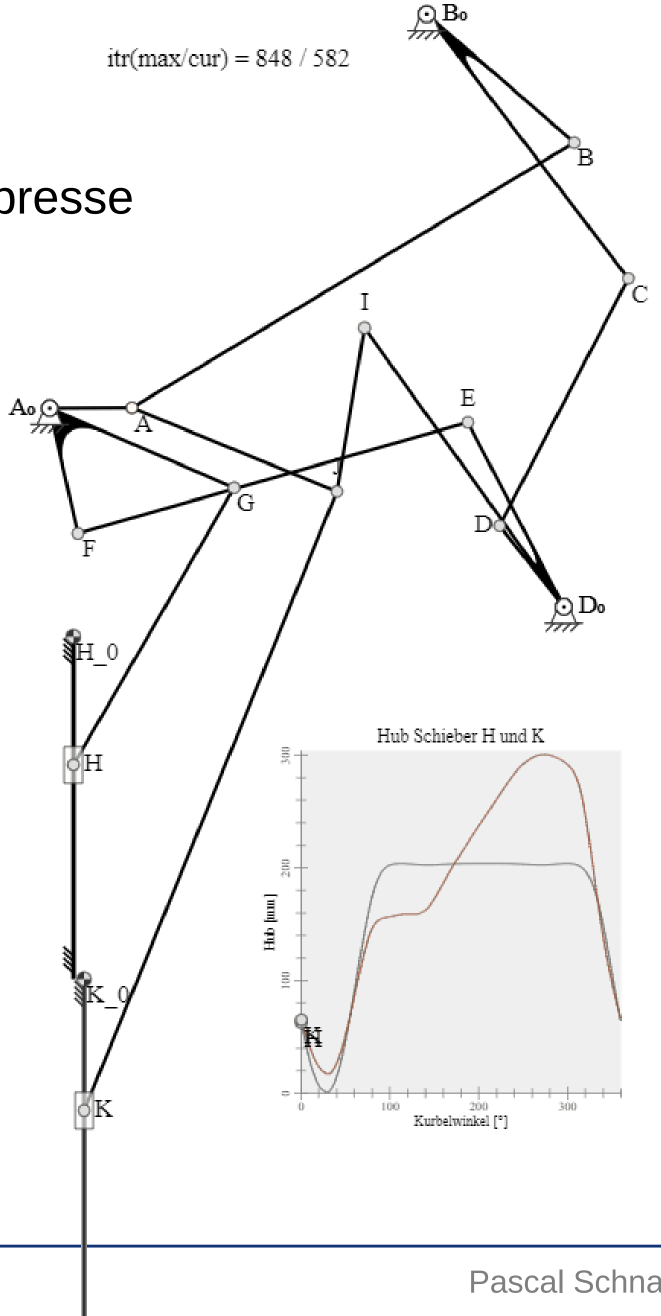
5) Beispiele

- Nadelbarrenantrieb einer Textilmaschine



- 8 Glieder
- $N = 10\,;\ N_{Base} = 4$ (G nicht beachtet)
- $C_1 = 11; C_2 = 0$
- $F = 2(N - N_{base}) - C_1 - 2C_2 = 1$
- $m = C_1 + C_2 - N + N_{base} = 5$

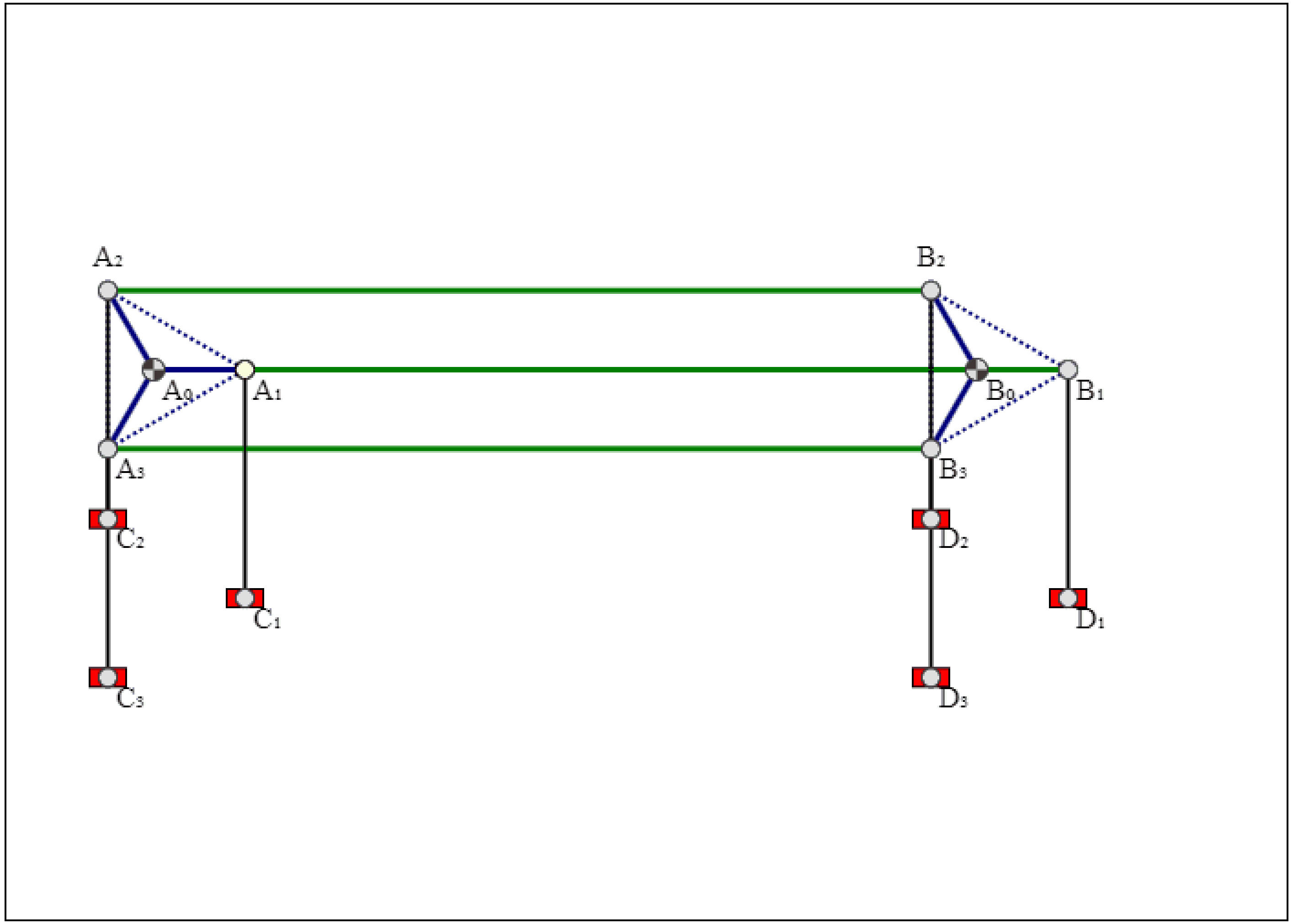
- Tiefziehkurbelpresse



5) Beispiele

Schreitkrabbler

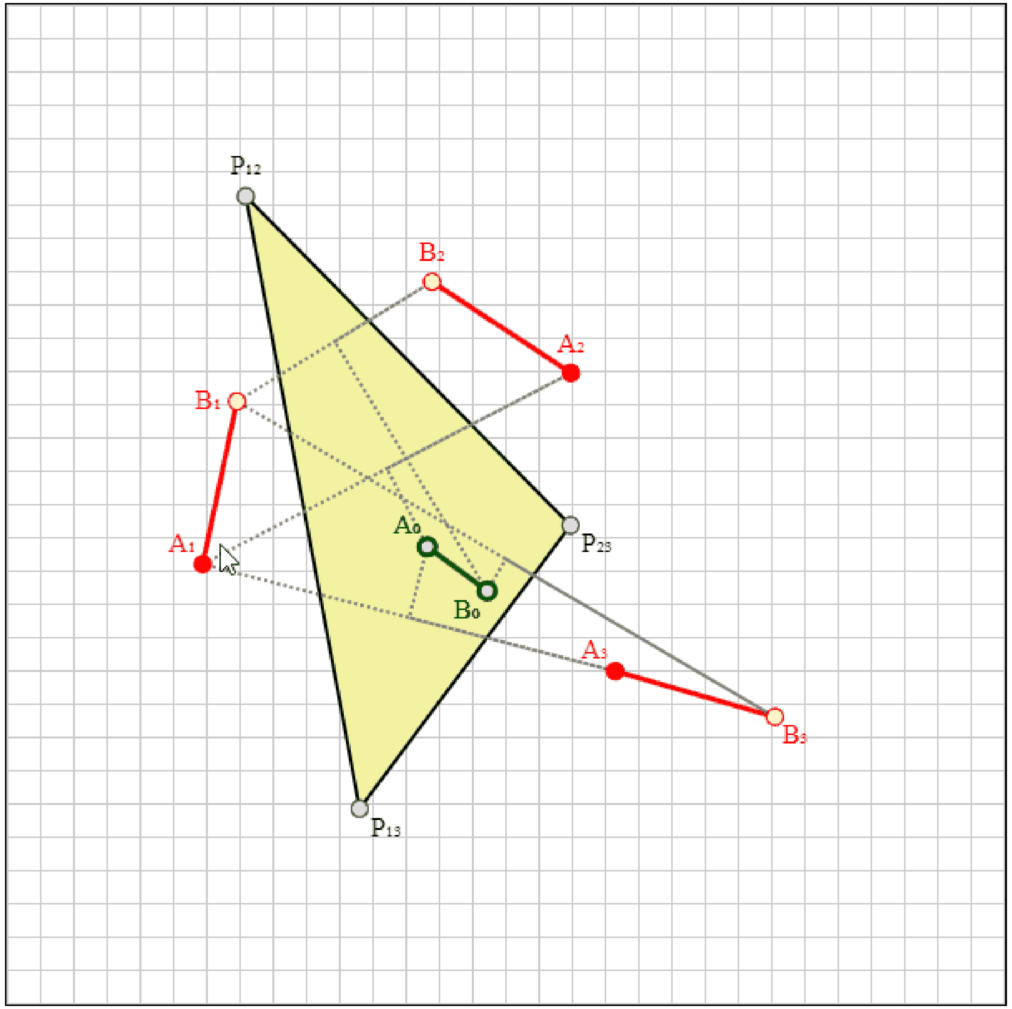
itr(max/cur) = 1 / 1 Dof:1



Geg.:	$N = 14;\ N_{base} = 2; C_1 = 1; C_2 = 11$
Freiheitsgrad:	$F = 2(N - N_{base}) - C_1 - 2C_2 = 1$
Maschenanzahl:	$m = C_1 + C_2 - N + N_{base} = 0$

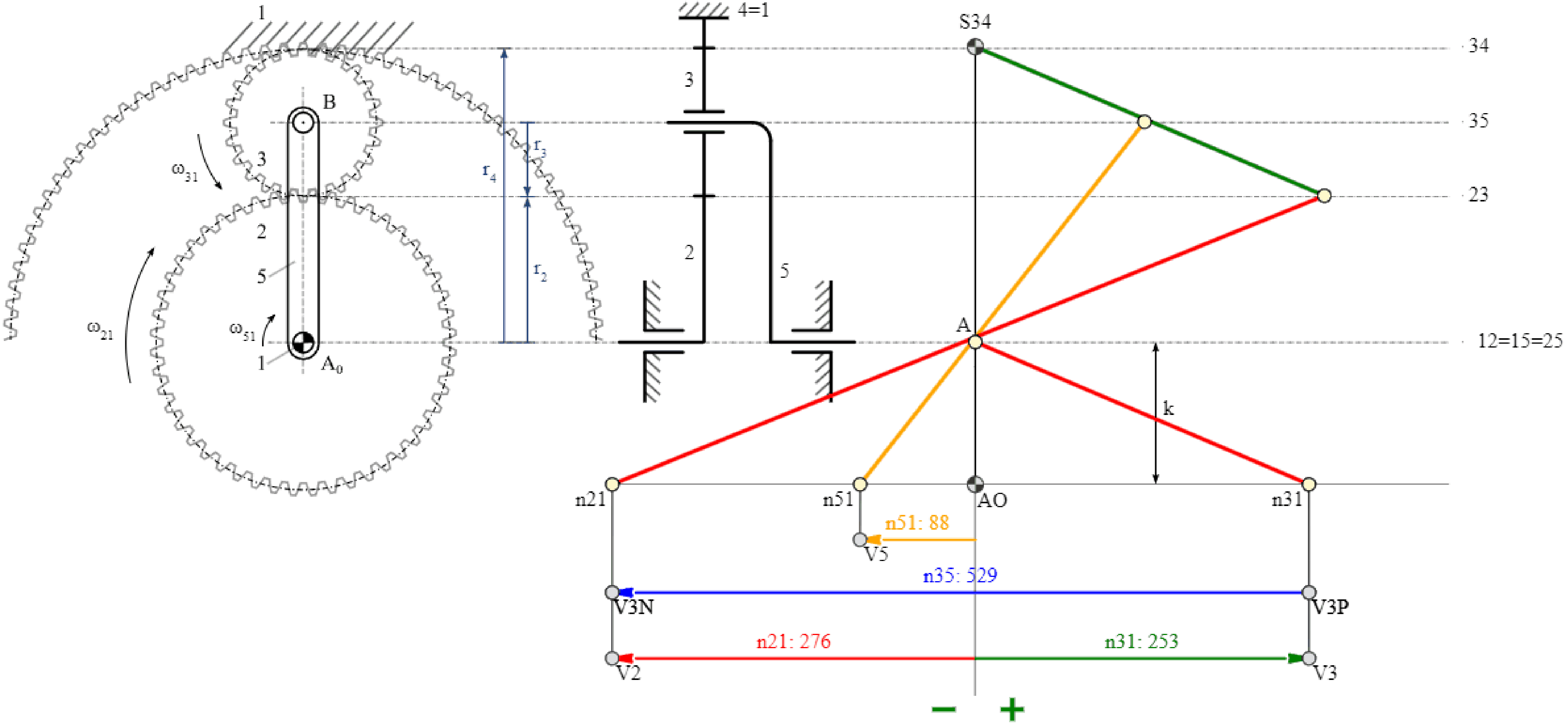
5) Beispiele

Synthese - Poldreieck



5) Beispiele

Drehzahlplan

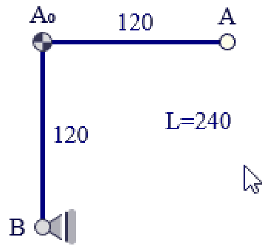


6) Fazit

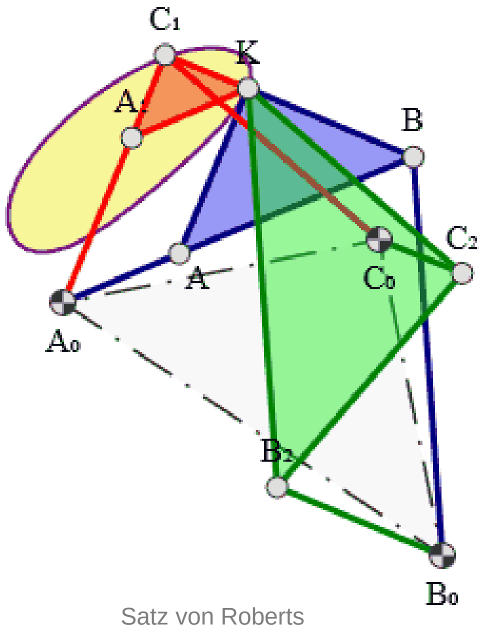
- Sehr einfache Anwendung
- Simulationsfehler, was nun?:
  - Reduktion der Schrittweite dt
  - alternative Modellierung
  - Steigerung der maximalen Iterationszahl `cstr.itrmax=1024`
  - Reduktion der Längentoleranz: `cstr.lentol=0.005`
  - (*n3()-Constraint* stabiler)

7) Ausblick

- Veröffentlichung der Dokumentation
- Seilbindungen
- Animation weiterer Lehrsätze



Seilbindung



Satz von Roberts

Beispiele:
https://github.com/Pasquale19/Beispiele_GT2022

8) Quellen

[1] Gössner, S., " Ebene Mechanismenmodelle als Partikelsysteme – ein neuer Ansatz ", 13. Kolloquium Getriebetechnik, Dortmund, September 2019.

[2] Gössner, S., "g2", URL: http://goessner.github.io/g2/.

[3] Gössner, S., "Make your HTML canvas Interactive", URL: https://goessner.github.io/canvasInteractor/.

[4] Gössner, S. Mechanismentechnik. Vektorielle Analyse ebener Mechanismen. Logos Verlag Berlin, 2017, ISBN 978-3-8325-4362-4.

[5] Heinrich, S., Berger, M.: Module based synthesis of a 14 bar deep drawing press considering kinetic criteria, ESI SIMULATIONX CONFERENCE. Messe Dresden, 8.-9. November 2018, Dresden, 2018.

[6] Heinrich, S.: Modulbasierte Synthese ebener Koppelgetriebe unter Einbeziehung kinetischer Kenngrößen, Technische Universität Chemnitz, Dissertation, 2018, ISBN: 978-3-96100-066-1

[7] Schnabel, P., “Beispiele Getriebetagung 2022”, URL: https://github.com/Pasquale19/Beispiele_GT2022.

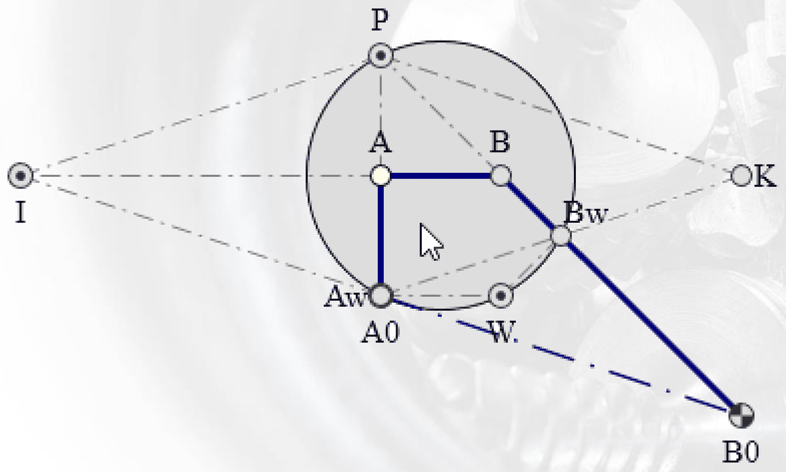
Professur Montage- und Handhabungstechnik
Prof. Dr.-Ing. Maik Berger

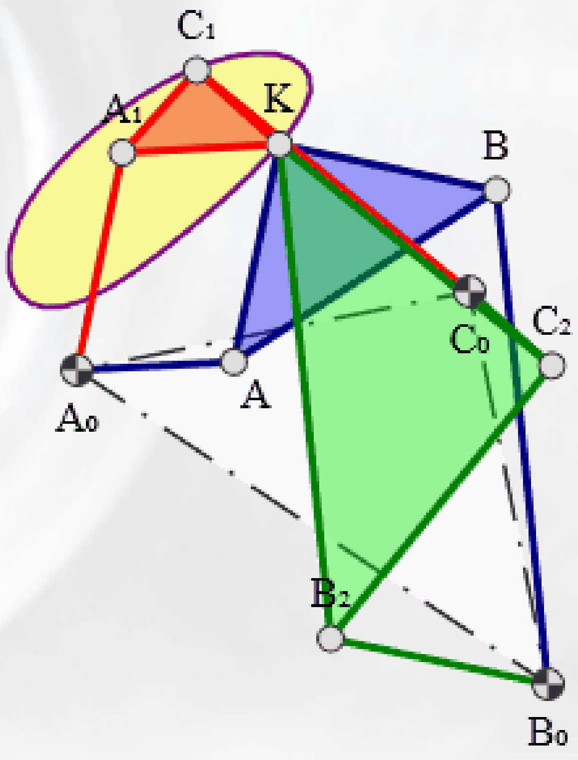
Email: mht@mb.tu-chemnitz.de
Tel: +49 (0) 371 531 – 32841
Fax: +49 (0) 371 531 – 23739

www.tu-chemnitz.de/mb/mht

Technische Universität Chemnitz
Fakultät für Maschinenbau
Professur Montage- und Handhabungstechnik
Reichenhainer Straße 70
D-09126 Chemnitz

Zertifikats-Reg.-Nr. 103325 QM08





Vielen Dank für Ihre Aufmerksamkeit

https://github.com/Pasquale19/Beispiele_GT2022