



TECHNISCHE UNIVERSITÄT
IN DER KULTURHAUPTSTADT EUROPAS
CHEMNITZ



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



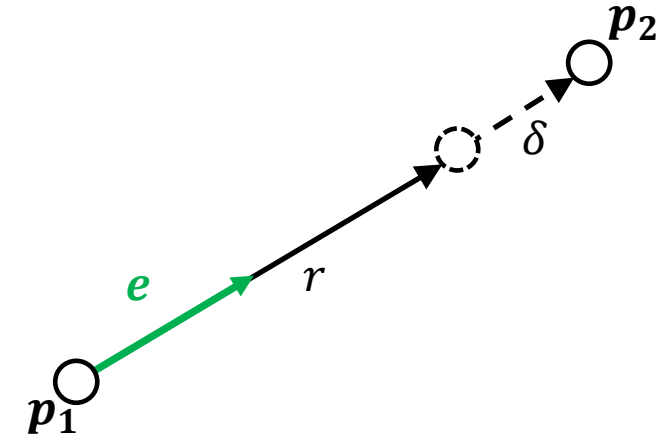
Zertifikats-Reg.-Nr. 103325 QM08

Agenda

1. Einleitung
2. Modellierung mit *Nodes* und *Constraints*
3. Herleitung
4. Implementierung
5. Beispiele
6. Fazit
7. Ausblick

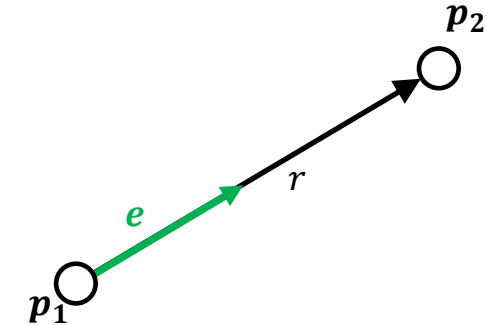
1) Einleitung

- Modellierung ebener Mechanismen als Partikelsystem
 - Ausdehnungslose, massebehaftete Punkte (*Nodes*)
 - Geometrische Bindungen (*Constraints*)
- Mechanismenmodellierung unter Verzicht auf kinematische Größen
 - Beschränkt auf Lagenkorrektur
- Ansatz in webbasierter Umgebung implementiert
 - Einfache Implementierung
 - Interaktive Bewegung einzelner Nodes
- Vektoren **fett** geschrieben



2) Modellierung mit *Nodes* und *Constraints*

- Zweidimensional, basierend auf Lagrange Formalismus
- *Nodes*
 - Ausdehnungslos
 - Massebehaftet
- Bindungen/ *Constraints*
 - Beschränken Beweglichkeit zwischen Nodes (Länge, Orientierung)
 - Interpretierbar als polare Vektoren
 - null- bis zweiwertige Bindung

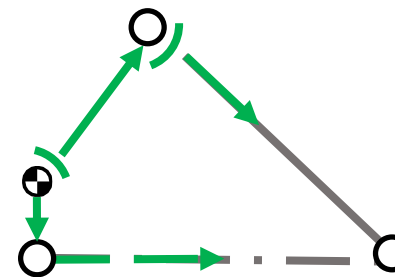


- Gesamtfreiheitsgrad im \mathbb{R}^2

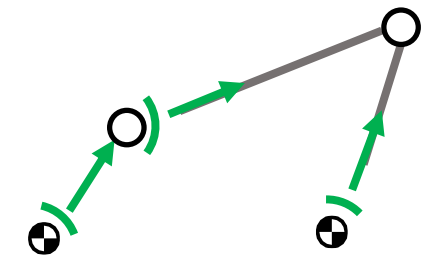
$$F = 2(N - N_{base}) - C_1 - 2C_2 \quad (1)$$

- Maschenanzahl:

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



$$\begin{aligned}
 F &= 2(4 - 1) - 3 - 2 \cdot 1 = 1 \\
 m &= 3 + 1 - 4 + 1 = 1
 \end{aligned}$$



$$\begin{aligned}
 F &= 2(4 - 2) - 3 = 1 \\
 m &= 4 + 0 - 4 + 2 = 1
 \end{aligned}$$

3) Herleitung

- Vektorielle Bindungsgleichung zwischen zwei Nodes

$$\mathbf{C}(\mathbf{q}) \equiv (\mathbf{p}_2 - \mathbf{p}_1) - r(t) \cdot \mathbf{e} = \mathbf{0} \quad \mathbf{q} = \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{pmatrix} \quad (3)$$

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

$$\dot{\mathbf{C}}(\mathbf{q}) = \mathbf{J}\dot{\mathbf{q}} + \mathbf{C}_t = \mathbf{0} \quad (4)$$

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

$$\mathbf{M} \ddot{\mathbf{q}} = \mathbf{J}^T \boldsymbol{\lambda} + \mathbf{Q} \quad (5)$$

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

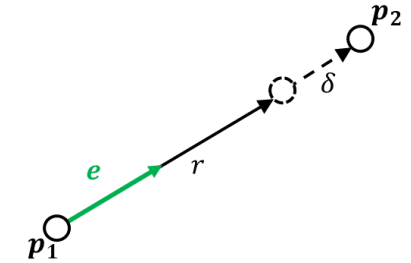
$$\dot{\mathbf{C}} = \mathbf{J}(\dot{\mathbf{q}} + \delta \dot{\mathbf{q}}) + \mathbf{C}_t = \mathbf{0} \quad (6)$$

- Vernachlässigung der eingepprägten Kräfte $\mathbf{Q} = 0$:

$$\mathbf{M} \frac{\delta \dot{\mathbf{q}}}{\delta t} = \mathbf{J}^T \boldsymbol{\lambda} \quad (7)$$

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

$$\delta \dot{\mathbf{q}} = \mathbf{W} \mathbf{J}^T \boldsymbol{\lambda} \delta t \quad (8)$$



3) Herleitung

- Linksseitige Multiplikation mit Jacobimatrix J

$$J\delta\dot{q} = JW J^T \lambda \delta t$$

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

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

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

$$\lambda\delta t = -(JWJ^T)^{-1}(J\dot{q} + C_t) = -m_c \cdot \dot{C}(\dot{q}) \quad (9)$$

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

$$\delta\dot{q} = W J^T \lambda\delta t = -W J^T m_c \dot{C}(\dot{q}) \quad (10)$$

- Integration

$$\delta q = -W J^T m_c C(q) \quad (11)$$

3) Herleitung

Distanzerhaltende Bindung

- Einführung des Längen-Korrekturvektors δ

$$(\mathbf{p}_2 - \mathbf{p}_1) - (r + \delta)\mathbf{e} = \mathbf{0} \quad (12)$$

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

$$\delta = (\mathbf{p}_2 - \mathbf{p}_1)\mathbf{e} - r \quad (\mathbf{e} \cdot \mathbf{e} = 1)$$

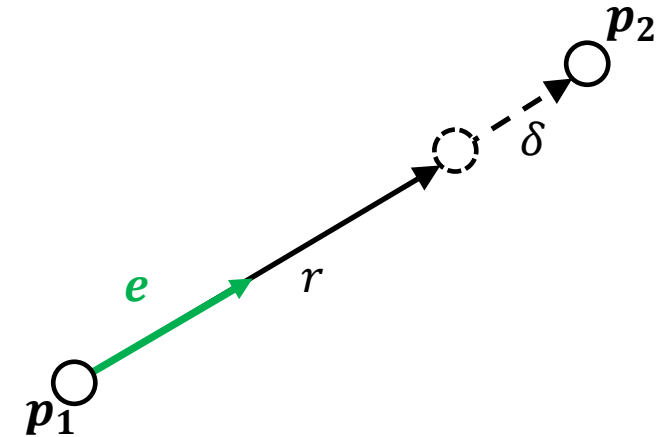
- Aufteilung gemäß Nodemassen

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

- Einsetzen in Bindungsgleichung

$$(\mathbf{p}_2 - \mathbf{p}_1) - (r + \delta_1 + \delta_2)\mathbf{e} = (\mathbf{p}_2 - \delta_2\mathbf{e}) - (\mathbf{p}_1 + \delta_1\mathbf{e}) - r\mathbf{e} = \mathbf{0}$$

$$\left[\left(\mathbf{p}_2 - \frac{m_C}{m_2} \delta \mathbf{e} \right) - \left(\mathbf{p}_1 + \frac{m_C}{m_1} \delta \mathbf{e} \right) - r \mathbf{e} = \mathbf{0} \right] \quad (14)$$



3) Herleitung

Distanzerhaltende Bindung – alternative Herleitung

- Skalare Bindungsgleichung

$$\mathbf{C}(\mathbf{q}) \equiv (\mathbf{p}_2 - \mathbf{p}_1)\mathbf{e} - (r) = \mathbf{0}$$

$$\mathbf{C}(\mathbf{q} + \delta) \equiv (\mathbf{p}_2 - \mathbf{p}_1)\mathbf{e} - (r + \delta) = \mathbf{0} \rightarrow \mathbf{C}(\mathbf{q}) = \delta$$

$$\mathbf{q} = \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{pmatrix}$$

- Ableitung nach den generalisierten Koordinaten ergibt

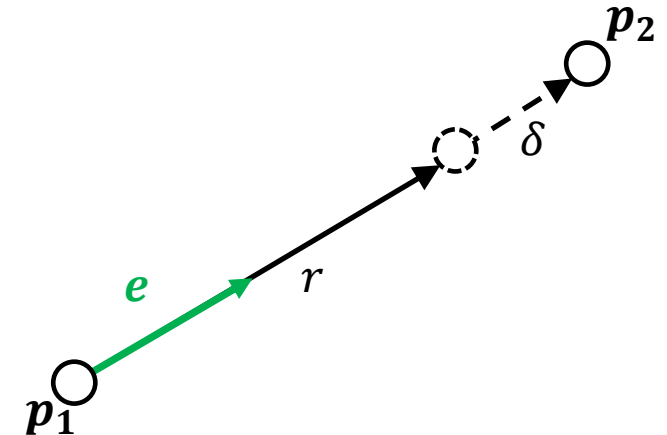
$$\mathbf{J} = \frac{\partial \mathbf{C}(\mathbf{q})}{\partial \mathbf{q}} = (-\mathbf{e} \quad \mathbf{e}) \text{ mit } \mathbf{W} = \begin{pmatrix} 1/m_1 & 0 \\ 0 & 1/m_2 \end{pmatrix}$$

- Es resultiert die Bindungsmasse

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

- Einsetzen in (11)

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



3) Herleitung

Orientierungserhaltende Bindung

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

$$(\mathbf{p}_2 - \mathbf{p}_1) - (\rho \mathbf{e} + \delta \tilde{\mathbf{e}}) = \mathbf{0} \quad (15)$$

- Multiplikation mit \tilde{e}

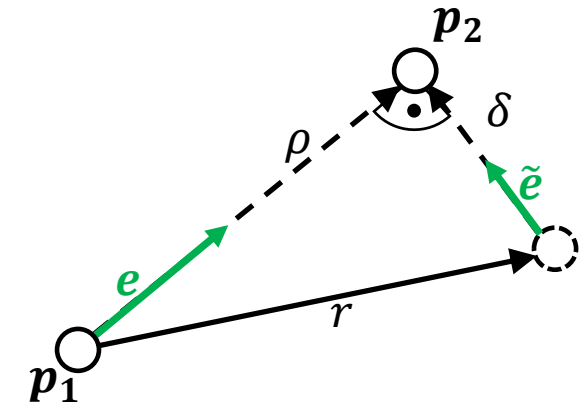
$$\delta = (\mathbf{p}_2 - \mathbf{p}_1) \tilde{\mathbf{e}} \quad (\mathbf{e} \cdot \tilde{\mathbf{e}} = 0)$$

- Aufteilung gemäß Nodemassen

$$\delta = \delta_1 + \delta_2 = \frac{m_c}{m_1} \delta + \frac{m_c}{m_2} \delta \quad (13)$$

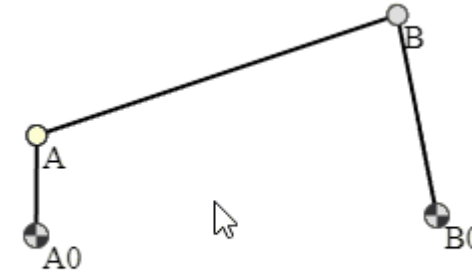
- Einsetzen in Bindungsgleichung

$$\left(\mathbf{p}_2 - \frac{m_c}{m_2} \delta \tilde{\mathbf{e}} \right) - \left(\mathbf{p}_1 + \frac{m_c}{m_1} \delta \tilde{\mathbf{e}} \right) - \rho \mathbf{e} = \mathbf{0} \quad (16)$$



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

```

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

```

//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
- Command Pattern
- 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>

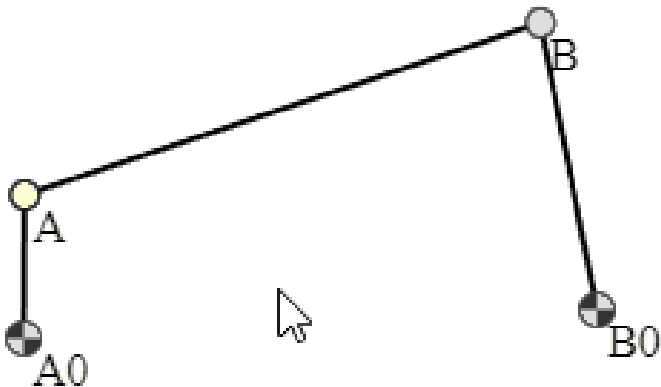
```

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

4) Implementierung

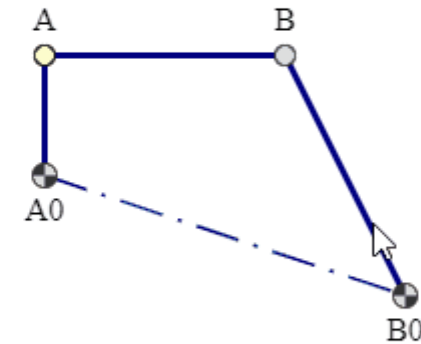
• Vorteile

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



max/itr= 1 / 1

correct



4) Implementierung

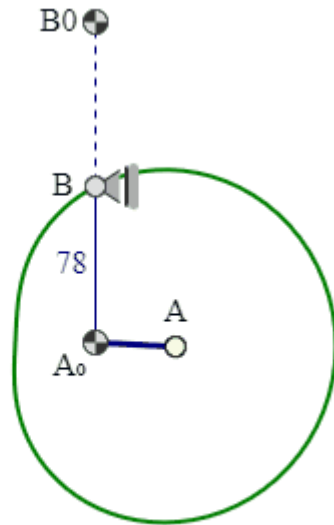
Sonstiges

- Konstanten:
 - Standardmäßige Längentoleranz: `cstr.lentol=0.1`
 - Standardmäßige Anzahl der Iterationen: `cstr.itrmax=256`
- Berechnung des Freiheitsgrades:
 - `cstr().dof`

4) Implementierung

Sonstiges

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



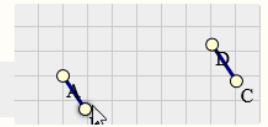
2.3 Constraints between Four Nodes

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'};
```

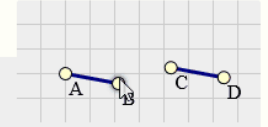
Two parallel lines

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



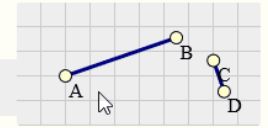
Two parallel lines of equal length

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



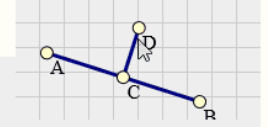
Two orthogonal lines

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



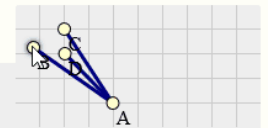
Perpendicular bisector of a line

```
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)})
```



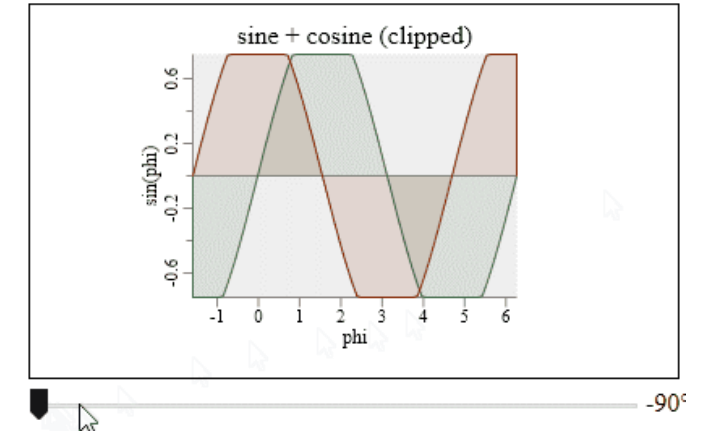
4) Implementierung

Sonstiges

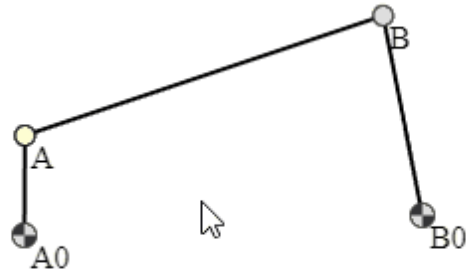
- Grafikbibliothekerweiterung g2-Chart

- Dokumentation:

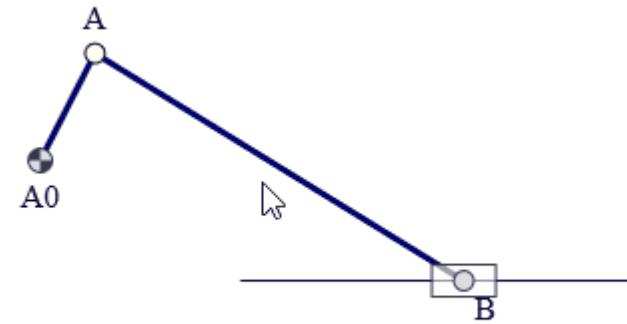
<https://goessner.github.io/g2/microjam.md/g2.chart.html>



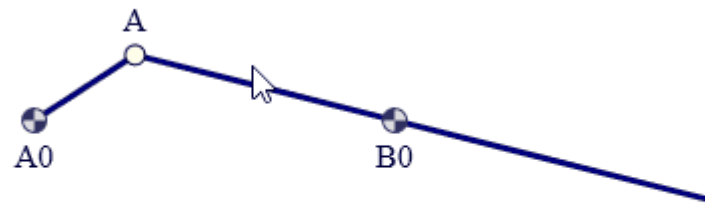
5) Beispiele



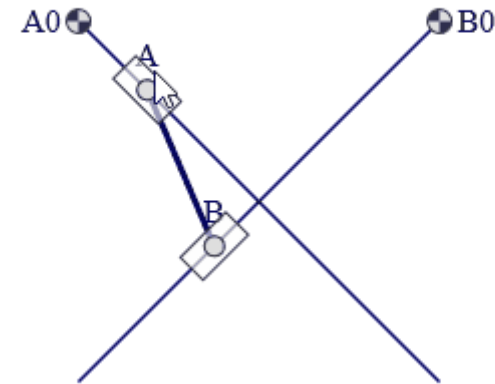
Kurbelschwinge



Schubkurbel



Schubschleife

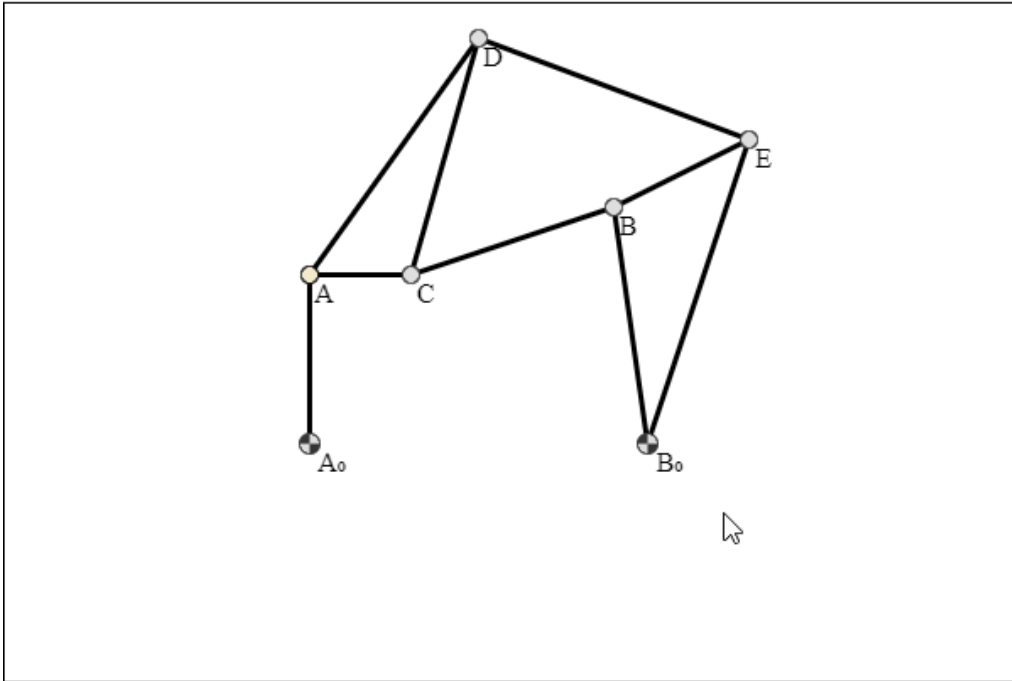


Kreuzschubkurbel

5) Beispiele

- Stephenson II Mechanismus
 - keine analytische Lösung

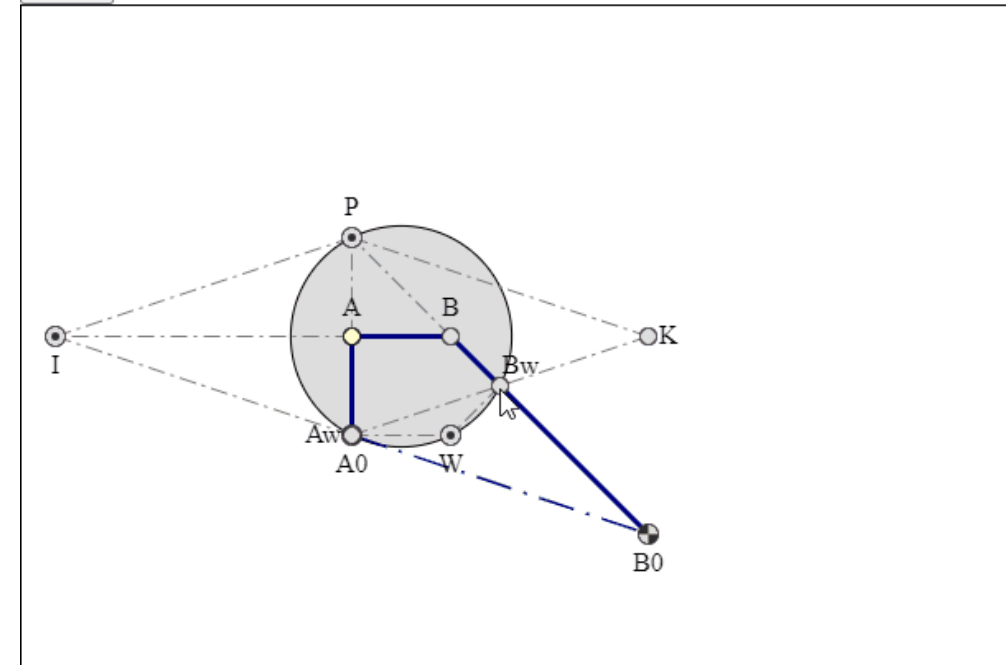
```
itr(max/cur) = 1 / 1
```



- Satz von Borell

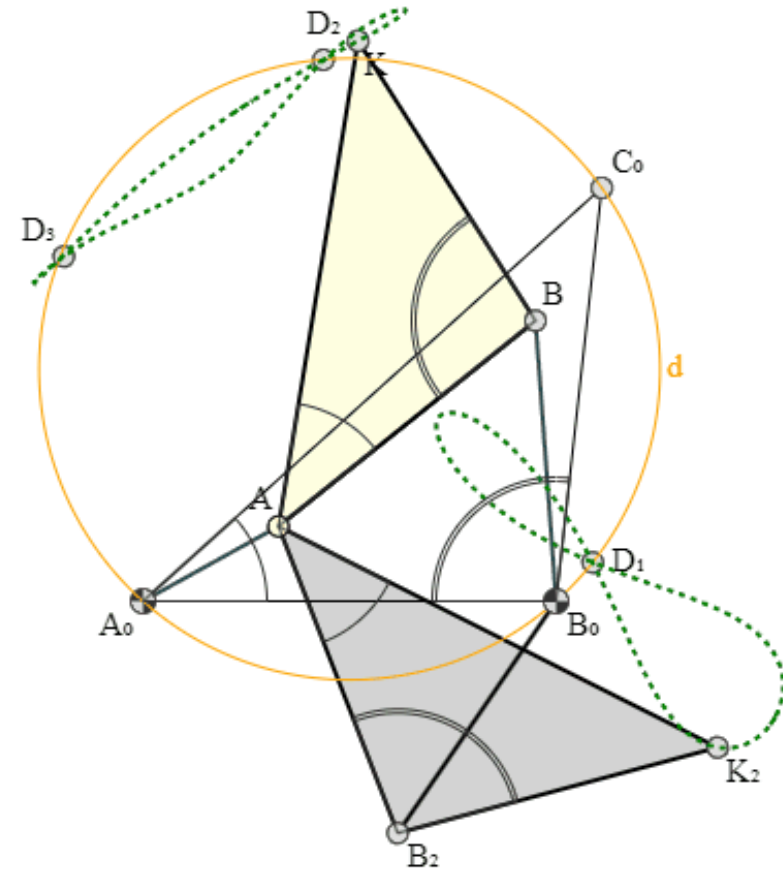
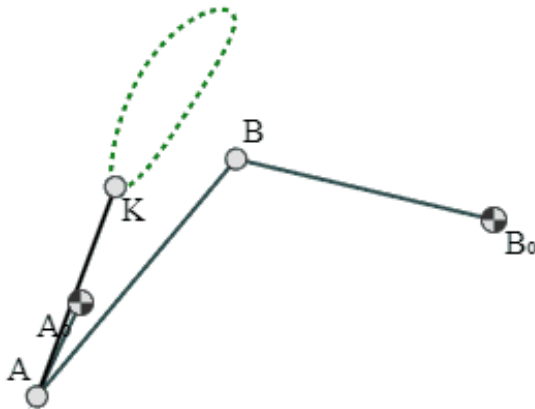
$$\text{itr}/\text{max} = 1 / 60$$

correct



Satz von Bobillier

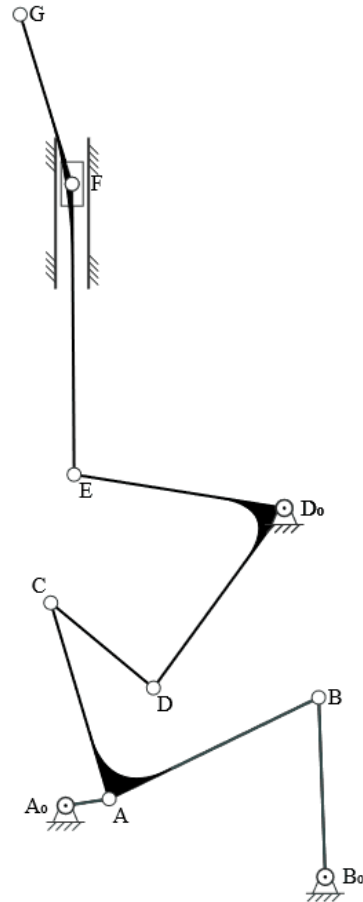
5) Beispiele Koppelkurven



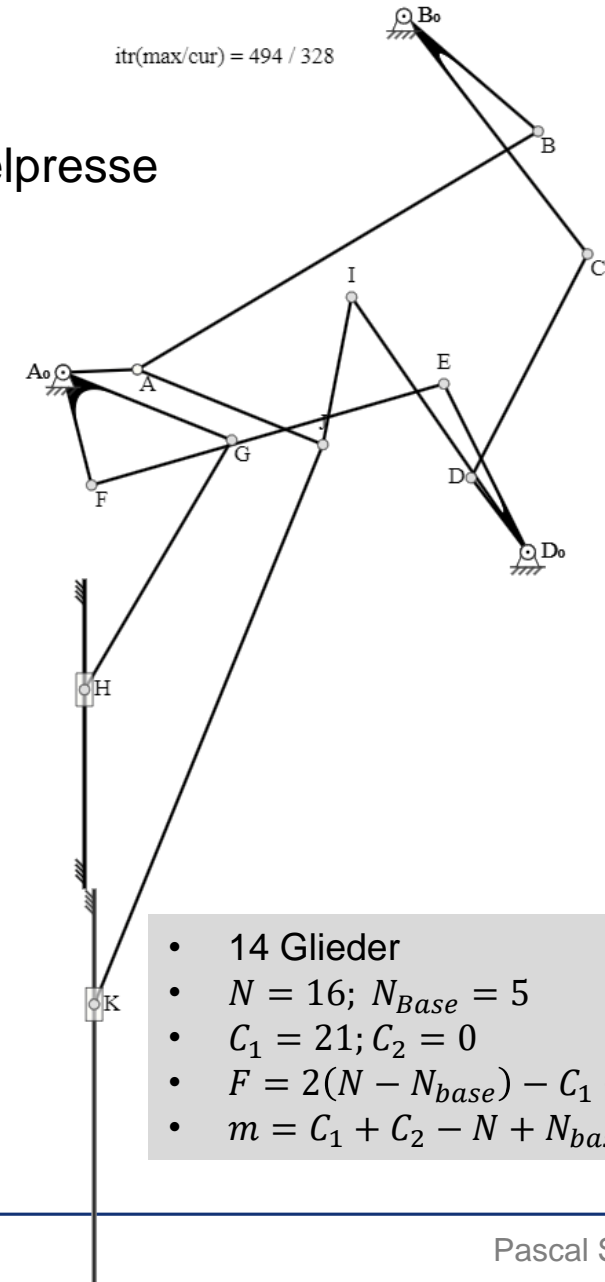
5) Beispiele

- Nadelbarrenantrieb einer Textilmaschine

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



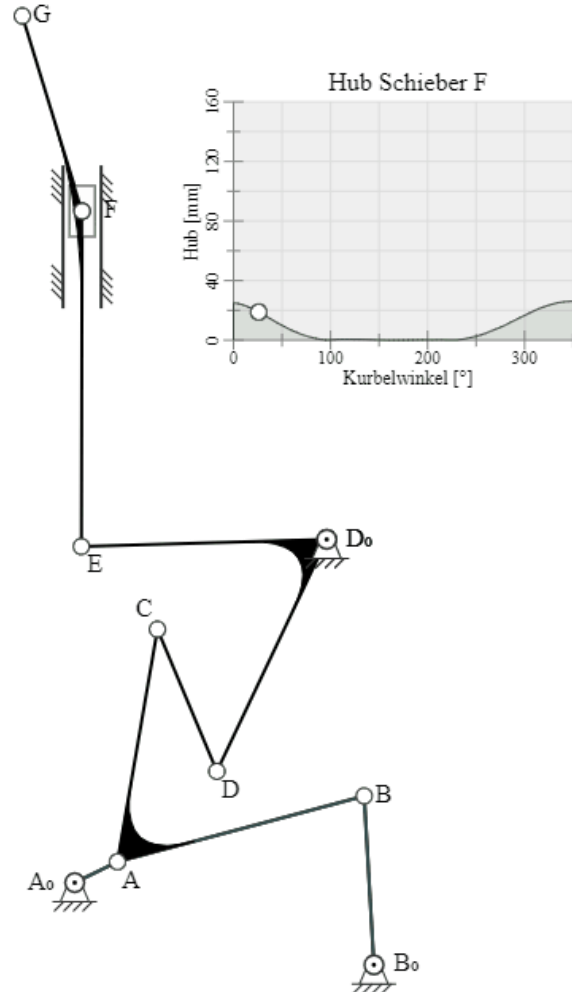
- Tiefziehkurbelpresse



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

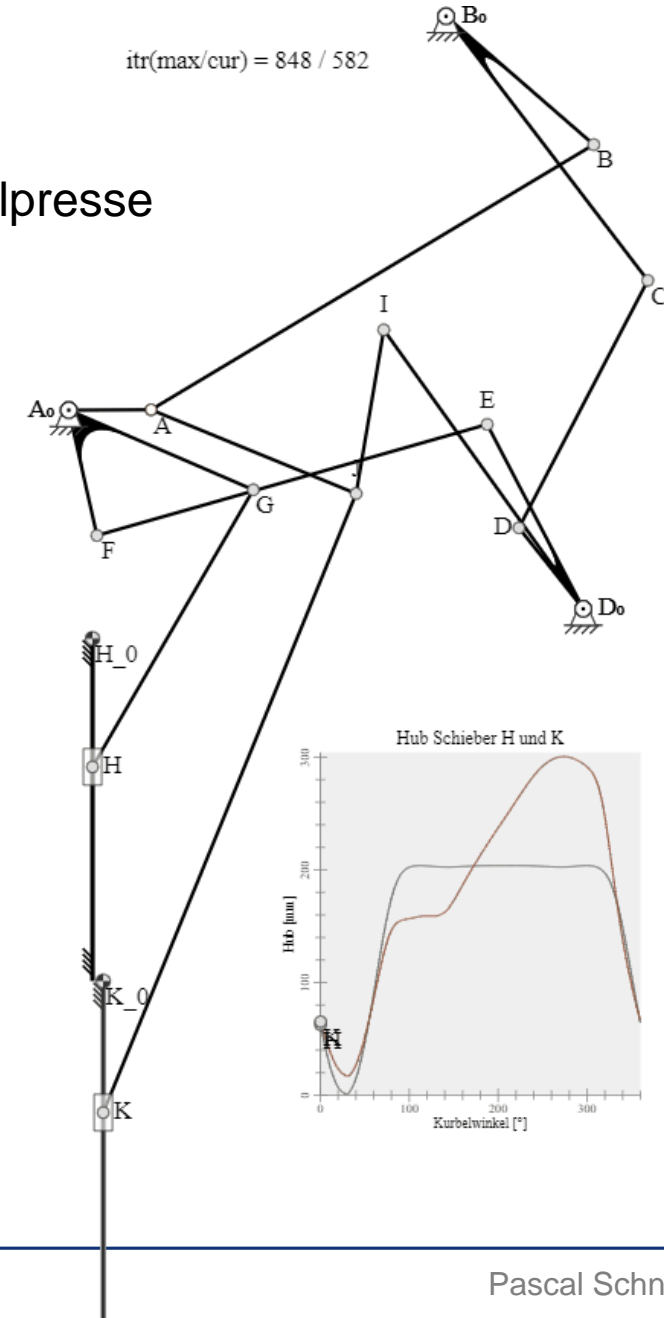
5) Beispiele

• 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



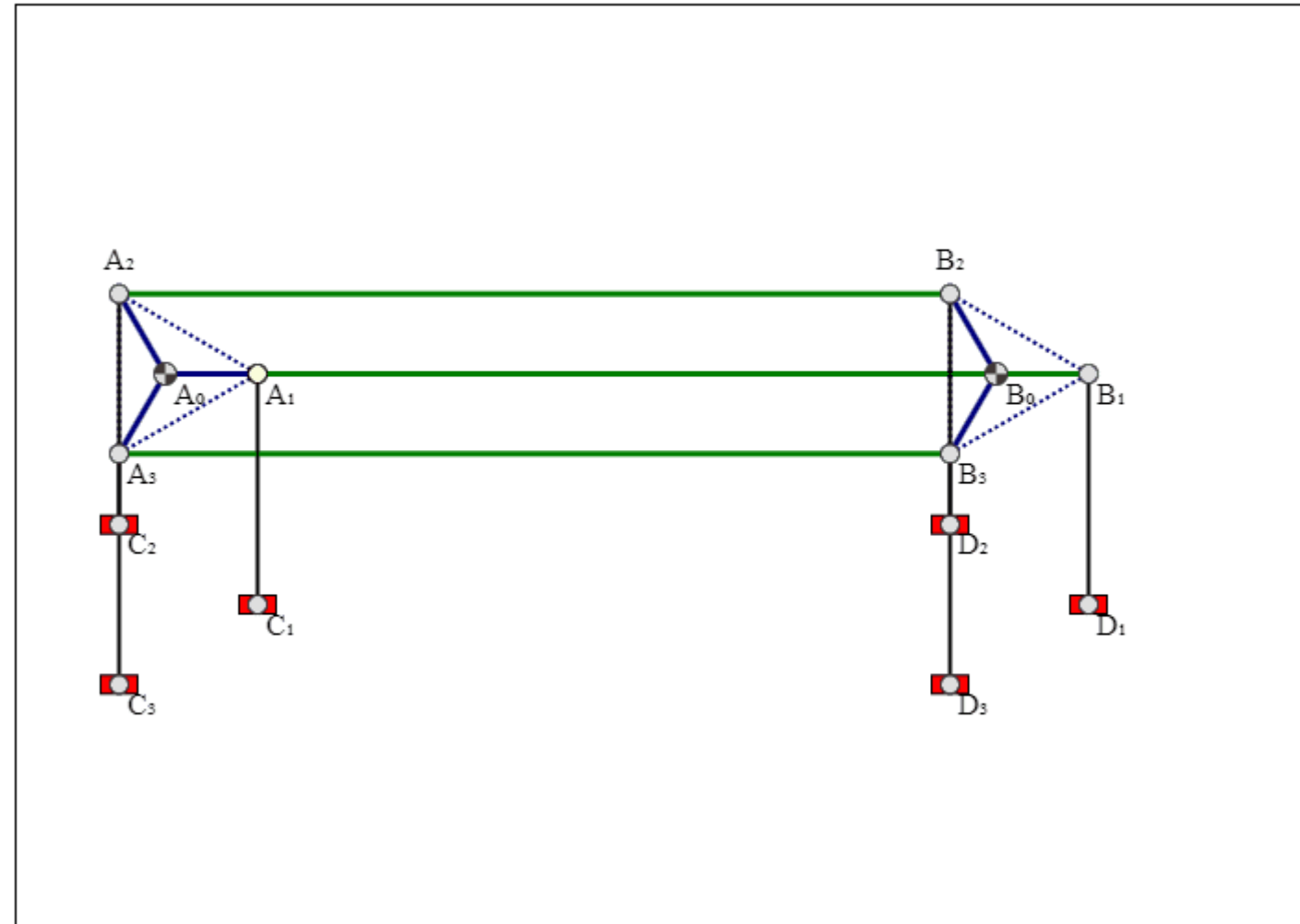
itr(max/cur) = 848 / 582

5) Beispiele Schreitkrabbler

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

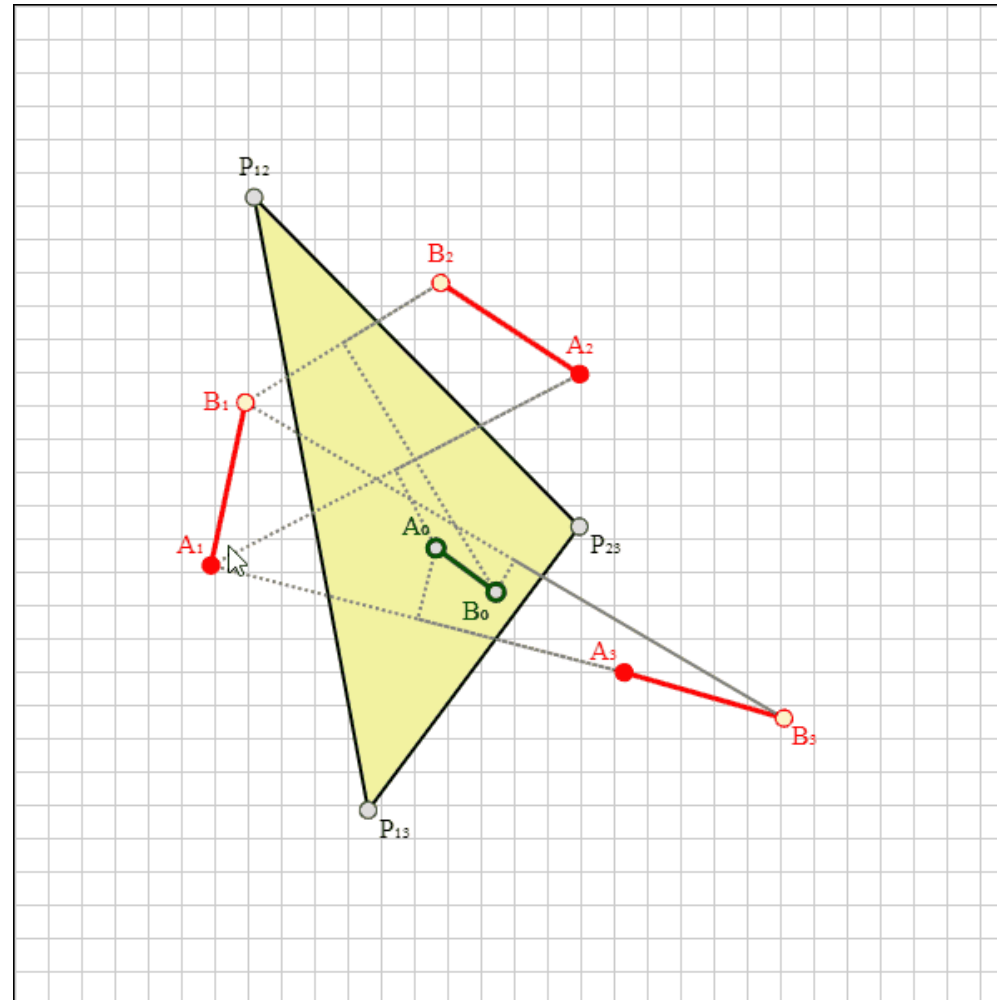
itr(max/cur) = 1 / 1

Dof:1



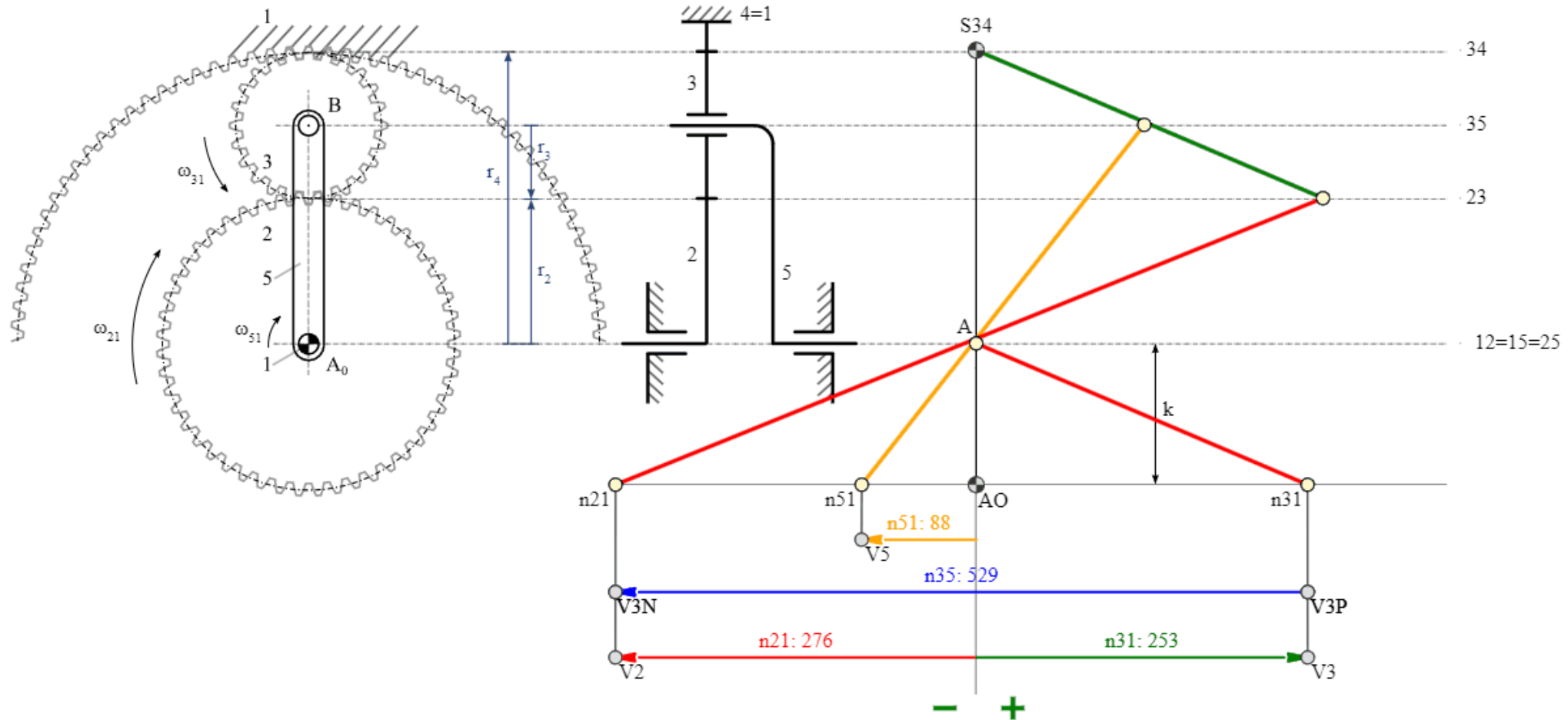
5) Beispiele

Synthese - Poldreieck



5) Beispiele

Drehzahlplan



6) Fazit

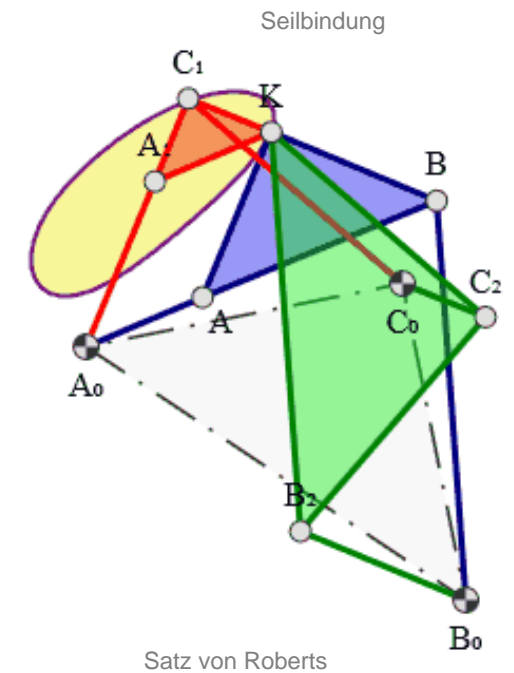
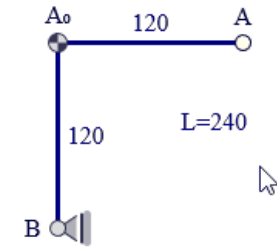
- Sehr einfache Anwendung
- Simulationsfehler, was nun?:
 - Reduktion der Schrittweite dt
 - alternative Modellierung
 - Steigerung der maximalen Iterationszahl `cstr.itrmax=1024`
 - Reduktion der Längentoleranz: `cstr.lentol=0.005`
 - ($n3()$ -Constraint stabiler)

7) Ausblick

- Veröffentlichung der Dokumentation
- Seilbindungen
- Animation weiterer Lehrsätze

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

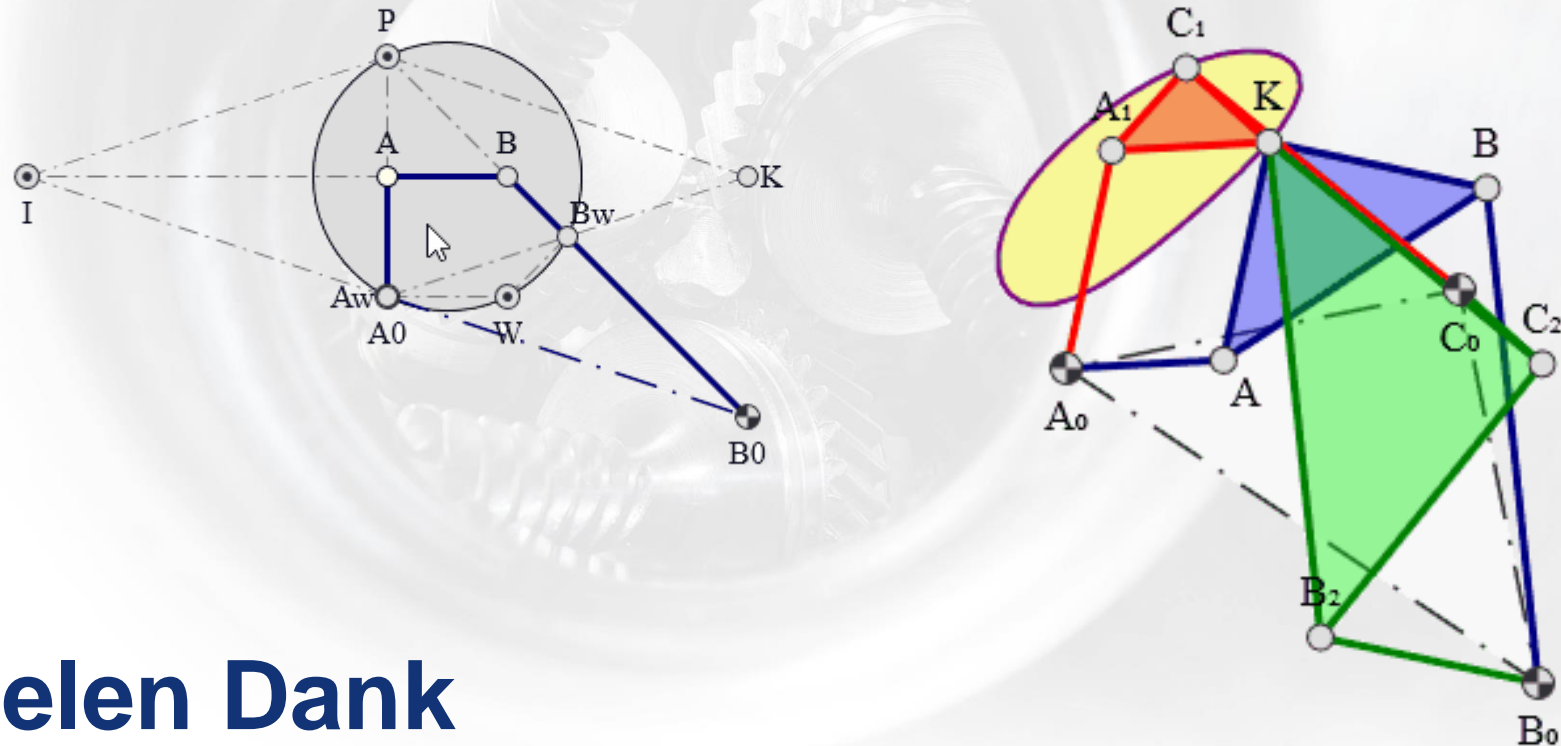
Email: mht@mb.tu-chemnitz.de
Tel: +49 (0) 371 531 – 32841
Fax: +49 (0) 371 531 – 23739

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