

Vorlesung Wintersemester 2016/17:

# **Numerische Methoden in der Biomechanik**

*Dr.-Ing. Ulrich Simon*

Hinweis auf andere Vorlesungen

„**MKS** in der Biomechanik“

„Beispiele **FEM** in der Biomechanik“

# Gliederung

- Einführung / Übersicht / Allgemeines
- Beispiel: Simulation der Frakturheilung
  - Finite-Elemente-Methode (FEM)
  - Fuzzy-Logic
- Einführung FEM
- Einführung MKS
- Hinweise zum Einstieg

# Modellbildung

---

## Wissenschaft will ...

- Beobachtungen erklären,
- Vorhersagen treffen.

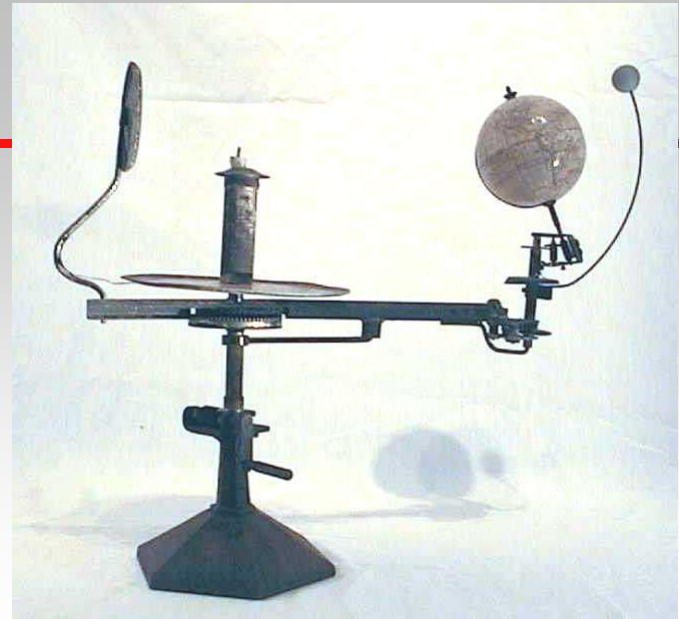
⇒ Modelle

## Modelle sind ...

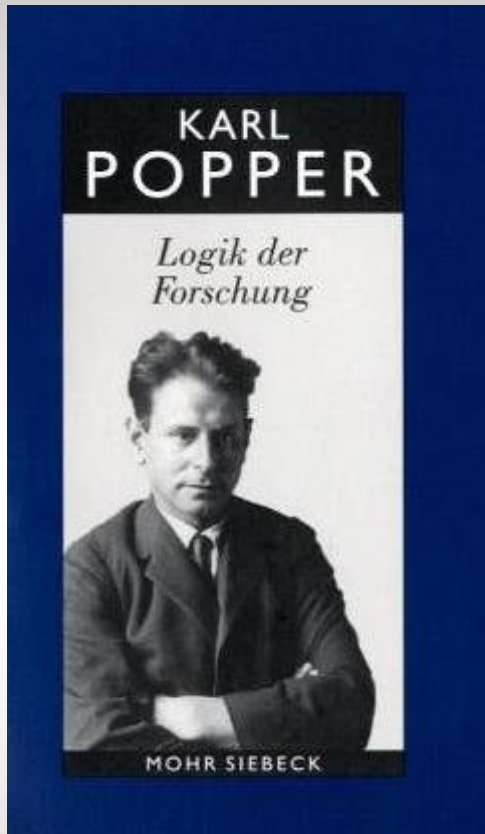
- Abbildungen von Systemen/Prozessen in andere begriffliche oder gegenständliche Systeme,
- mehr oder weniger phänomenologisch,
- fragespezifisch,
- immer Näherungen, Vereinfachungen, Idealisierungen
- mit beschränktem Gültigkeitsbereich.
- **das gilt auch für Experimente und klinische Tests!**

## Ein gutes Modell ist ...

...so kompliziert wie nötig und so einfach wie möglich!

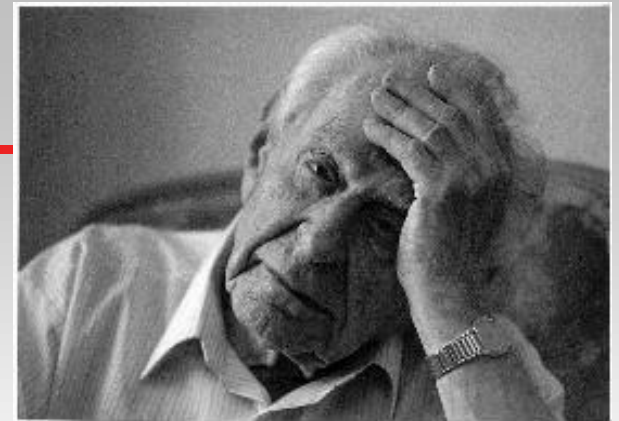


# Modellbildung

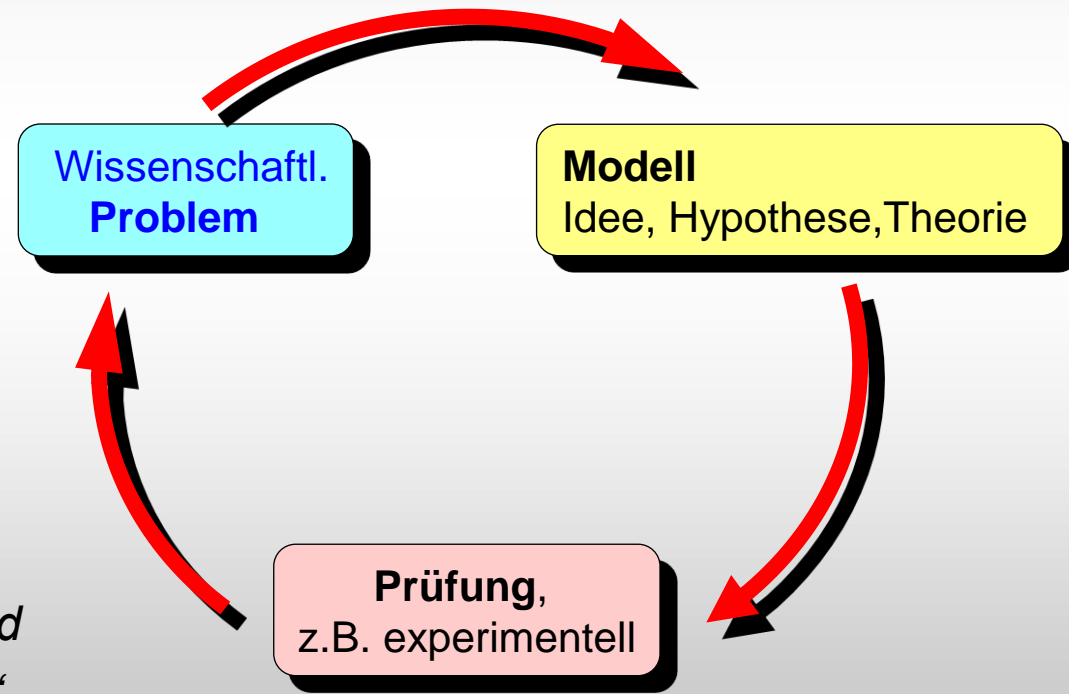


1934

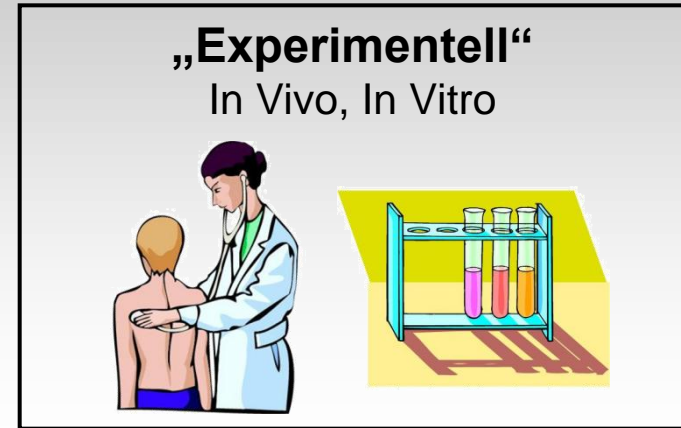
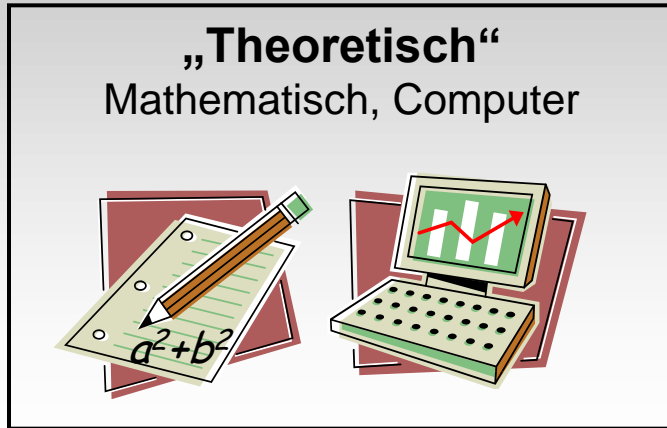
*„Wissenschaftliche Aussagen sind immer modellhaft, also vorläufig!“*



Sir Karl Raimund Popper (1902-1994)



# Theoretische gegenüber experimentellen Methoden



## Eigenschaften theoretischer Methoden

- + Reproduzierbar, eindeutig (determiniert), leicht verfügbar.
- + Parameterstudien durch vielfache Wiederholungen möglich.
- + Modellierung von unerreichbaren oder fiktiven Prozessen möglich.
- + Ethisch unbedenklich
  - Oft stark vereinfacht
- Modellparameter und Validierung erfordern oft zusätzliche Experimente.

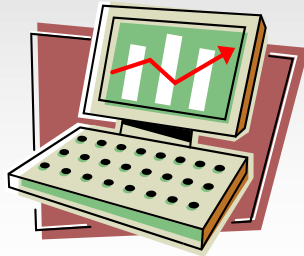
## Eigenschaften experimenteller Methoden

- + Oft realistischer: „Die Natur vergisst keinen Effekt“.
  - Ohne tiefes Verständnis möglich
- Immer Fehlerbehaftet: Unbekannte Einflüsse, Messfehler
- Aber: Selbst In-vivo-Studien sind stets modellhaft und deren Ergebnisse nur eingeschränkt gültig.

# Numerische gegenüber analytischen Methoden

## „Numerisch“

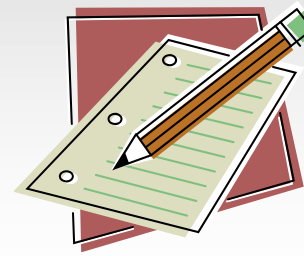
⇒ Rechnen mit Zahlenwerten



1.37491

## „Analytisch“

⇒ Rechnen mit Variablen



$a^2+b^2$

### Eigenschaften numerischer Methoden

- Oft Näherungsverfahren
- Alle Parameter, Rand- und Anfangsbedingungen werden als Zahlenwerte eingegeben: Lösung gilt nur für diesen speziellen Fall.
- Parametereinflüsse müssen durch vielfache Rechenwiederholungen untersucht werden.
- Für viele komplexe Probleme sind numerische Methoden bekannt.

### Eigenschaften analytischer Methoden

- (Mathematisch) exakte Lösung.
- Analytische Gleichungen mit Variablen: allgemeingültige Aussagen für ganze Modellklassen.
- Parametereinflüsse sofort, allgemeingültig (analytisch) zu erkennen.
- Analytische Lösungen sind nur für wenige einfache Probleme bekannt.

# Landkarte der numerischen Methoden

## Daten charakterisieren und analysieren

Daten = gemessene, physikalische Größen

- ⇒ Statistik: Mitteln, Ausgleichen
- ⇒ Bildverarbeitung
- ⇒ Komprimieren (JPEG)

## Festigkeitsanalysen (Randwertproblem)

- Spannungen, Dehnungen, Verformungen
- Hält das Implantat?

- ⇒ Differentialgleichung (Randwertproblem)
- ⇒ Finite-Elemente-Methode (FEM)

## Signale charakterisieren und analysieren

Signal = gemessene, zeitabhängige Größe (z.B. EMG).

- ⇒ Glätten: Filtern (Hoch-, Tiefpass)
- ⇒ Frequenzanalyse: Fourieranalyse (FFT)
- ⇒ Wavelet-Transformation (JPEG-Kompression)

## Dynamische Prozesse simulieren, steuern, regeln

- ⇒ Differentialgleichungen
- ⇒ Fuzzy Logic
- ⇒ Neuronale Netze

## Bewegung beschreiben

Kinematik: Bewegung gemessen, Beschreibung gesucht.

- ⇒ Euler-, Kardanwinkel
- ⇒ Helikale Achsen

## Bewegung analysieren

Inverse Dynamik: Bewegung gemessen, Muskelkräfte gesucht.

- ⇒ Lösen unterbestimmter Gleichungssysteme
- ⇒ Optimierung

## Bewegung simulieren (Anfangswertproblem)

(Vorwärts-)Dynamik: Kräfte gegeben, Bewegung gesucht.

- ⇒ Differential (Anfangswertproblem)
- ⇒ Mehr-Körper-Simulation (MKS)
- ⇒ Explizite FEM (Crashsimulation)

## Parameteridentifikation

- Ein- und Ausgang gegeben, System(-parameter) gesucht
- ⇒ Kleinste Fehlerquadrat-Summe

# Numerische Methoden allgemein

---

## Lineare und nichtlineare Gleichungssysteme

- Gaußscher Algorithmus (für lineare Gleichungssysteme)
- Spezielle Algorithmen (symmetrische Systeme, Bandstruktur, positiv definite Systeme)
- Newton-Verfahren (für nichtlineare Gleichungssysteme)

## Optimierungs- oder Ausgleichsprobleme

- Kleinste Fehlerquadrate (least squares)
- z.B. Inverse Dynamik und Berechnung der Muskelkräfte beim Gang

## Eigenwertprobleme

- meist dynamische (periodische) Probleme
- z.B. Berechnung von Eigenfrequenzen

## Anfangswertprobleme

- meist dynamische (transiente) Probleme
- numerische Integration (Runge-Kutta-Verfahren)
- Mehr-Körper-Simulation (**MKS**)

## Randwertprobleme

- oft statische Probleme (z.B. Festigkeitsanalysen)
- Finite-Elemente-Methode (**FEM**)

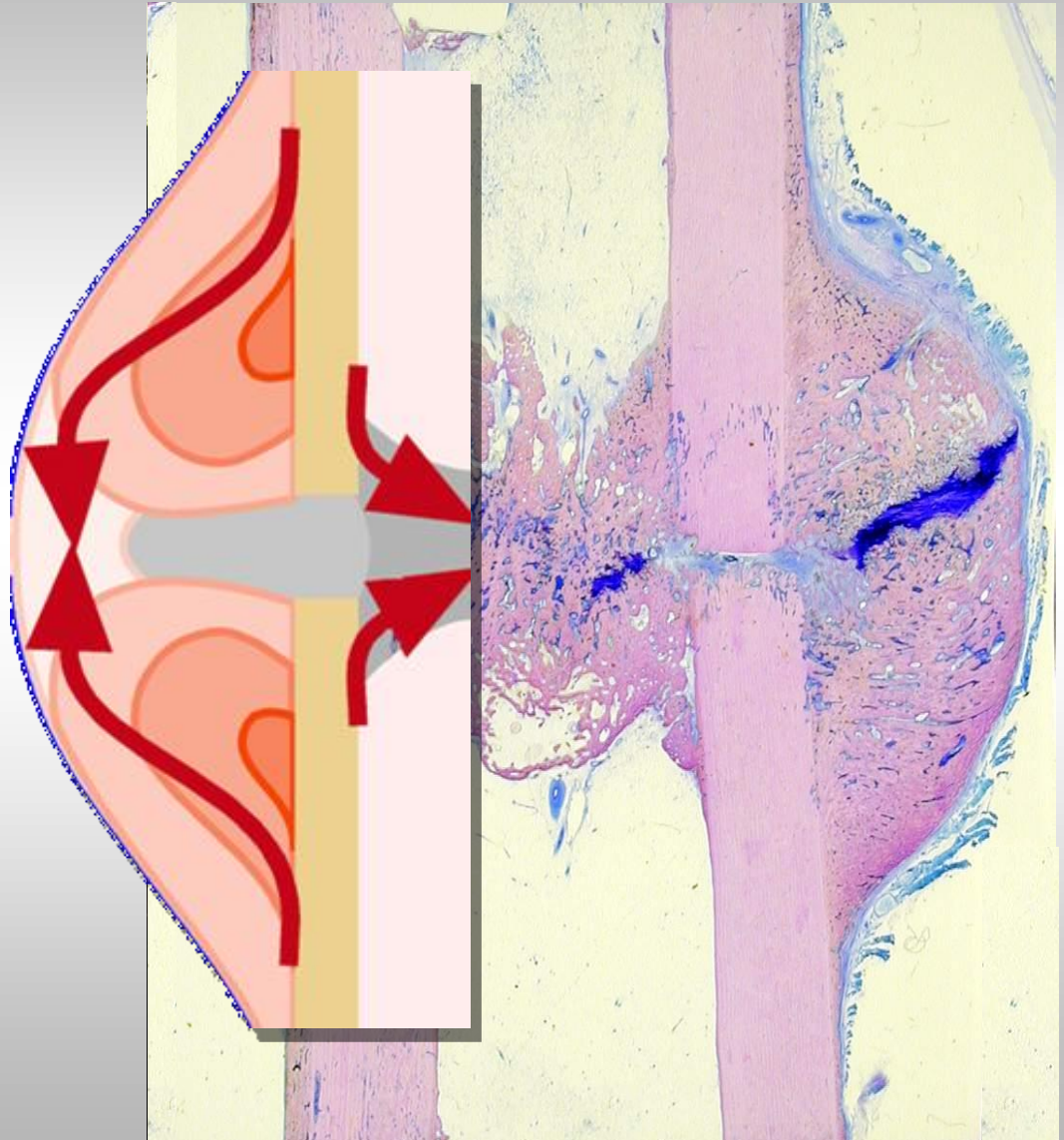


# Einheitensysteme

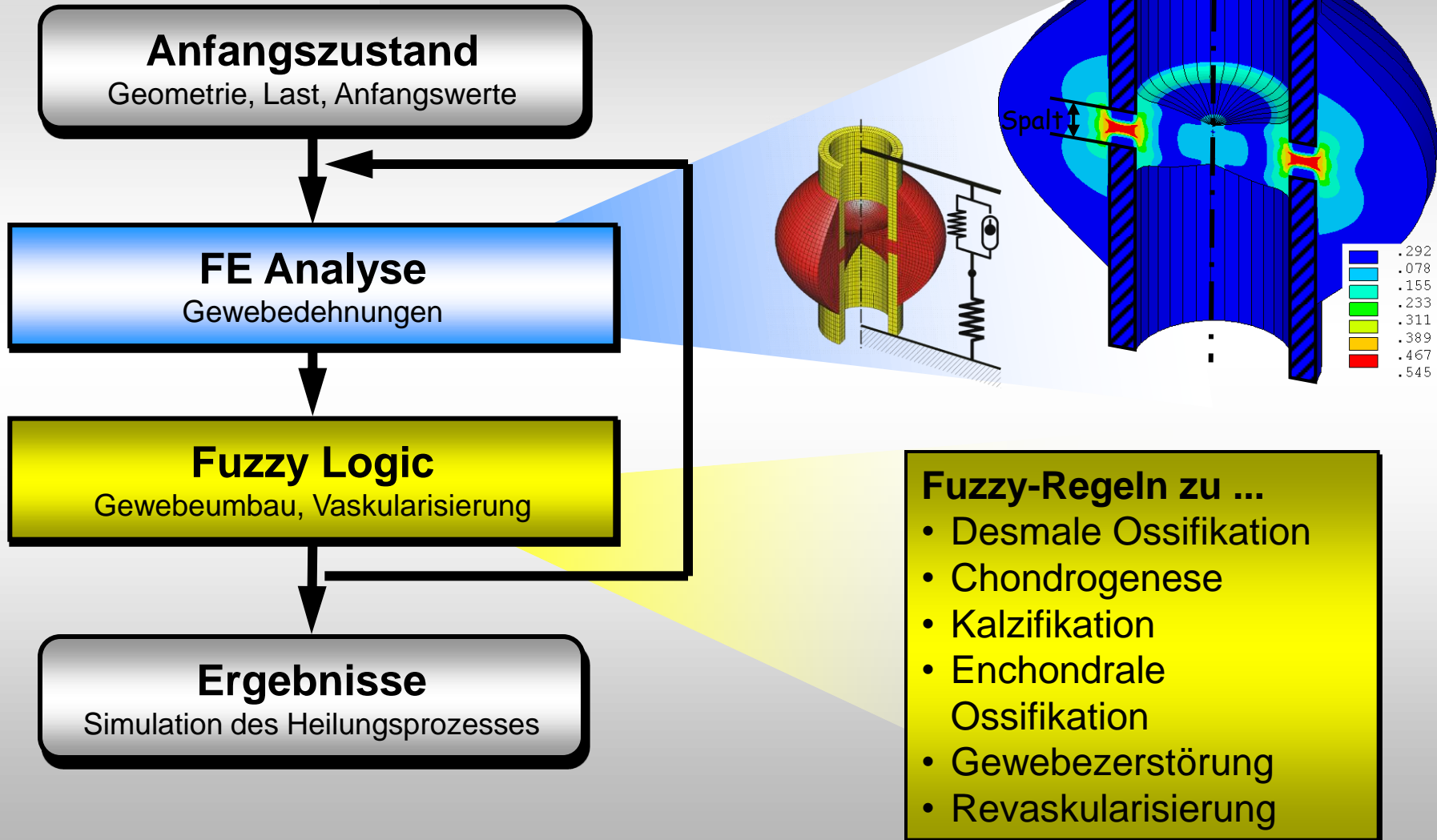
<b>Basiseinheiten</b>			<b>Abgeleitete Einheiten</b>					<b>Bemerkung</b>
<i>Länge</i>	<i>Masse</i>	<i>Zeit</i>	<i>Kraft</i> = <i>Masse*Länge/Zeit²</i>	<i>Spannung</i> = <i>Kraft/Länge²</i>	<i>Dichte</i> = ...	<i>Beschl.</i> = ...	...	
m	kg	sec	N	Pa	kg/m³	m/sec²		SI-Basis-Einheiten
mm	t	sec	N	MPa	t/mm³	mm/sec²		Organ-Level
µm	t	sec	mN	GPa				Tissue-Level

Merke: Übliche (insbesondere altmodischere) Simulationsprogramme arbeiten wie ein Taschenrechner: man kann nur Zahlen und keine Einheiten eingeben. Also muss man wissen, welches Einheitensystem bei der Programmierung vorausgesetzt wurde. Üblicherweise das SI-Basis-System (m, kg, sec) (1. Zeile). Dann kann man ohne Schaden auch immer ein konsistentes SI-System bei der Eingabe der Zahlen verwenden (z.B. 2. und 3. Zeile). Dieses muss man dann natürlich konsequent für alle Ein- und Ausgaben verwenden. Bei moderneren Programmen kann man ggf. auch auf die Verwendung von nicht konsistenten Einheiten-Systemen umschalten oder gar bei den einzelnen Ein- und Ausgaben individuell die Einheit wählen.

# Beispiel: „Simulation der Frakturheilung“



# Simulation der Knochenheilung



# Fuzzy Logic: Allgemeines

---

## Ursprung:

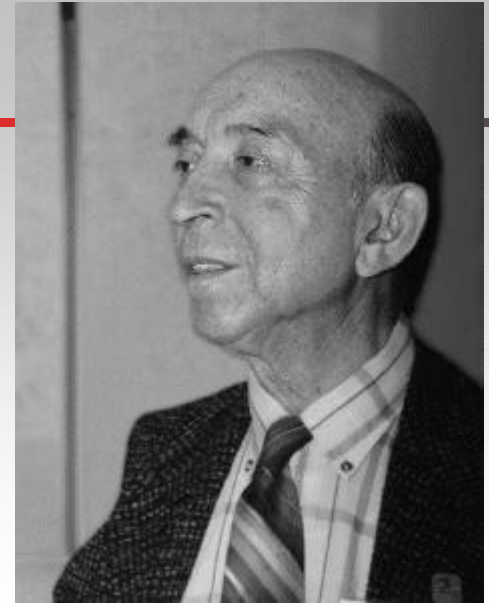
- Lotfi A. Zadeh, Berkeley (1965): „Fuzzy Sets“

## Prinzip:

- Fuzzy Logic = „Krause Logik“ im Gegensatz zur scharfen Logik
- *MEHR-ODER-WENIGER* statt *ENTWEDER-ODER*

## Einsatzgebiete:

- Regelungstechnik (komplexe Mehrvariablen-Regelung)
- Industrieautomation und Konsumgüter
- Sensorik, Datenanalyse, Betriebs- und Finanzwirtschaft

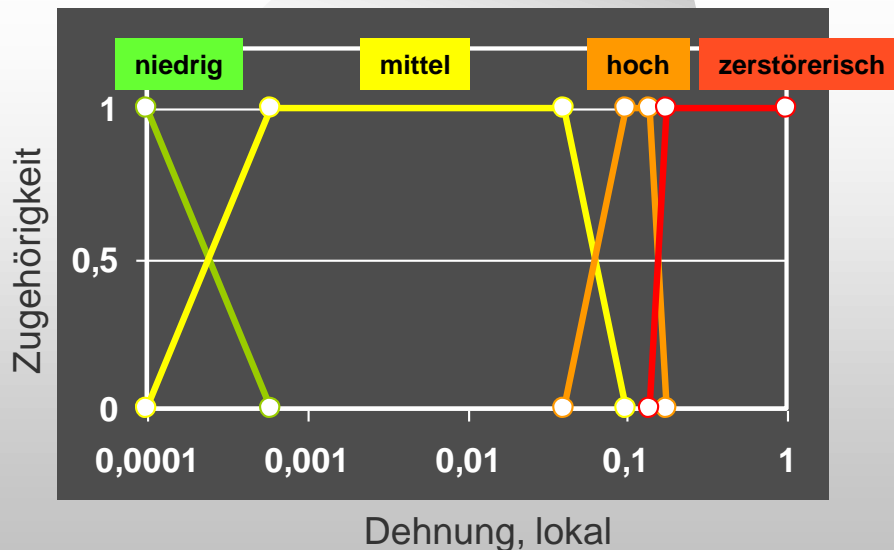


L.A. Zadeh

# Fuzzy-Regler



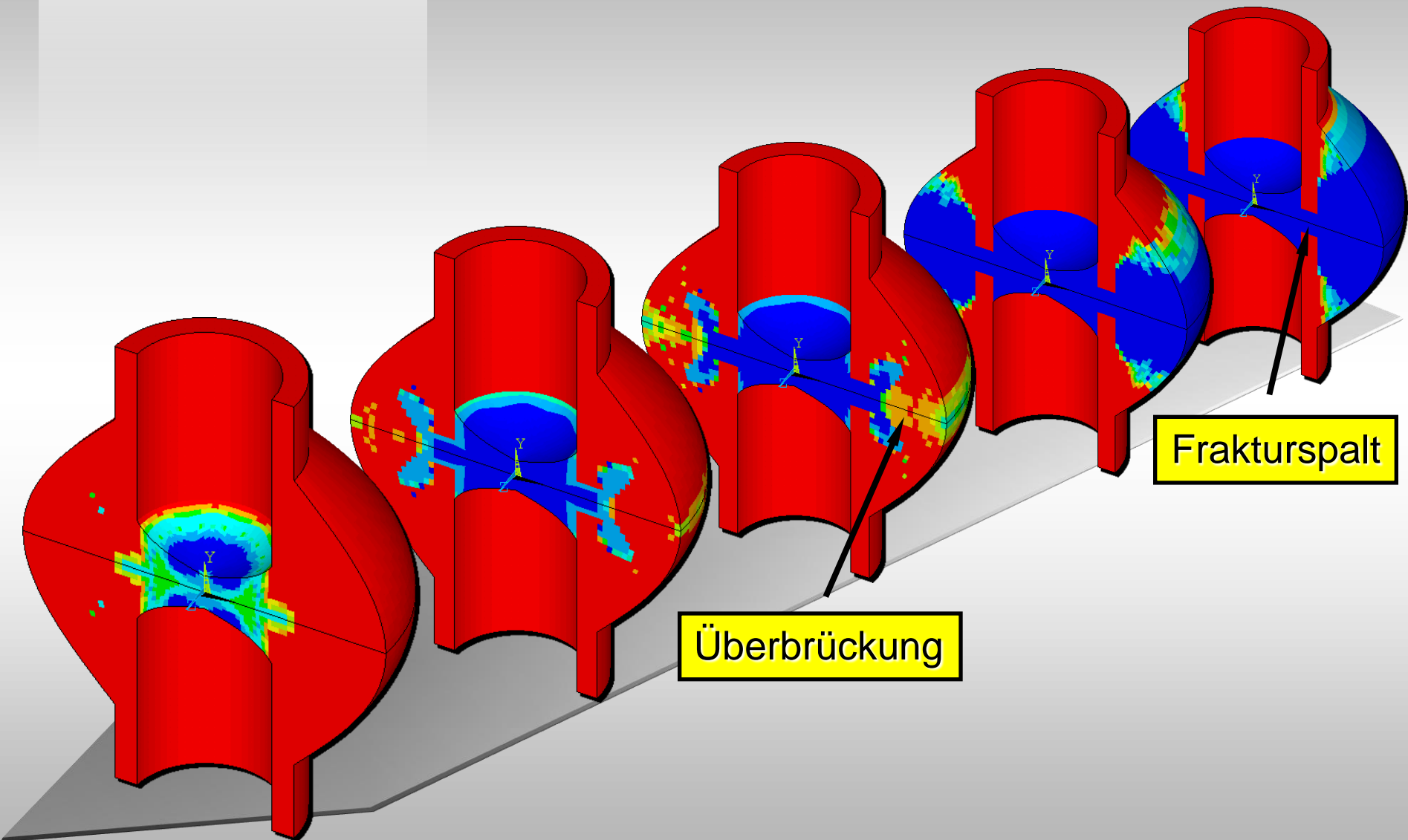
## Fuzzyfizierung:



## Fuzzy-Regeln:

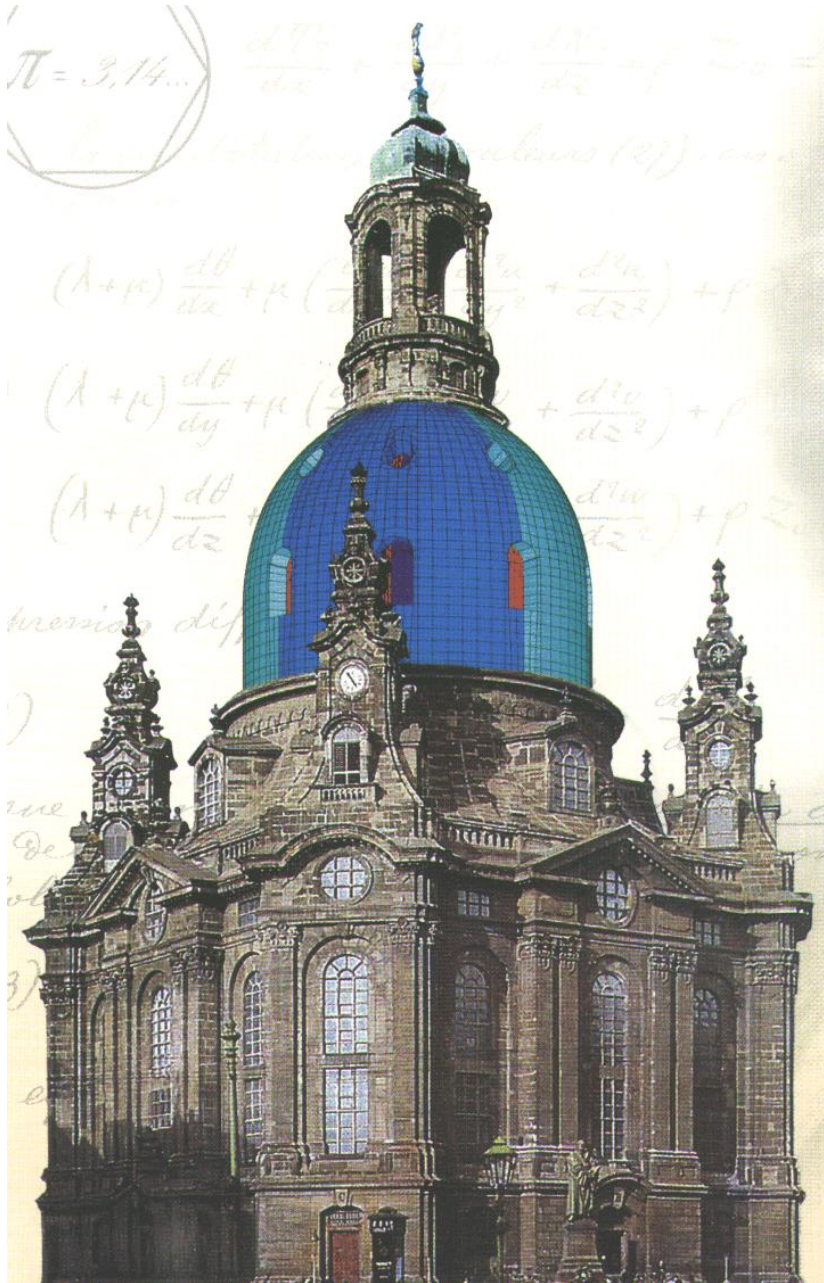
Wenn ...  
Durchblutung = gut  
und  
Dehnung = mittel,  
  
dann ...  
Knochenanteil = erhöhen.

# Ergebnis: Knochenneubildung



# Anwendungsbereiche FEM





# Statik, Festigkeit

Wie stark Verformt es sich?

Hält es?

- Verschiebungen
- Spannungen, Dehnungen

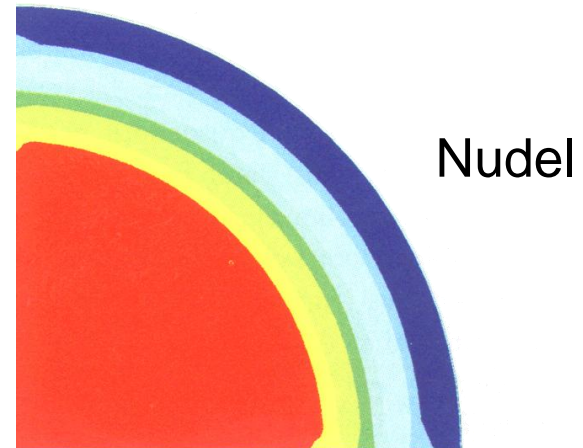
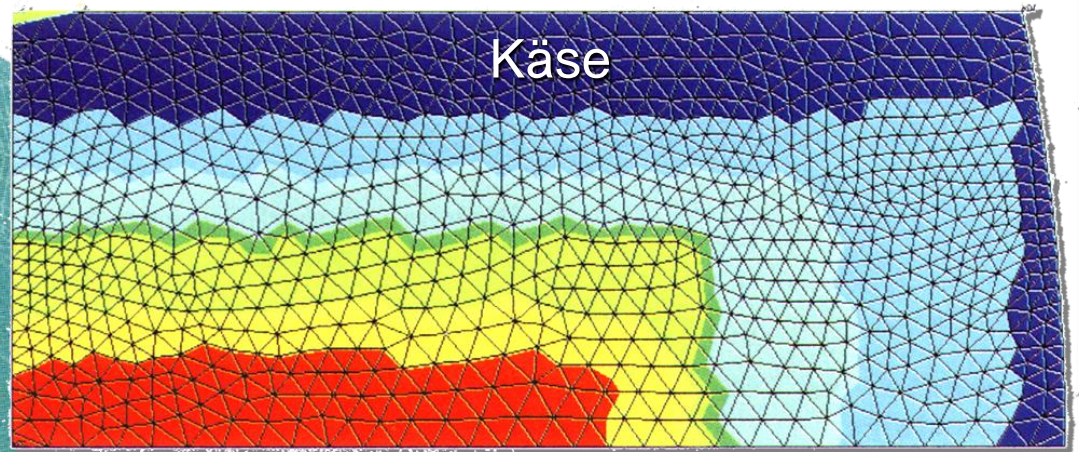
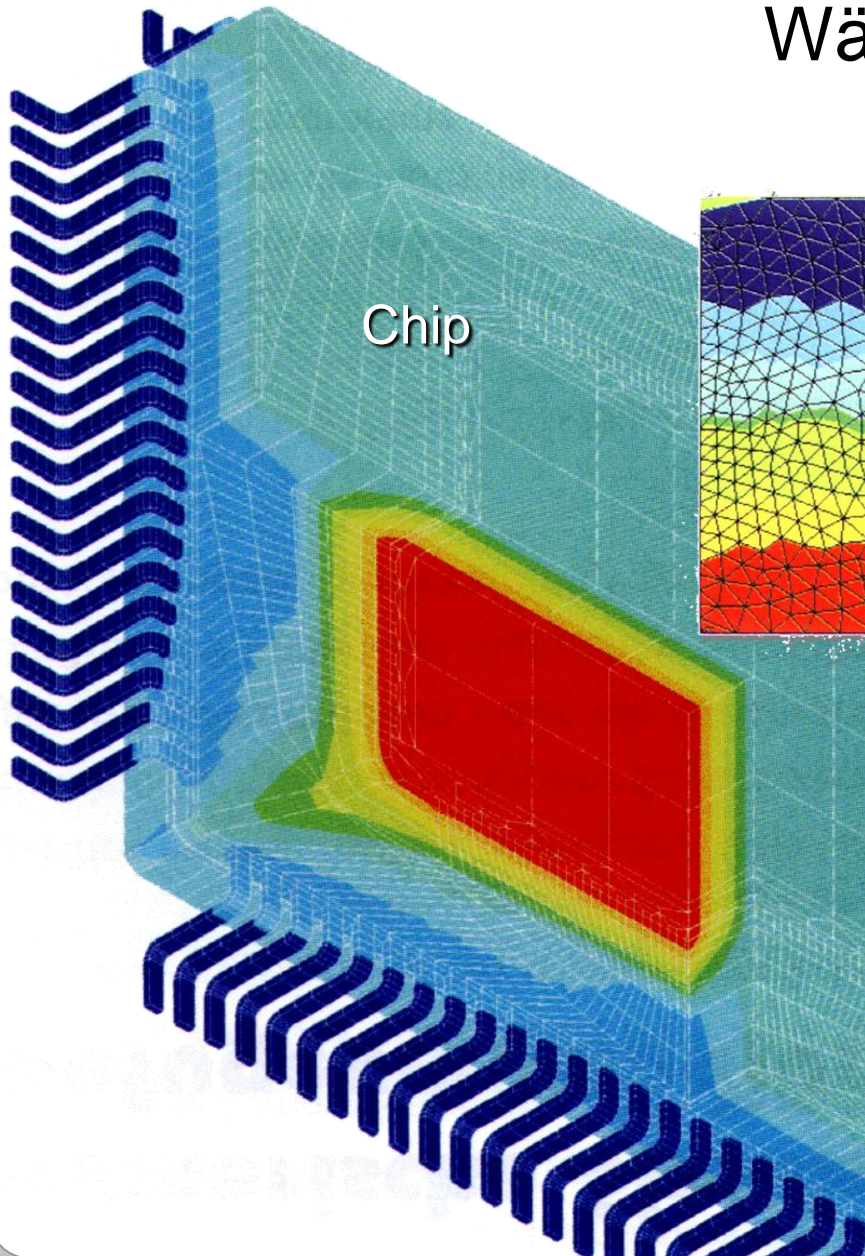
Nichtlinear:

- Kontakt, Reibung
- Plastizität, Verfestigung
- Ermüdung, Bruchmechanik

Frauenkirche, Dresden

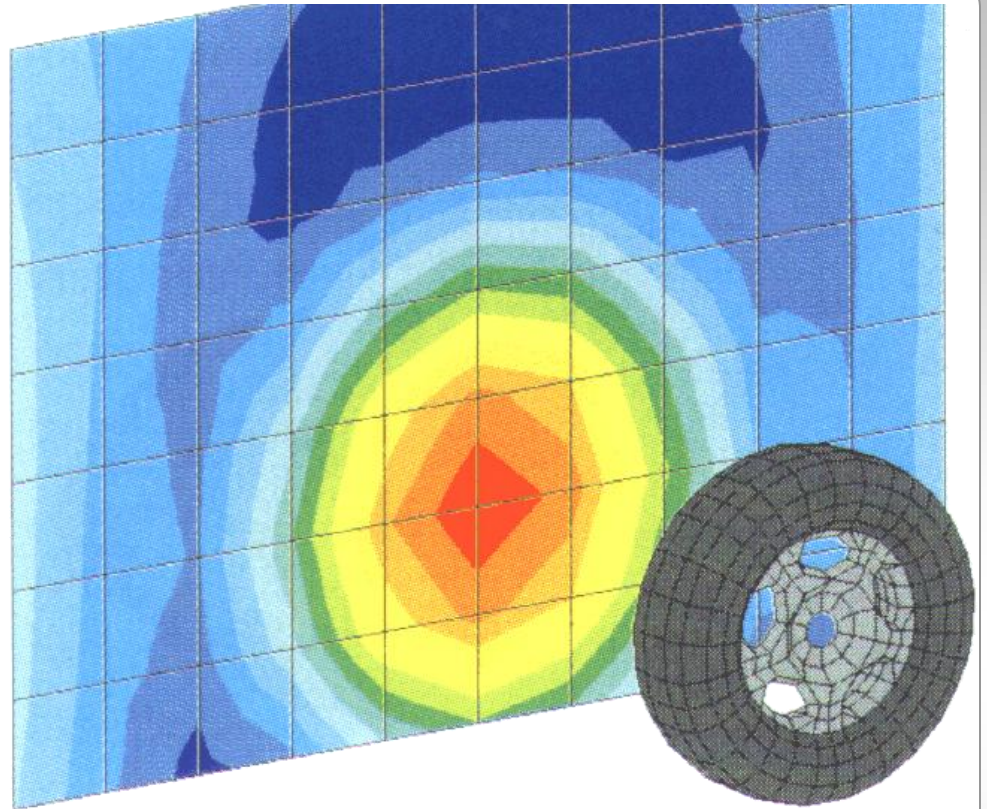
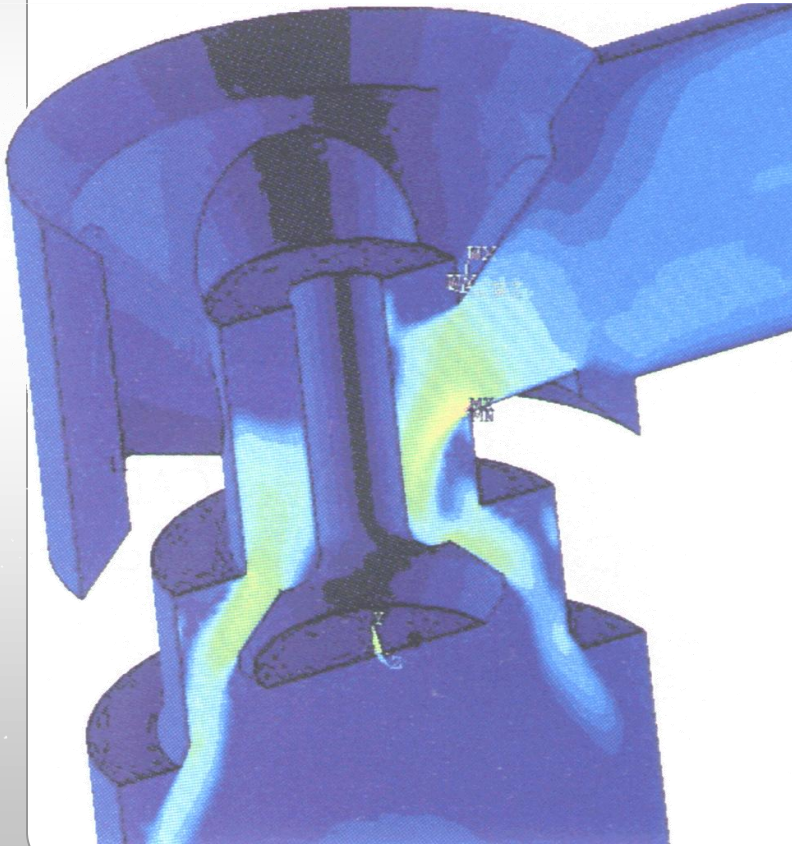


# Wärmeleitung und Diffusion



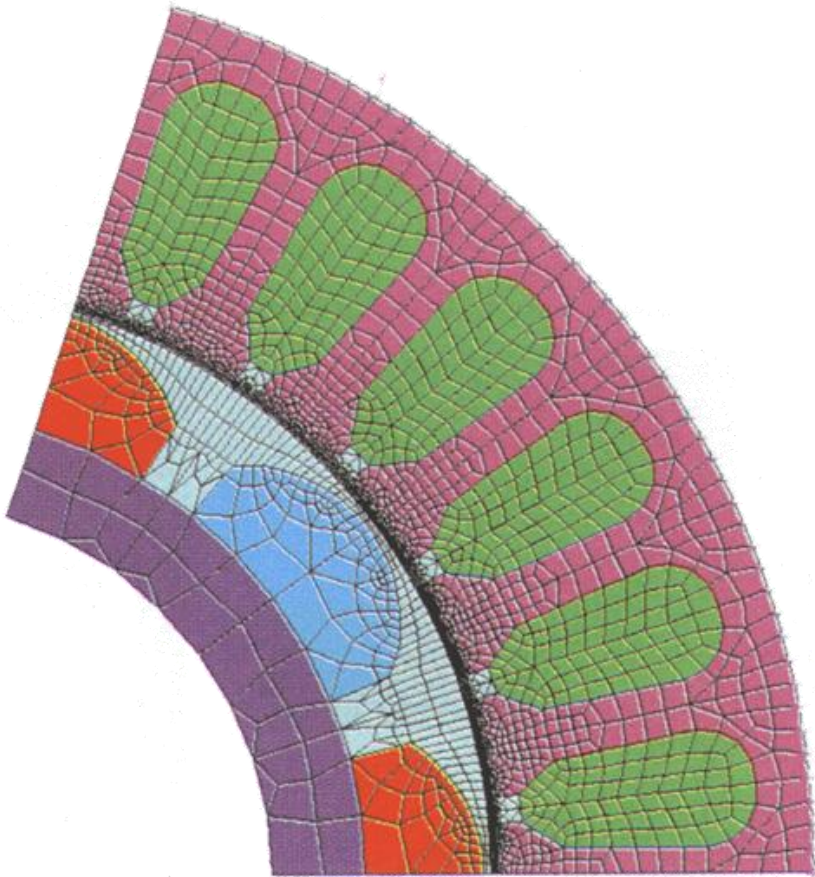


# Strömungen

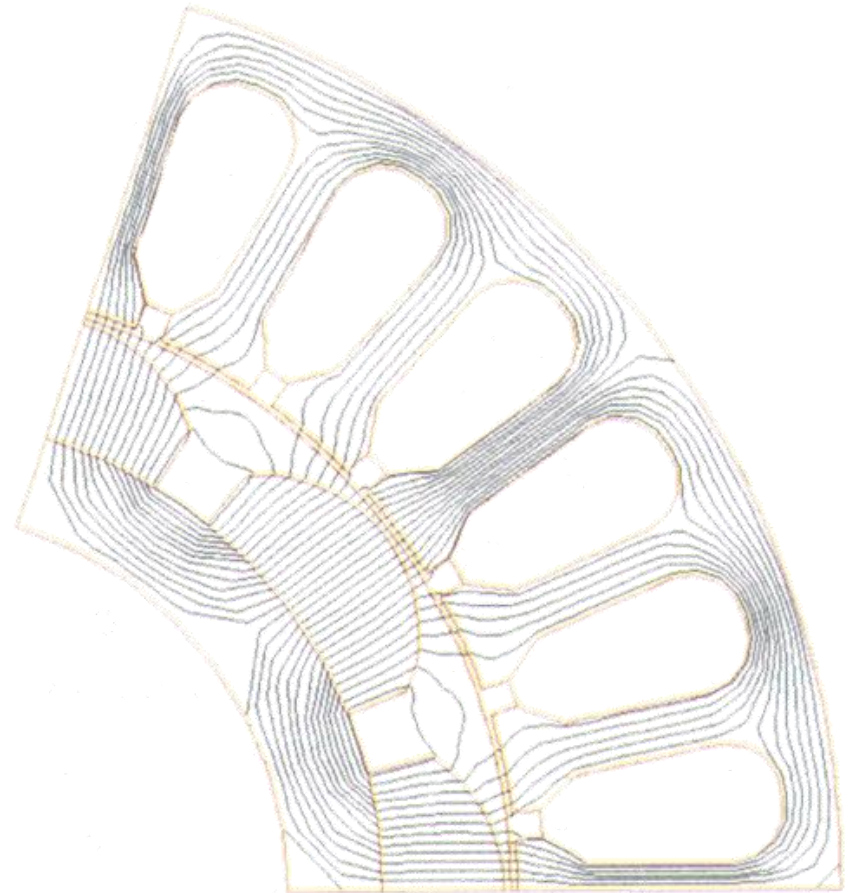


Akustik

# Elektromagnetische Felder



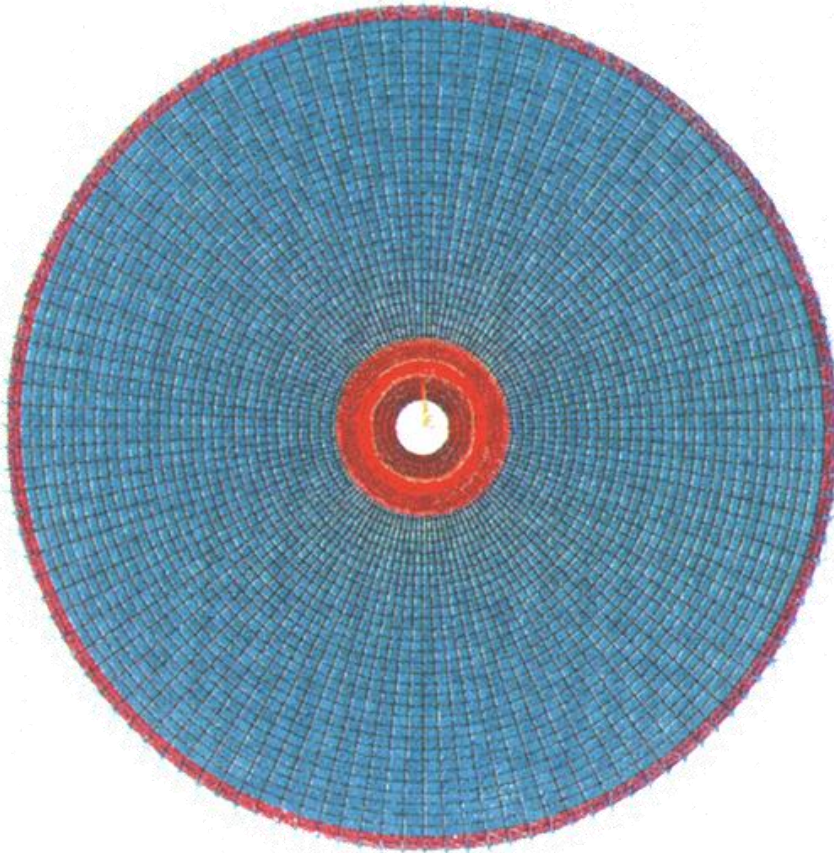
Modell: Elektromotor



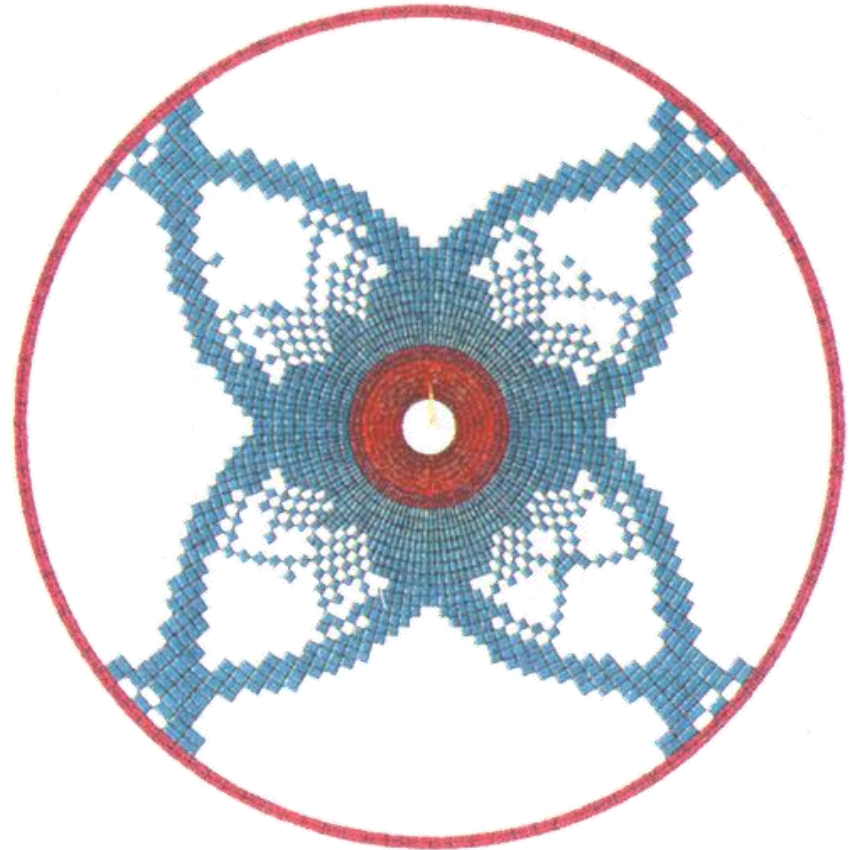
Lösung: Feldlinien



# Gestaltoptimierung



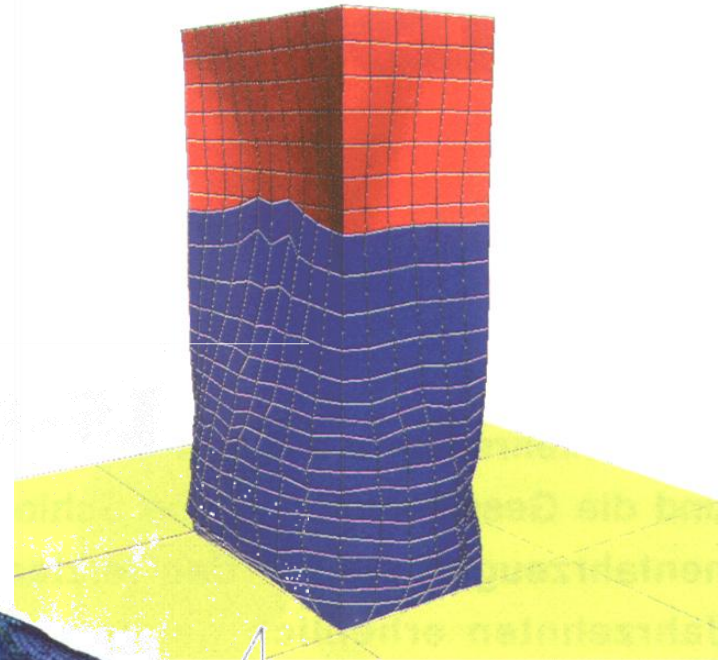
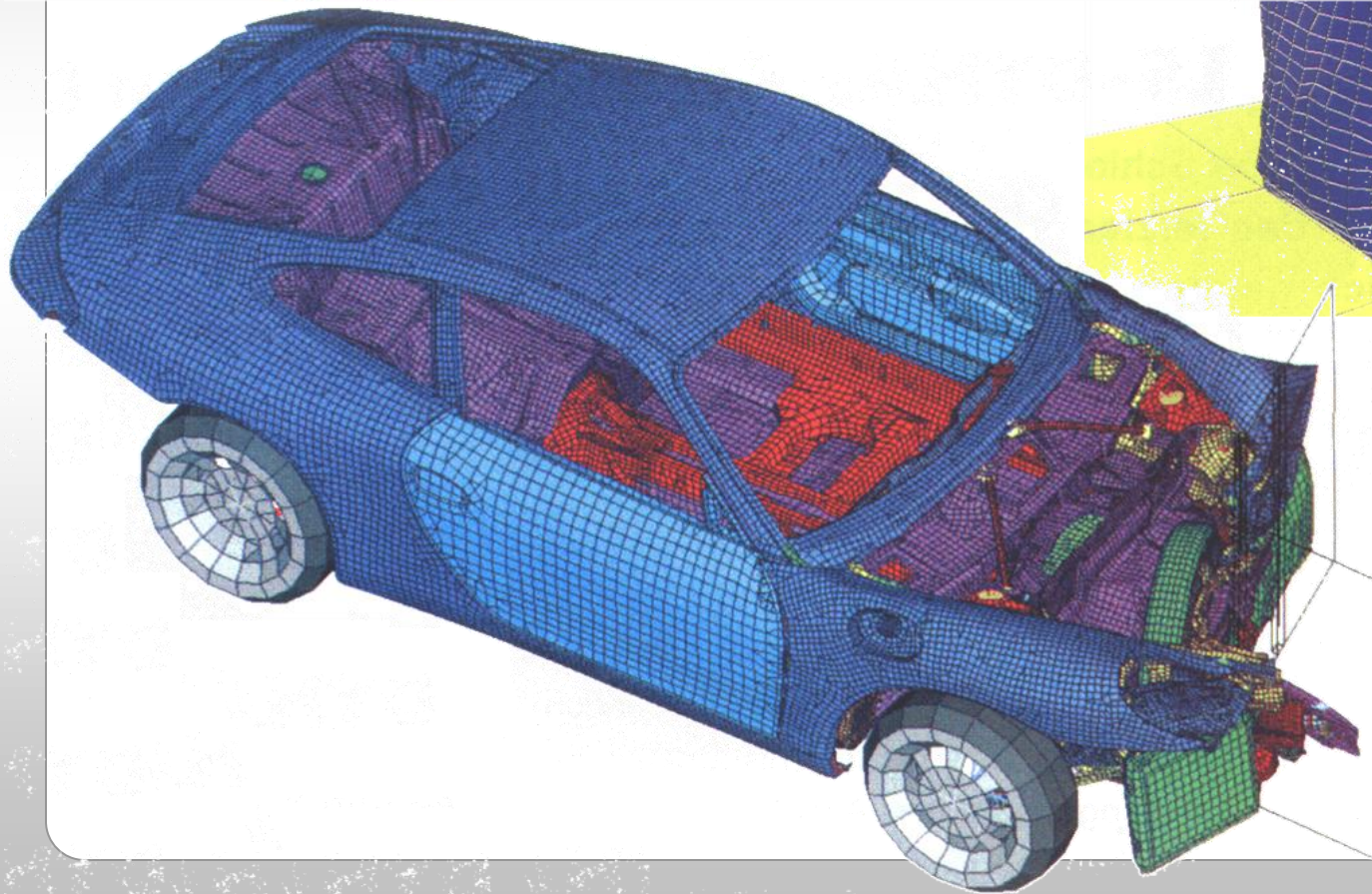
Anfangszustand: Vollscheibe



Endzustand: Speichen

# Explizite Dynamik

z.B. Stoßvorgänge

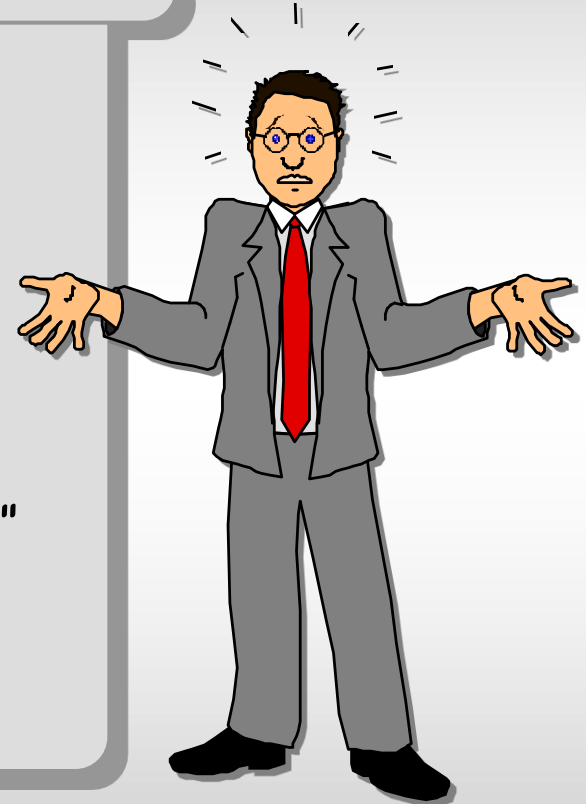


# Einführung in die Theorie der FEM

# FEM: Erklärung in einem Satz

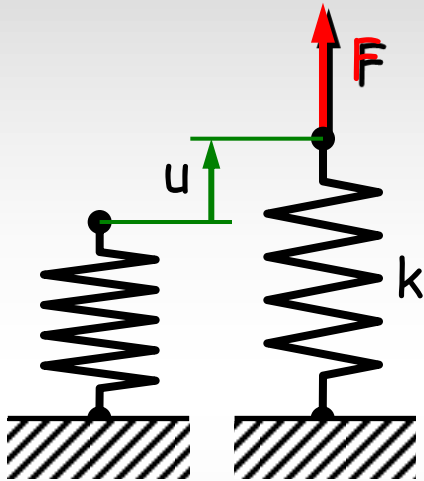
---

*Finite-Elemente-Methode  
=  
„Numerisches Verfahren  
zur näherungsweise Lösung von  
partiellen Differentialgleichungen“*



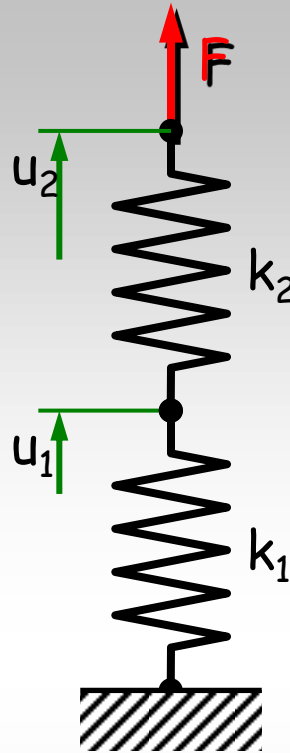


# FEM: Erklärung auf einer Folie



$$k \cdot u = F$$

$$u = k^{-1} F$$

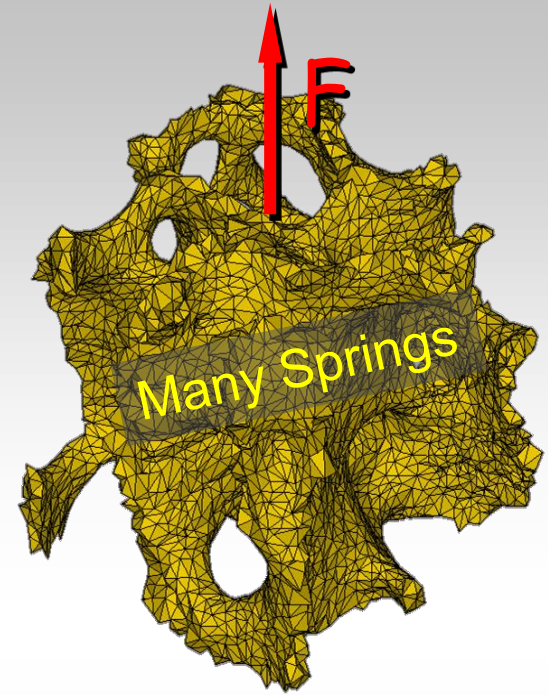


$$k_1 u_1 = k_2 (u_2 - u_1)$$

$$k_2 (u_2 - u_1) = F$$

$$\underbrace{\begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix}}_{\underline{\underline{K}}} \cdot \underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\underline{u}} = \underbrace{\begin{bmatrix} 0 \\ F \end{bmatrix}}_{\underline{F}}$$

$$\underline{u} = \underline{\underline{K}}^{-1} \underline{F}$$



FE-Software

$$\underline{\underline{K}} \cdot \underline{u} = \underline{F}$$

FE-Software

$$\underline{u} = \underline{\underline{K}}^{-1} \underline{F}$$

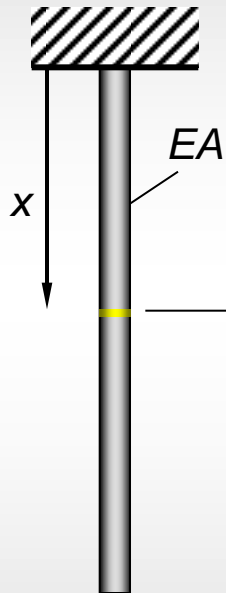


# FEM: Erklärung in 20 Minuten am supereinfachen Beispiel

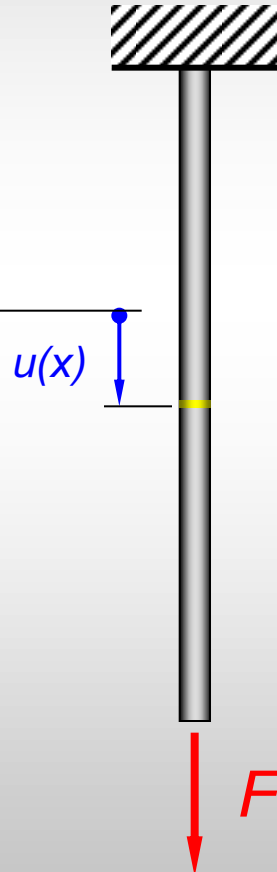
U. Simon, UZWR, Uni Ulm

# Beispiel: Dehnstab mit Endlast

Unbelastet:  
(Referenzzustand)



Belastet:



Gegeben:

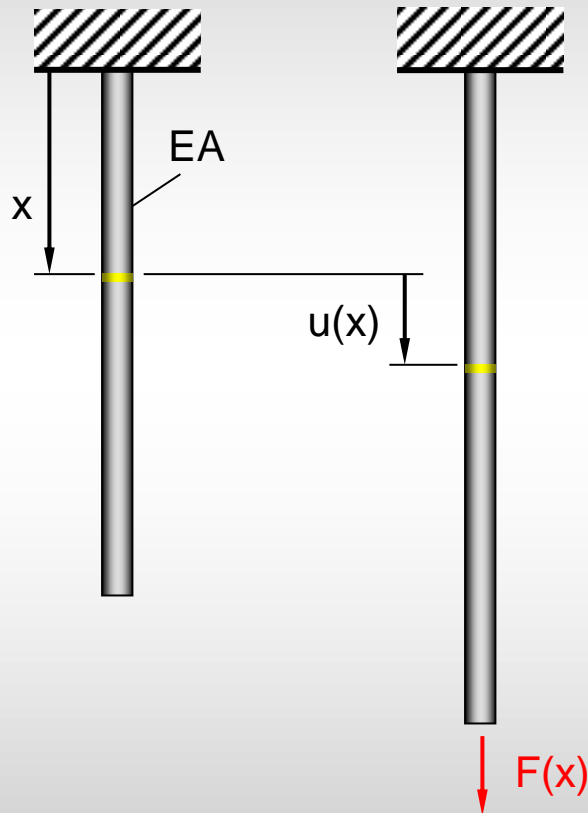
**Dehnstab** mit ...

- Länge  $L$ ,
- Querschnitt  $A$  (konstant),
- E-Modul  $E$  (Material homogen u. konstant),
- Länge  $L$  und
- Endlast  $F$  (axial),
- oben fest eingespannt.

Gesucht:

- **Verschiebungsfunktion**  $u(x)$
- Dehnungen  $\varepsilon(x)$
- Spannungen  $\sigma(x)$

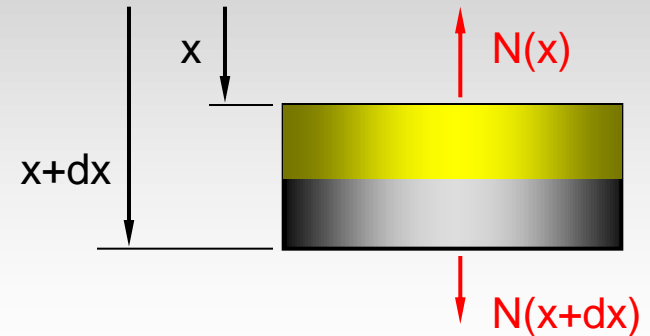
# A) Klassische Lösung (Methode der „infiniten“ Elemente)



Referenzlage

Belastet

**Differentielles Element**  
(infinitesimale Höhe  $dx$ )



## Aufstellen der Differentialgleichung

1. Kinematik:  $\varepsilon = u'$
2. Werkstoff:  $\sigma = E\varepsilon \Rightarrow N = EAu'$
3. Gleichgewicht:  $N' = 0$

$$\Rightarrow \text{DGL: } (EA u')' = 0$$

Annahme:  $EA = \text{const}$

$$u'' = 0$$

# A) Klassische Lösung (Methode der „infiniten“ Elemente)

## Lösen der Differentialgleichung

$$u''(x) = 0$$

2 mal integrieren:

$$u'(x) = C_1$$

$$u(x) = C_1 * x + C_2$$

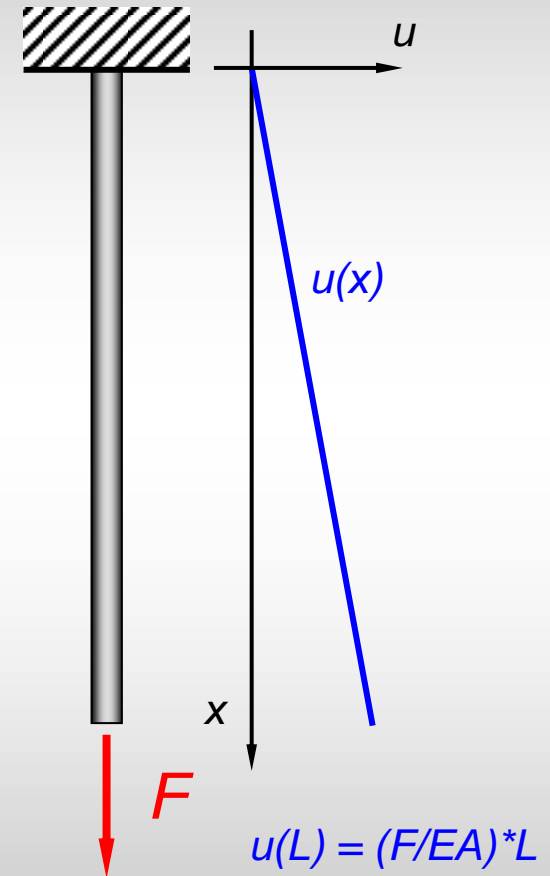
## Anpassen an Randbedingungen

Oben (Einspannung):  $u(0) = 0 \Rightarrow C_2 = 0$

Unten (Kraft):  $N(L) = F \Rightarrow u'(L) = F/(EA)$   
 $\Rightarrow C_1 = F/EA$

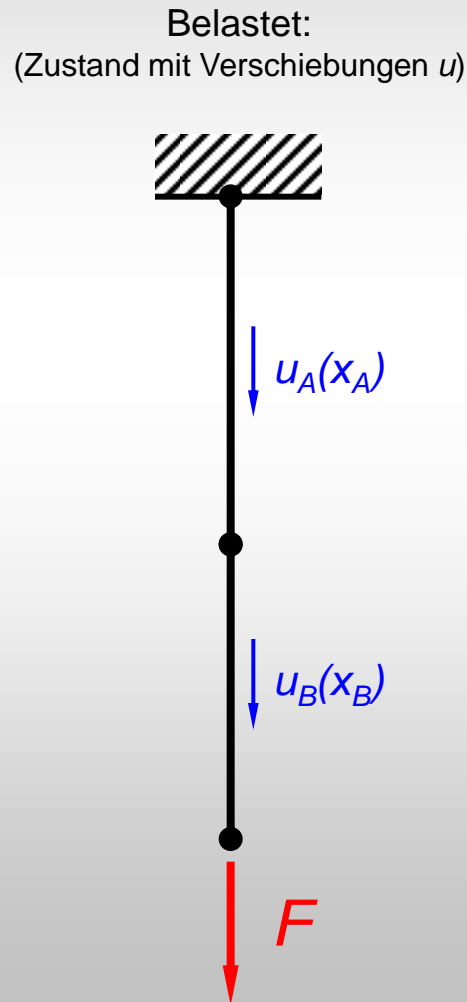
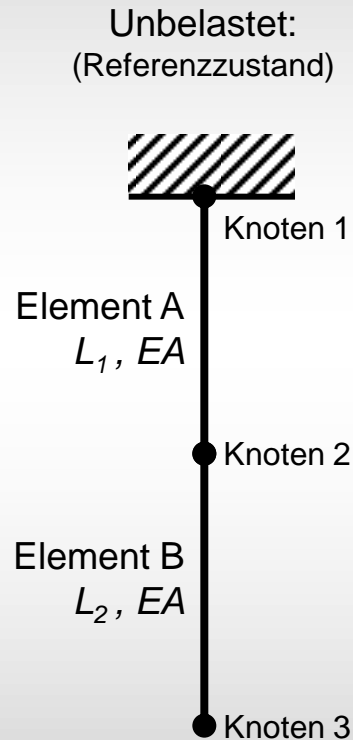
## Angepasste Lösung

$$u(x) = (F/EA) * x$$

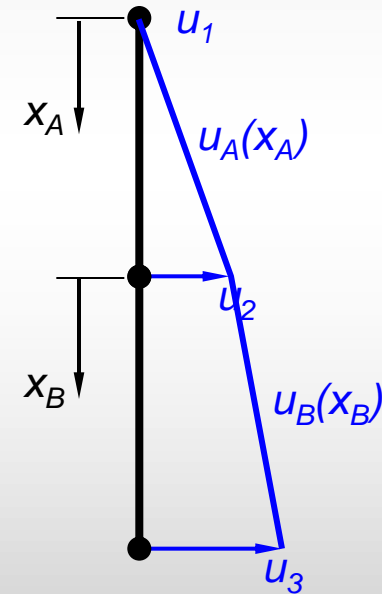


## B) Lösung mit FEM (Methode der Finiten Elemente)

**Diskretisierung:** Wir teilen das Gebiet in (nur) zwei finite (= endliche, nicht differentiell kleine) Elemente ein. Die Elemente sind an den Knoten verbunden.



**Ansatzfunktionen** (lineare) für die gesuchten Verschiebungen  $u$ .



Das gesuchte Verschiebungsfeld wird elementweise mit einfachen (linearen) **Ansatzfunktionen** beschrieben (siehe Bild). Das ist die **Grundidee** der FEM. Achtung: Die Verschiebungen  $u$  sind hier natürlich axial gerichtet, auch wenn es in den Ansatzfunktionen anders erscheint.

$$u_A(x_A) = \hat{u}_1 + (\hat{u}_2 - \hat{u}_1) \frac{x_A}{L_A} = \hat{u}_1 \left( 1 - \frac{x_A}{L_A} \right) + \hat{u}_2 \frac{x_A}{L_A}$$

$$u_B(x_B) = \hat{u}_2 + (\hat{u}_3 - \hat{u}_2) \frac{x_B}{L_B} = \hat{u}_2 \left( 1 - \frac{x_B}{L_B} \right) + \hat{u}_3 \frac{x_B}{L_B}$$

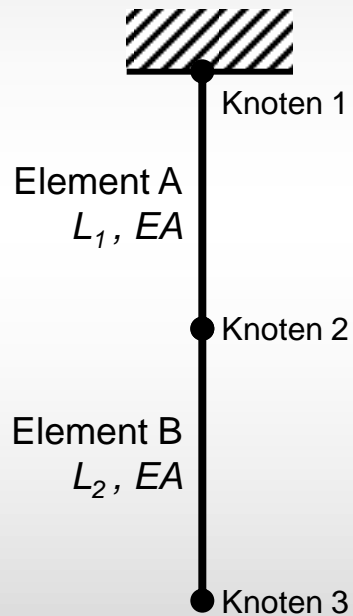
Es gibt jetzt als **Unbekannte nur noch die drei „Knotenverschiebungen“**  $\hat{u}_1, \hat{u}_2, \hat{u}_3$  und keine ganze Funktion  $u(x)$  mehr.

Dann führen wir so genannte „**virtuelle Verschiebungen**“ (**VV**) ein. Das sind zusätzliche, gedachte, kleine, beliebige Verschiebungen  $\delta \hat{u}_1, \delta \hat{u}_2, \delta \hat{u}_3$  konsistent zu Randbedg. Anschaulich: wir „wackeln“ ein bisschen an den Knoten.

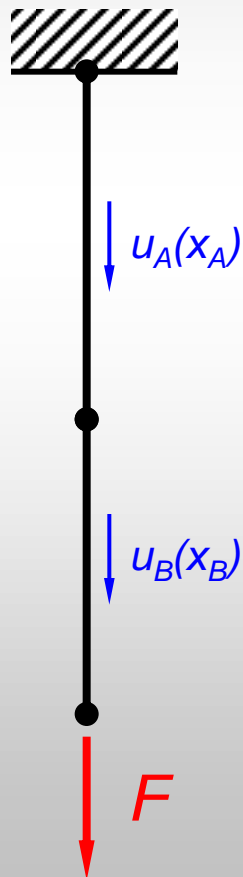
Nun gilt das **Prinzip der virtuellen Verschiebungen (PVV)**: Ein mechanisches System ist im Gleichgewicht, wenn die Virtuelle Gesamtarbeit  $\delta W$  (also virtuelle elastische minus virtuelle äußere Arbeit) gerade verschwindet. Die Arbeit heißt „virtuell“ weil sie nur der durch die VV verursachte (zusätzliche) Anteil ist.

$$\delta W = 0 \quad \Rightarrow \quad \delta W_{el} - \delta W_a = 0$$

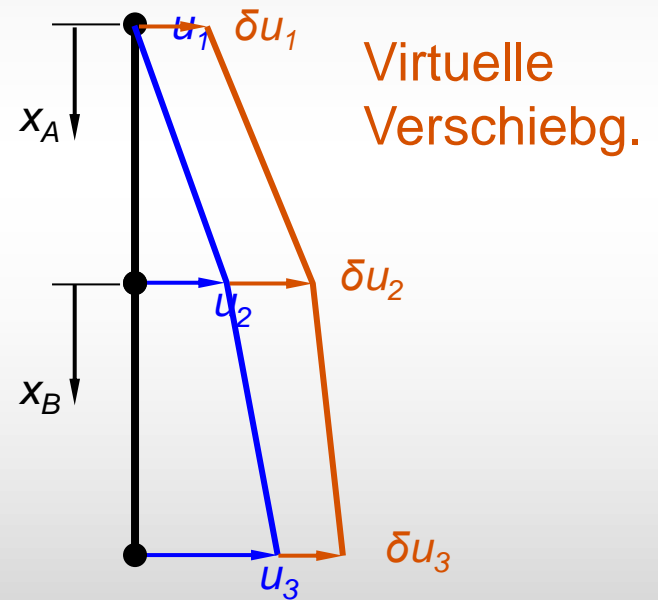
Unbelastet:  
(Referenzzustand)



Belastet:  
(Zustand mit Verschiebungen  $u$ )



**Ansatzfunktionen** (lineare) für die  
gesuchten Verschiebungen  $u$ .



Für unser einfaches Beispiel gilt:

Virt. elastische Arbeit = Normalkraft  $N$  mal VV

Virt. äußere Arbeit = äußere Kraft  $F$  mal VV

und die Normalkraft wiederum kann durch Ausdrücke  $EA/L$  mal Verschiebungsdifferenz ersetzt werden:

$$\begin{aligned}\delta W &= N_A(\delta \hat{u}_2 - \delta \hat{u}_1) + N_B(\delta \hat{u}_3 - \delta \hat{u}_2) - F \delta \hat{u}_3 \\ &= \frac{EA}{L_A}(\hat{u}_2 - \hat{u}_1)(\delta \hat{u}_2 - \delta \hat{u}_1) + \frac{EA}{L_B}(\hat{u}_3 - \hat{u}_2)(\delta \hat{u}_3 - \delta \hat{u}_2) - F \delta \hat{u}_3 \\ \delta W &= \delta \hat{u}_1 \left( + \frac{EA}{L_A} \hat{u}_1 - \frac{EA}{L_A} \hat{u}_2 \right) \\ &\quad + \delta \hat{u}_2 \left( - \frac{EA}{L_A} \hat{u}_1 + \frac{EA}{L_A} \hat{u}_2 + \frac{EA}{L_B} \hat{u}_2 - \frac{EA}{L_B} \hat{u}_3 \right) \\ &\quad + \delta \hat{u}_3 \left( - \frac{EA}{L_B} \hat{u}_2 + \frac{EA}{L_B} \hat{u}_3 - F \right) = 0\end{aligned}$$

Wir haben mit diesem Prinzip aber leider nur eine Gleichung für die drei unbekannten Knotenverschiebungen  $\hat{u}_1, \hat{u}_2, \hat{u}_3$ . **Schade!** Aber da gibt es einen Trick ...



Abgekürzt schreiben wir:

$$\delta \hat{u}_1(\dots)_1 + \delta \hat{u}_2(\dots)_2 + \delta \hat{u}_3(\dots)_3 = 0$$

Die **VV** dürfen ja **unabhängig von einander** gewählt werden. Also z.B. alle Null nur immer eine (egal welche) ungleich Null. Dann muss die eine Klammer, bei der wir die VV nicht Null hatten gleich Null sein, damit die Gleichung gelten kann. Weil wir aber die VV wählen können wie wir wollen und auch eine andere VV ungleich Null hätten wählen dürften, müssen nun alle drei Klammern einzeln gleich Null sein, damit die Gleichung oben dennoch immer erfüllt ist. Wir bekommen damit drei Gleichungen. **Juhu!**

$$(\dots)_1 = 0; \quad (\dots)_2 = 0; \quad (\dots)_3 = 0$$

... die wir auch in Matrizenform hinschreiben können:

$$\begin{bmatrix} \frac{EA}{L_1} & -\frac{EA}{L_1} & 0 \\ -\frac{EA}{L_1} & \frac{EA}{L_1} + \frac{EA}{L_2} & -\frac{EA}{L_2} \\ 0 & -\frac{EA}{L_2} & \frac{EA}{L_2} \end{bmatrix} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \hat{u}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix}$$

oder kurz:

$$\underline{\underline{K}} \underline{\hat{u}} = \underline{\underline{F}}$$

$\underline{\underline{K}}$  - Gesamt-Steifigkeitsmatrix  
 $\underline{\hat{u}}$  - Vektor der unbekannten Knotenverschiebungen  
 $\underline{\underline{F}}$  - Vektor der Knotenkräfte

Dieses ist die klassische Gleichung, auf die eine lineare strukturmechanische FE-Analyse zunächst hinausläuft. Ein **lineares Gleichungssystem** für die unbekannten Knotenverschiebungen  $u$ .

Wir müssen noch die **Randbedingungen** berücksichtigen. Der Stab ist am oberen Ende fest eingespannt. Knoten 1 kann sich nicht verschieben:

$$\hat{u}_1 = 0$$

Damit müssen wir die erste Zeile im Gleichungssystem streichen, denn hier wurde die Bedingung das die VV konsistent zu den Rbedg. sein müssen verletzt. Die erste Spalte können wir ebenfalls streichen, denn diese Elemente würden ja ohnehin mit einer Null multipliziert werden. Es bleibt also ein 2x2-System ...

$$\begin{bmatrix} \frac{EA}{L_1} + \frac{EA}{L_2} & -\frac{EA}{L_2} \\ -\frac{EA}{L_2} & \frac{EA}{L_2} \end{bmatrix} \begin{bmatrix} \hat{u}_2 \\ \hat{u}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ F \end{bmatrix}$$

Wir **lösen das Gleichungssystem** und erhalten die Knotenverschiebungen

$$\hat{u}_2 = \frac{L_A}{EA} F \quad \text{und} \quad \hat{u}_3 = \frac{L_A + L_B}{EA} F$$

Hier entspricht die FE-Lösung exakt der (bekannten!) analytischen Lösung (vgl. Teil A). Das gilt bei komplizierteren Fällen nicht mehr. Gewöhnlich gilt dann aber, dass sich die numerische Lösung sich mit zunehmender Anzahl von finiten Elementen immer besser an die exakte Lösung annähert. Bei komplexer Geometrie gibt es i.d.R. gar keine analytische Lösung mehr; in solchen Fällen geht es ausschließlich numerisch!

Aus den Knotenverschiebungen kann man in einer Nachlaufrechnung durch differenzieren die Dehnungen und mit bekanntem Werkstoff auch die Spannungen bestimmen.

$$\varepsilon_A(x_A) = \frac{\hat{u}_2 - \hat{u}_1}{L_A}$$

Dehnungen

$$\varepsilon_B(x_B) = \frac{\hat{u}_3 - \hat{u}_2}{L_B}$$

$$\sigma_A(x_A) = E \frac{\hat{u}_2 - \hat{u}_1}{L_A}$$

Spannungen

$$\sigma_B(x_B) = E \frac{\hat{u}_3 - \hat{u}_2}{L_B}$$

Fertig!

# Zusammenfassung

Die **wesentlichen Schritte** und Ideen der FEM sind also:

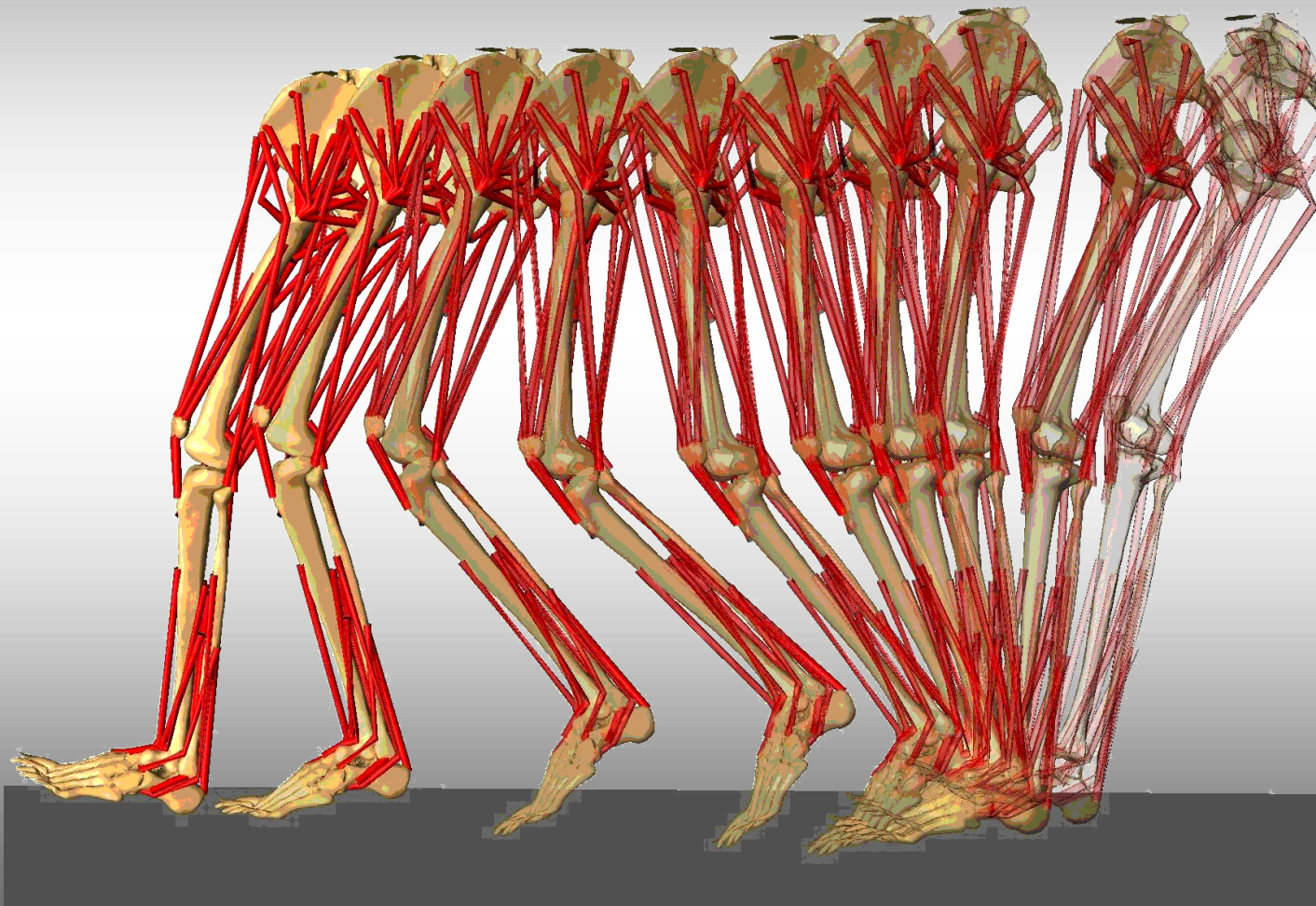
- Diskretisierung: Einteilung des Gebiets in finite Elemente
- Einfache Ansatzfunktionen (Polynome) für die gesuchte Größe innerhalb der Elemente wählen. Man reduziert das Problem auf endlich viele Unbekannte.
- Mechanisches Prinzip (hier PVV) anschreiben und
- daraus Gleichungssystem für unbekannte Knotengrößen ableiten
- Gleichungssystem lösen

## **Bemerkungen:**

Viele der hier gezeigten Schritte spielen bei der Benutzung eines kommerziellen FE-Programms keine Rolle, sondern eher schon bei dessen Programmierung, wie das hier gewählte mechanische Prinzip (PVV), das die Struktur für den Aufbau des Gesamt-Gleichungssystems liefert. Während einer FE-Analyse selbst wird kein PVV oder ähnliches ausgeführt.

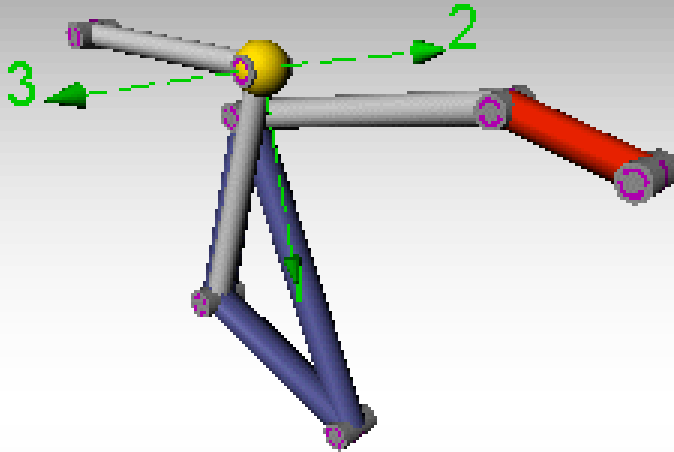
Für ein 3D Problem würden selbst bei aller einfachster Wahl des Werkstoffs zwei Werkstoffparameter (z.B.  $E$  und  $\nu$ )

# Mehrkörper-Systeme (MKS) in der Biomechanik



# Mehrkörpersysteme (MKS):

---



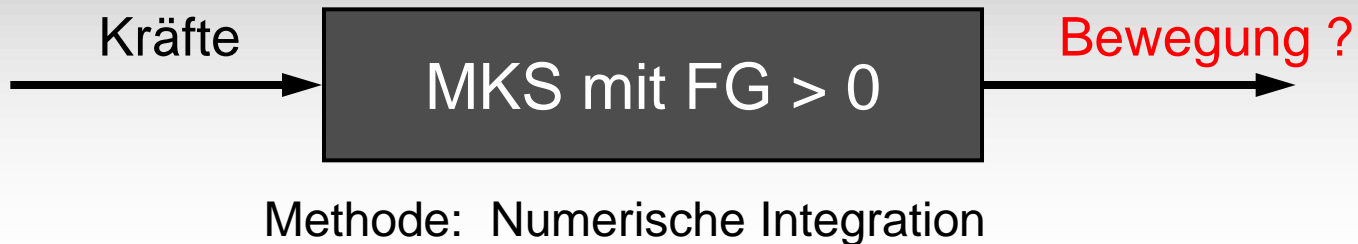
Quelle: IFM, TU Chemnitz, alaska

- Starrkörpern (mit Massen, Massenmomenten)
- Gelenken (Bindungen)
- $FG \text{ Gesamtsystem} = \text{Anzahl Körper} \times 6 - \text{Anzahl Bindungen}$

# VERGLEICH: Vorwärts-Dynamik / Inverse Dynamik

---

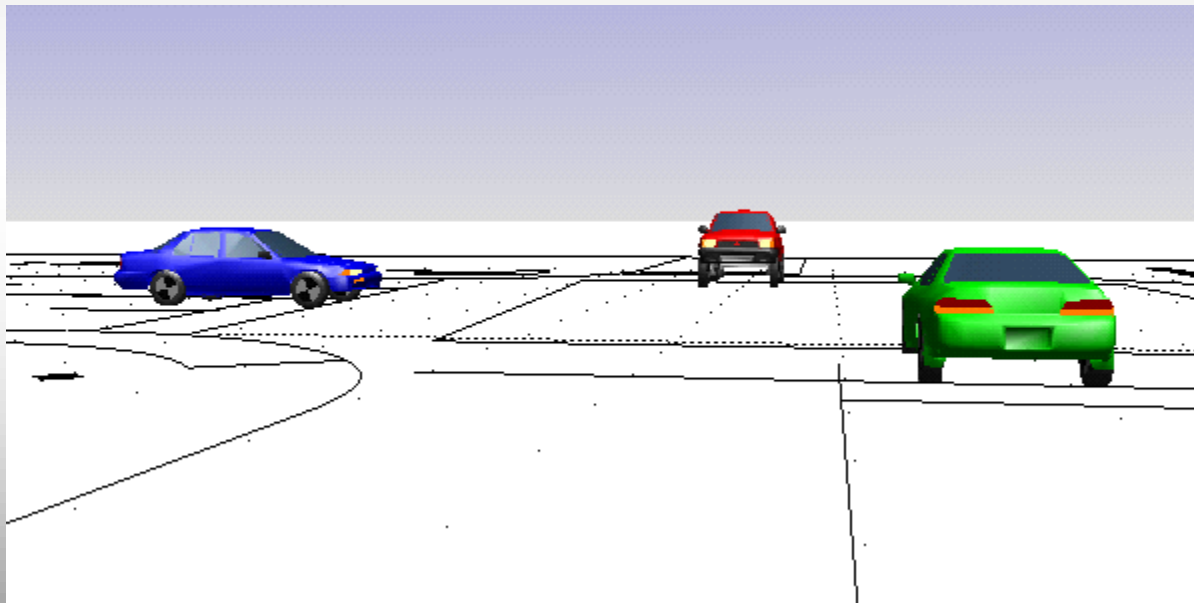
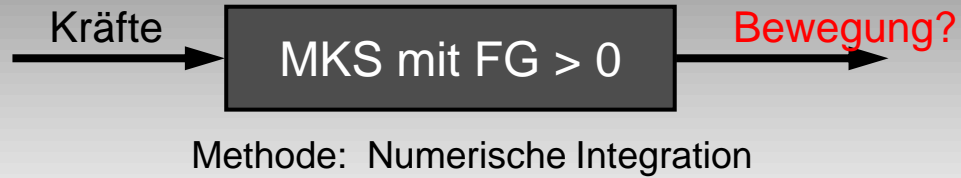
## Vorwärts-Dynamik



## Inverse Dynamik

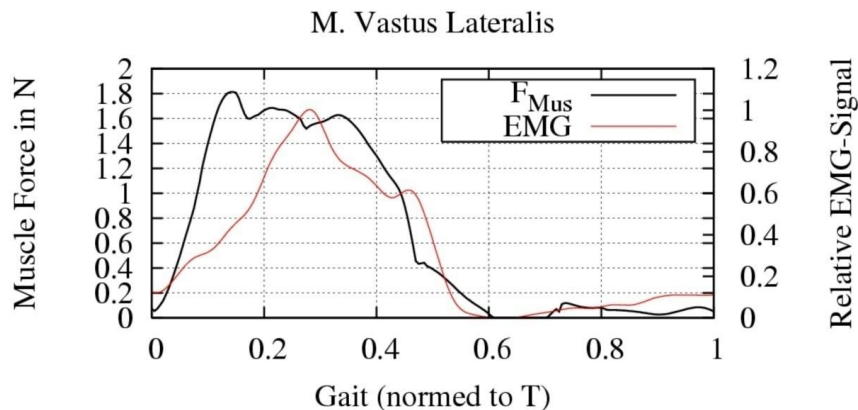


# Beispiel zur Vorwärts-Dynamik: “Unfallhergang”





# Beispiel zur Inversen Dynamik: „Ratten-Gang“



UFBSIM

Erik Forster  
Uwe Wolfram

# Software

Numerische Methoden, allgemein:	
MATLAB	Universelles Numerik-Programm (eigene Programmiersprache) <a href="http://www.mathworks.com">http://www.mathworks.com</a>
NAG	Große Sammlung sehr leistungsstarker Routinen für Fortran oder C <a href="http://www.nag.co.uk">http://www.nag.co.uk</a> (Vorcompilierte Object-Libraries)
Numerical Recipes	Große Sammlung leistungsstarker Routinen für Fortran, C, C++ <a href="http://www.nag.co.uk">http://www.nag.co.uk</a> (Bücher und Quellcode)
Finite-Elemente-Programme:	
ABAQUS	Universelles, professionelles FE-Software-Paket <a href="http://www.abaqus.com">http://www.abaqus.com</a>
ANSYS	Universelles, professionelles FE-Software-Paket <a href="http://www.ansys.com">http://www.ansys.com</a> Aktuelle Hochschulversion (128.000 Elemente): 7.0 Deutscher Vertrieb: Fa. CADFEM, <a href="http://www.cadfem.de">http://www.cadfem.de</a>
MSC.Patran/Nastran MSC.Marc/Mentat	Universelles, professionelles FE-Software-Paket <a href="http://www.mscsoftware.com">http://www.mscsoftware.com</a>
CAMMPUS	Für Lehre frei verfügbares, kleines FE-Programm von Prof. Dankert, FH Hamburg. Für Fachwerksberechnungen. Mit Quellcode FEMSET. <a href="http://www.fh-hamburg.de/rzbt/dnksoft/cammpus">http://www.fh-hamburg.de/rzbt/dnksoft/cammpus</a>
MATLAB (PDE)	PDE Toolbox: FE für 2D Probleme <a href="http://www.mathworks.com">http://www.mathworks.com</a>

# Software

## Fuzzy-Logic-Programme:

MATLAB	Fuzzy-Logic-Toolbox: leicht zu bedienen <a href="http://www.mathworks.com">http://www.mathworks.com</a>
--------	--

## Mehr-Körper-Simulations-Programme (MKS):

MSC.ADAMS	Universelles, professionelles MKS-Software-Paket <a href="http://www.mscsoftware.com">http://www.mscsoftware.com</a>
DADS	Universelles, professionelles MKS-Software-Paket <a href="http://www.lmsintl.com/">http://www.lmsintl.com/</a>
MADYMO	Software-Paket zur Insassensicherheit, „Virtuelle-Dummies“ <a href="http://www.madymo.com">http://www.madymo.com</a> (TNO, Niederlande)
alaska	Modellierung und Simulation mechatronischer Systeme <a href="http://www.tu-chemnitz.de/ifm">http://www.tu-chemnitz.de/ifm</a> (Institut für Mechatronic, TU Chemnitz)
SIMPACT	Universelles, professionelles MKS-Software-Paket, Intec GmbH (DLR) <a href="http://www.simpact.de">http://www.simpact.de</a>

## Programme zur Inversen Dynamik (Ganganalyse):

ANYBODY	Simulation muskulo-skelettaler Systeme, Aalborg/Dänemark <a href="http://anybody.auc.dk">http://anybody.auc.dk</a>
UFBSIM	Simulation muskulo-skelettaler Systeme (von Erik Forster), Institut für Unfallchirurgische Forschung und Biomechanik, Universität Ulm <a href="http://www.biomechanics.de">http://www.biomechanics.de</a> , <a href="mailto:erik.forster@medizin.uni-ulm.de">erik.forster@medizin.uni-ulm.de</a>

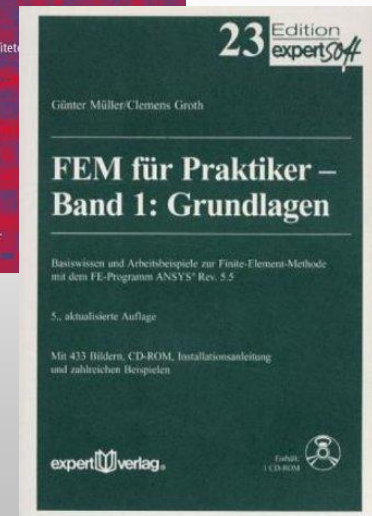
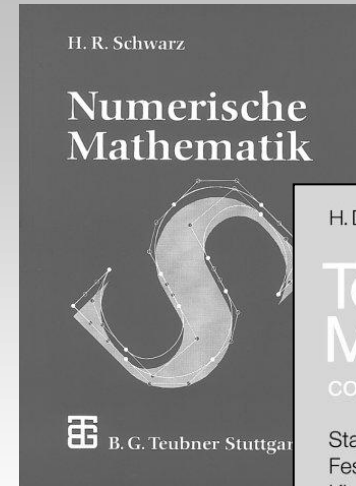
# Literatur

## Allgemein

- Schwarz, H.R.: „Numerische Mathematik“, Teubner 1997  
*Klassiker der numerischen Mathematik, 37,- €*
- Dankert, H. und Dankert, J.: „Technische Mechanik“; Statik, Festigkeitslehre, Kinematik/Kinetik, mit Programmen; 2. Aufl.; Teubner, 1995.  
*Lehrbuch mit numerischer Anwendung u.a. FEM*  
<http://www.fh-hamburg.de/rzbt/dnksoft/campus/>  
*(Scripte, auch „Num. Methoden“ zum Herunterladen)*

## Finite-Elemente-Methode (FEM):

- Zienkiewicz, O.C.: „Methode der finiten Elemente“; Hanser 1975 (engl. 2000).  
*„Bibel“ der FEM*
- Bathe, K.-J.: „Finite-Elemente-Methoden“; erw. 2. Aufl.; Springer 2001  
*Lehrbuch (Theorie) , 100,- €*
- Müller, G. und Groth, C.: „FEM für Praktiker, Band 1: Grundlagen“, mit ANSYS/ED-Testversion (CD). (Band 2: Strukturdynamik; Band 3: Temperaturfelder)  
*Übungsbuch, sehr gut für den Einstieg, 98,- €*



# Literatur

## Fuzzy-Logic:

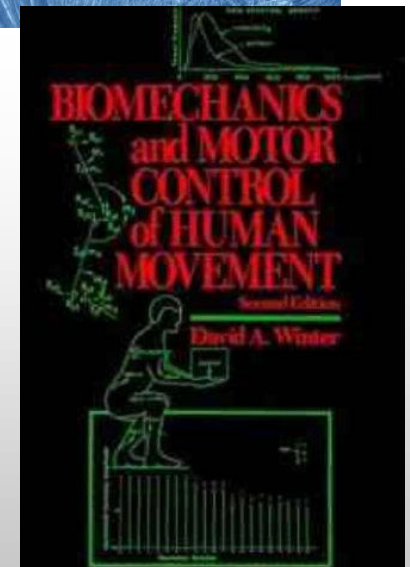
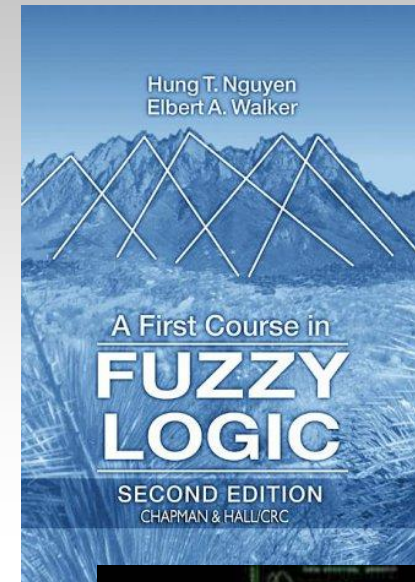
- Drösser, D.: „*Fuzzy Logic – Methodische Einführung in krauses Denken*“; rororo; 1996.  
Taschenbuch, gut zu lesen
- Hung T. u.a.: „*A First Course in Fuzzy Logic*“; 2. Aufl.; CRC Press; 1999.  
80,- €

## Mehr-Körper-Simulation (MKS):

- Haug, E.J.: „*Computer aided kinematics and dynamics of mechanical systems*“; 1989.  
Vom DADS-Entwickler geschrieben, sehr gut
- Nikravesh, P.E.: „*Computer aided analysis of mechanical systems*“  
Prentice-Hall; 1988.

## Inverse Dynamik (Ganganalyse):

- Nigg, B.M., Herzog, W.: „*Biomechanics of the Musculo-skeletal System*“; Wiley, 1994.
- Winter, D.A.: „*Biomechanics and motor control of human movement*“; Wiley, 1990.



# Hinweise und Warnungen

- Numerische Methoden (Programm-Pakete) sind Werkzeuge, keine Lösungen
  - Vorsicht vor bunten Bildern. Es gilt wie überall: „garbage in - garbage out“ (GiGo).
  - Systemparameter
  - Verifikation
  - Modelle sind immer fragespezifisch
- } erfordern Experimente

