4 Modellierung und Analyse kontinuierlicher Systeme

Bisher wurden ereignisdiskrete Systeme analysiert

Also Systeme, bei denen Zustandsänderungen nur zu Ereigniszeitpunkten auftreten

Solche Systeme sind insbesondere im technischen Bereich zu finden, wenn einzelne Individuen (Kunden, Teile,, ...) betrachtet werden

Andere Systeme sind gekennzeichnet durch kontinuierliche Änderungen der Zustandsvariablen oder durch die aggregierte Betrachtung diskreter Vorgänge (Übergänge sind fließend)

$\Rightarrow$ **kontinuierliche Simulation**

Basismodell Differentialgleichungssysteme

In manchen Bereichen wird der Begriff Simulation (fälschlicherweise) mit der numerischen Lösung von Differentialgleichungen gleichgesetzt

Ziele:

- Basismodelle der kontinuierlichen Simulation kennen lernen
- Erkennen, wann kontinuierliche und wann diskrete Simulation eingesetzt werden soll/kann
- Überblick über typische Anwendungsgebiete der kontinuierlichen Simulation bekommen
- Beschreibungstechniken und –sprachen für kontinuierliche Systeme kennen lernen
- Grenzen der kontinuierlichen Simulation erkennen
- Basismethoden zur numerischen Analyse kontinuierlicher Simulationsmodelle kennen lernen

Gliederung

4.1 Aufbau kontinuierlicher Simulationsmodelle

4.2 Analysetechniken für kontinuierliche Modelle

In diesem Semester aus Zeitgründen nur eine kurze Zusammenfassung

Literatur:

Viele Bücher zu speziellen Aspekten/Anwendungsgebieten der kontinuierlichen Simulation, aber kein kompaktes Lehrbuch als Einführung verfügbar.

Am ehesten geeignet:

- F. Cellier. Continuous System Modeling, Springer 1991 (750 S.)
  sehr umfangreich und umfassend, aber Teil über numerische Lösungsmethoden fehlt, Modellierungswerkzeuge & Programmiersprachen z.T. veraltet

Weitere Quellen:

- Skript Simulationstechnik Universität Siegen, Prof. Wiechert
- H. Bossel: Simulation dynamischer Systeme, Vieweg 1982
- diverse Lehrbücher der angewandten Mathematik zur Lösung von DGLs
- diverse Lehrbücher aus den Anwendungsgebieten zur Modellierung mit DGLs

4.1 Aufbau kontinuierlicher Simulationsmodelle

Beschreibung kontinuierlicher Vorgänge in der Natur und Technik

Ebenen der Modellierung

- Deduktive Modellierung und white-box-Modelle
  - Elektrotechnik, Mechanik, ...
    Abläufe sind gut verstanden und können durch Differentialgleichungen exakt beschrieben werden
- Mischung aus induktiver und deduktiver Modellierung und grey-box-Modelle
  - Biologie, Chemie, ...
    Abläufe sind nur zum Teil detailliert erklärbar/herleitbar, gewisse Einflüsse müssen durch die Kalibrierung der Modelle induktiv ergänzt werden
- Rein induktive Modellierung und black-box-Modelle
  - Sozialwissenschaften, Weltmodelle, ...
    Beobachtung der Realität liefert Zusammenhang zwischen Eingaben C, U und Ausgaben P, Zusammenhang wird durch mathematische Funktion dargestellt

Verwandte Modelltypen:

- Algebraische Gleichungssysteme
  - statische Beschreibung von black-box-Modellen
    Zitat aus Cellier 1991: *„Finally, in the darkest of all worlds, i.e., in social and psychological modeling, the models used are mostly static. They are described by algebraic equations (AEs). They are usually entirely inductive on „gut feeling“ or the position of the stars in the sky“.*
- Differenzengleichungen
  - dynamische Beschreibung von grey-box-Modellen
    Bei ungenauen Modellen, ist eine detaillierte Beschreibung durch Differentialgleichungen unnötig, es reichen einfache Differenzengleichungen zur Beschreibung der Systemdynamik in diskreten Schritten (die Qualität heutiger Löser für Differentialgleichungen macht den Einsatz von Differenzengleichungen oft unnötig)
- Gewöhnliche Differentialgleichungen
  - übliches Modell zur Beschreibung des Verhaltens, wenn nur der zeitliche Ablauf, nicht aber die räumliche Ausdehnung eine Rolle spielen
- Partielle Differentialgleichungen
  - zusätzliche Einbeziehung räumlicher Aspekte, führt oft zu aufwändigen Lösungen

Etwas detailliertere Darstellung eines (kontinuierlichen) Modells

$\mathbf{x}(0)$ Startvektor

Eingaben

kontrollierbar

$\mathbf{c}(t)$

$\mathbf{u}(t)$

unkontrollierbar

System S mit Zustand $\mathbf{x}(t)$

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{c}, \mathbf{u}, t)$$

$$\mathbf{p} = g(\mathbf{x}, \mathbf{c}, \mathbf{u}, t)$$

Ausgaben

beobachtbar

$\mathbf{p}(t)$

Oft zusammengefasst in einem Vektor, der mit **u**(t) bezeichnet wird

Oft wird **x**(t) direkt beobachtet, so dass g(.) wegfällt

<u>Bemerkungen</u>

- In manchen Darstellungen wird ein Parametervektor von *f* (.) und *g*(.) mit einbezogen, wir gehen davon aus, dass die Parameter mit dem Modell festliegen
- Die Schreibweise $\dot{\mathbf{x}}$ statt $\mathbf{x}'$ wird verwendet, wenn nach der Zeit t abgeleitet wird
- Wir betrachten im Folgenden die einfache Variante der Darstellung mit einem Eingabevektor **u**(t), Ausgaben **x**(t) und Systemfunktion *f*(**x**,**u**,t) oder teilweise auch nur *f*(**x**,t)

Analyse kontinuierlicher Modelle

Zu berechnen ist **x**(T) (Zustand des Modells zum Zeitpunkt T ≥ 0)

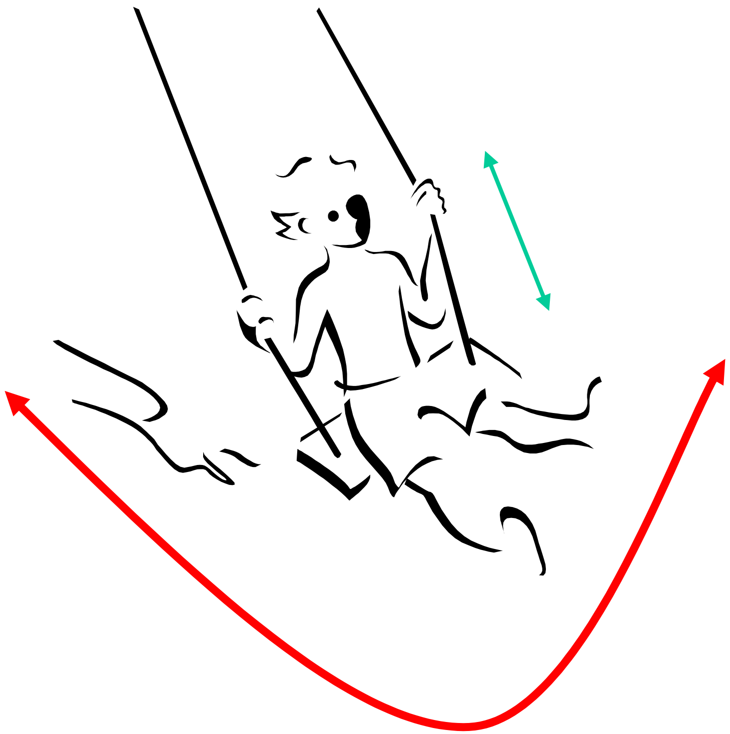
ausgehend vom Startwert **x**(0), der Kenntnis des Verlaufs von **u**(t) im Intervall [0,T] und der Gleichung $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t)$

Es gilt: $$\int_0^T \dot{\mathbf{x}}(t)dt = \mathbf{x}(T) - \mathbf{x}(0) \Rightarrow \mathbf{x}(T) = \int_0^T f(\mathbf{x}(t), \mathbf{u}(t), t)dt + \mathbf{x}(0)$$

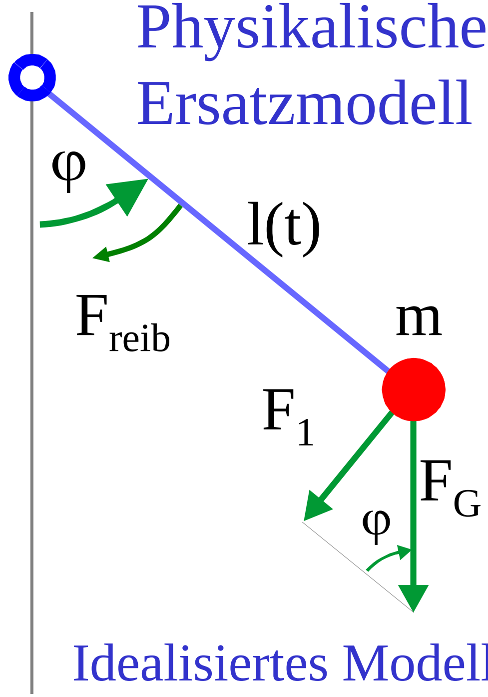
<u>Analyse</u>:

- Falls die Stammfunktion von f bekannt, so ist **x**(T) analytisch berechenbar (dies ist die Ausnahme!)
- Ansonsten muss **x**(T) numerisch berechnet werden.
  Dazu wird (mehr oder weniger detailliert/geschickt) aus bekannten Werten (in Gegenwart und Vergangenheit) ein Wert in der Zukunft (numerisch) berechnet.
  Also Nachspielen der Dynamik in diskreten Schritten (**Simulation**)

Beispiel: Schaukel (aus einem Skript von Prof. Wiechert, Uni Siegen)



(Vereinfachtes) Physikalisches Ersatzmodell



Physikalischer Vorgang, in den zahlreiche Parameter einfließen, z.B.

- Gewicht und Form des/der Schaukelnden
- Materialeigenschaften der Schaukel
- Aufhängung der Schaukel

Idealisiertes Modell:

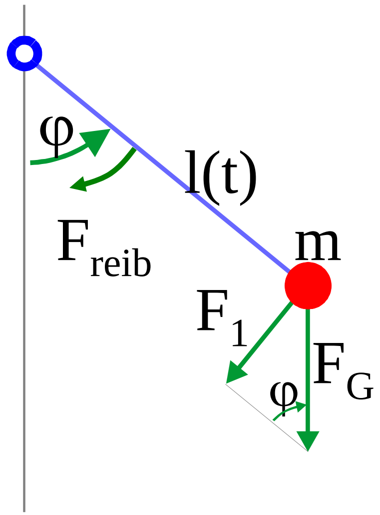
- so einfach wie möglich
- so realitätsnah wie nötig

immer unter Berücksichtigung der Zielstellung (hier Nachbildung der Position der Schaukel)

Vereinfachende Annahmen:

- Masse m ist punktförmig am Ende der Schaukel (dadurch entsteht eine gewöhnliche Diff.Gl.)
- Massepunkt ändert sich mit der Bewegung des Schaukelnden (dargestellt durch Länge l(t))
- Reibung hängt linear von der Geschwindigkeit ab (eigentlich komplexer nichtlinearer Vorgang)

Annahmen müssten eigentlich validiert werden!



Mathematisches Modell:

Gewichtskraft: $F_G = m \cdot g$ (g ist die Erdbeschleunigung)

Nach dem Kräfteparallelogramm: $F_1(t) = \sin(\varphi) \cdot F_G$

Winkelgeschwindigkeit: $\omega = \frac{d\varphi}{dt}$

Reibungskraft: $F_{reib} = -d \cdot \omega$

Trägheitsmoment: $J = m \cdot l^2(t)$

Drehmoment: $\tau = - l(t) \cdot F_1$

Gesetz von Newton: $\frac{d(J\omega)}{dt} = \tau + F_{reib}$

Weiterhin gilt: $\frac{d(J\omega)}{dt} = \dot{J} \cdot \omega + J \cdot \dot{\omega}$ und $\dot{J} = 2 \cdot m \cdot l \cdot \dot{l}$

Eingesetzt ergibt sich:

$$\frac{d(J\omega)}{dt} = 2 \cdot m \cdot l \cdot \dot{l} \cdot \omega + m \cdot l^2 \cdot \dot{\omega} = m \cdot l \cdot \left(2 \cdot \dot{l} \cdot \omega + l \cdot \dot{\omega}\right)$$

Herleitung einer Differentialgleichung in $\dot{\omega}$:

$$\begin{aligned}
m \cdot l \cdot \left(2 \cdot \dot{l} \cdot \omega + l \cdot \dot{\omega}\right) &= \tau + F_{reib} && \Leftrightarrow \\
2 \cdot \dot{l} \cdot \omega + l \cdot \dot{\omega} &= \frac{F_{reib}}{m \cdot l} - \frac{F_1}{m} && \Leftrightarrow \\
\dot{\omega} &= \frac{1}{m \cdot l} \cdot \left(\frac{F_{reib}}{l} - F_1 \right) - 2 \cdot \omega \cdot \frac{\dot{l}}{l}
\end{aligned}$$

Differentialgleichungssystem zur Beschreibung des Modells

$$\dot{\omega} = \frac{1}{m \cdot l} \cdot \left(\frac{F_{reib}}{l} - F_1 \right) - 2 \cdot \omega \cdot \frac{ld}{l} \qquad \dot{\varphi} = \omega$$

$$F_1 = \sin(\varphi) \cdot m \cdot g \qquad F_{reib} = -d \cdot \omega$$

$$\dot{l} = ld$$

Differentialgleichung mit den Variablen ω und φ und den Hilfsvariablen

Hilfsvariablen

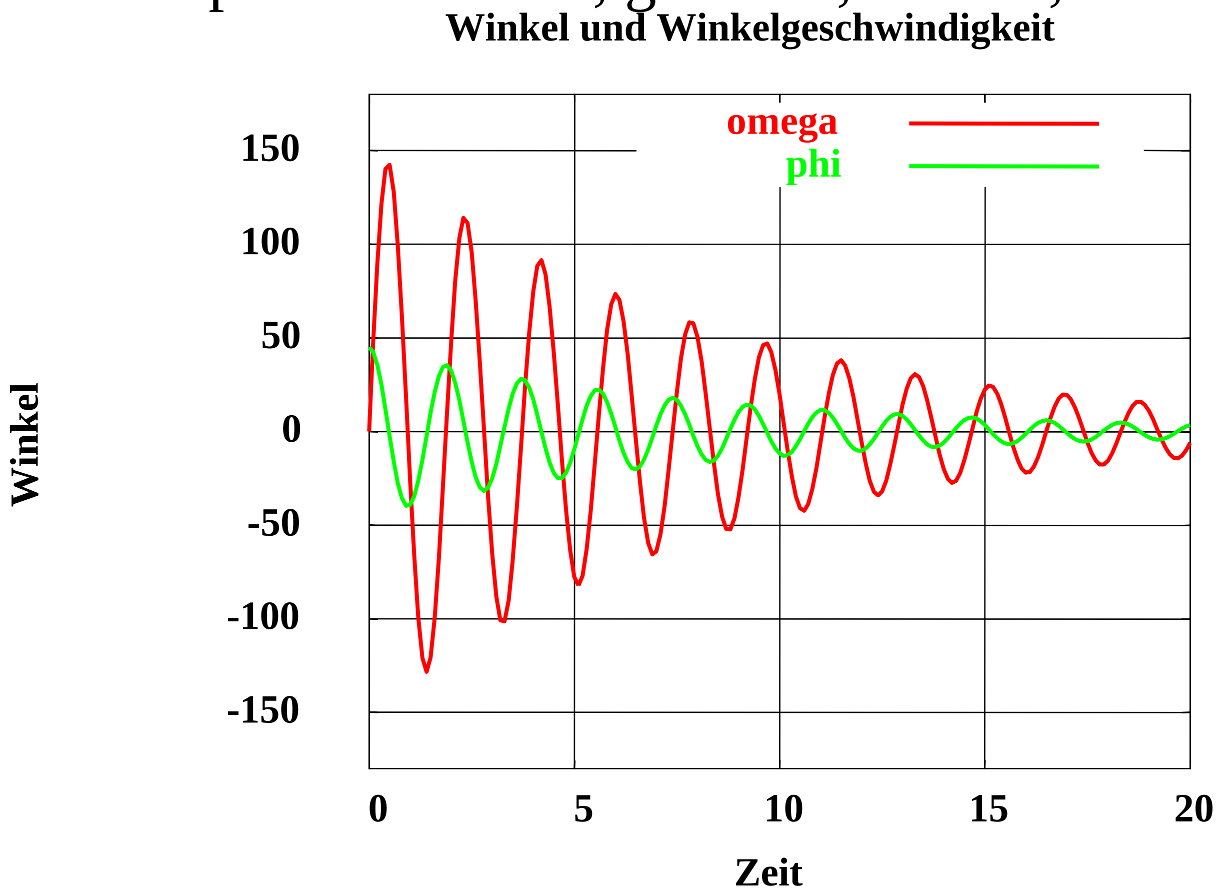
Steuerung von Außen durch Vergrößerung/Verkleinerung der Länge

Anfangszustand definieren durch

- Festlegung des Winkels φ, der Winkelgeschwindigkeit ω und der Länge l (jeweils zum Startzeitpunkt)
- Festlegung der Strategie zur Veränderung von ld üblicherweise ld Funktion von φ und l (Rückkopplung)

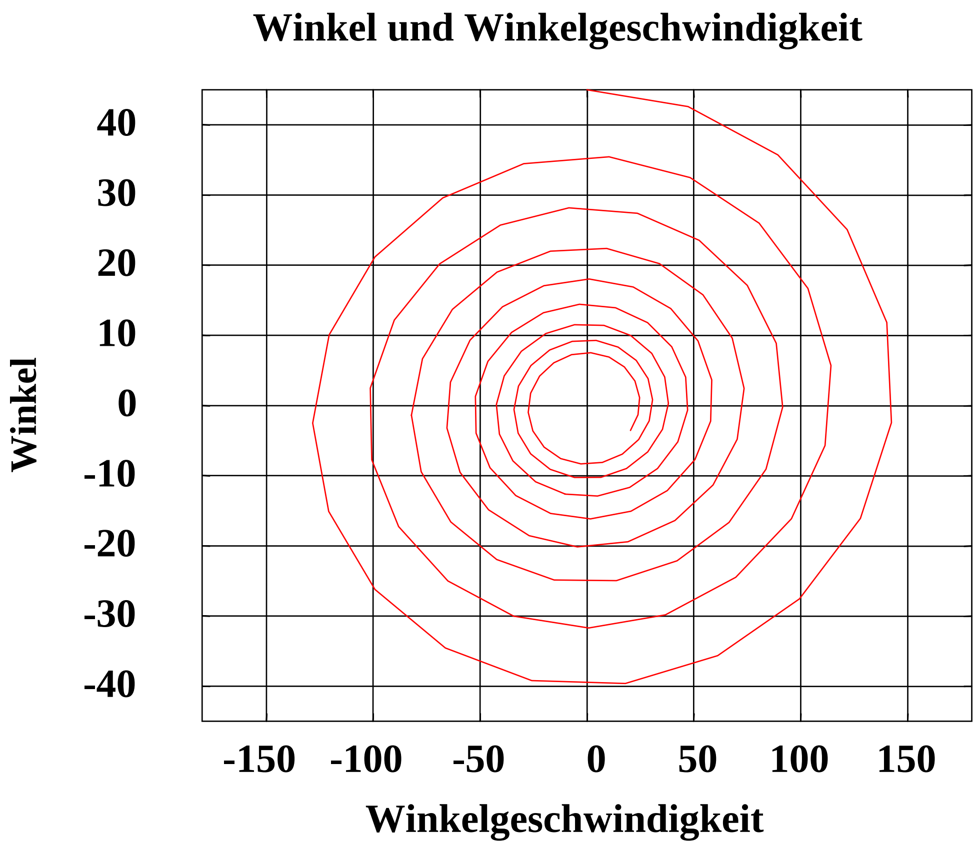
Differentialgleichungssystem kann in ein Softwaretool eingegeben und analysiert werden
(zu Softwarewerkzeugen gleich etwas mehr)

Beispiel für m = 70, g = 9.81, d = 0.5, l = 2



- Winkelgeschwindigkeit nimmt ab
- Ausschläge werden kleiner
- Schaukel kommt zur Ruhe

Darstellung im Phasenraum



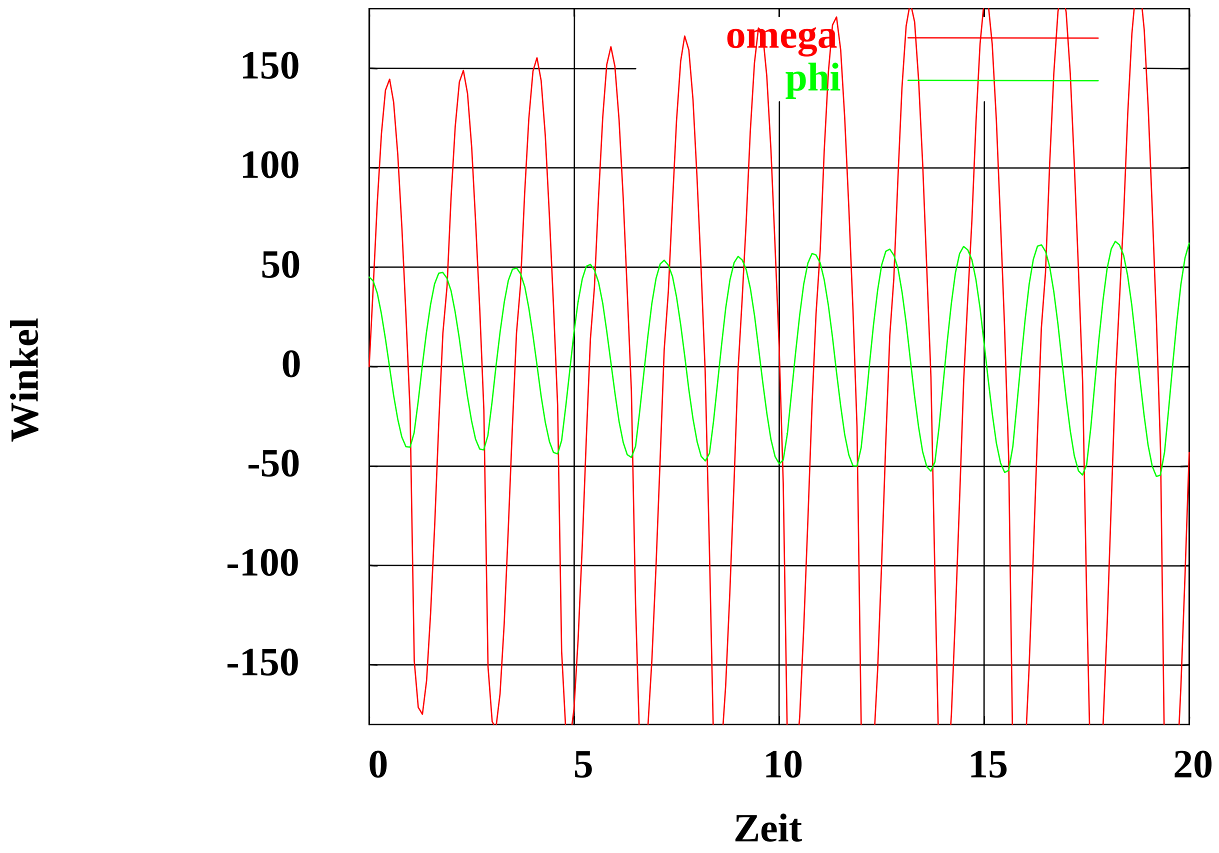
- Zusammenhang zwischen Position der Schaukel und Geschwindigkeit erkennbar
- Jeder Kreis beschreibt einen vollständigen Schwung der Schaukel
- Abnehmende Geschwindigkeit und abnehmender Ausschlag der Schaukel erkennbar
- Zeitliches Verhalten geht verloren

Weitere Möglichkeit: Dreidimensionale Darstellung
(in der Praxis oft unübersichtlich!)

Aktives Schaukeln durch Setzen von ld:

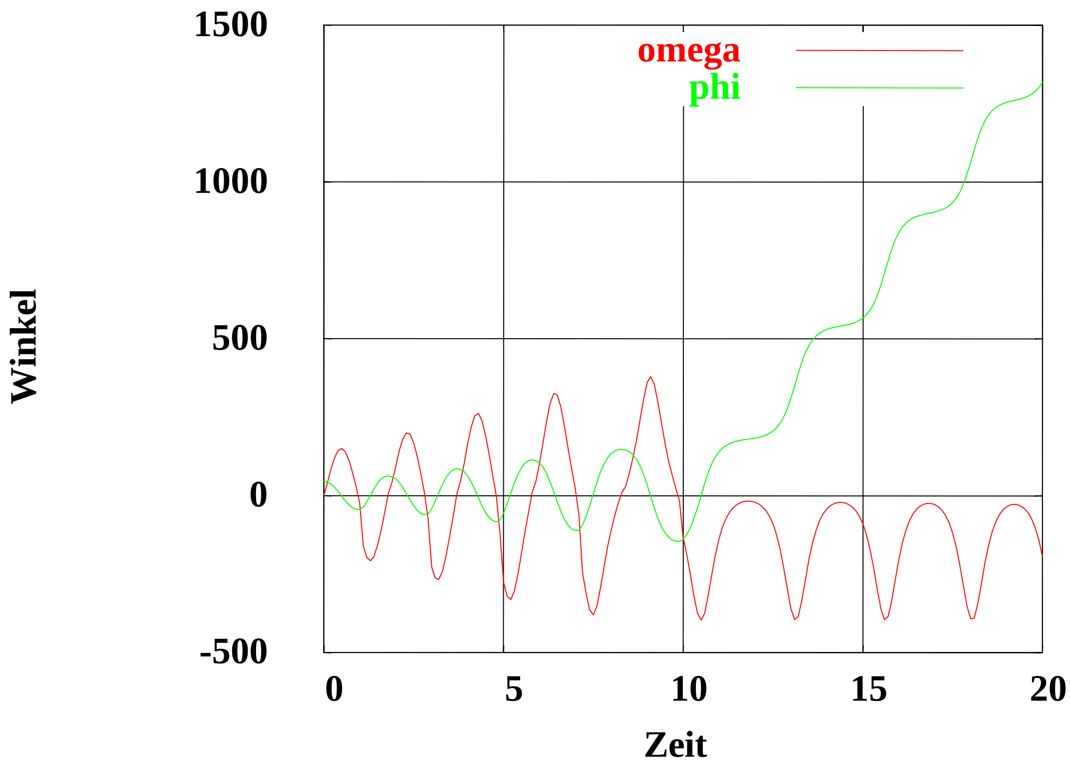
- Falls $\omega < 0$ und $l > 1.3$ dann $ld = -1$
- Falls $\omega > 0$ und $l < 2.7$ dann $ld = 1$
- Ansonsten $ld = 0$

Länge verringern/vergrößern zum aktiven Schaukeln



- Geschwindigkeit nimmt zu
- Ausschlag der Schaukel nimmt leicht zu

Was passiert ohne Reibung?



Offensichtlich eine Folge von Überschlägen, Schaukel bewegt sich nur in eine Richtung

Auf Basis des Modells können unterschiedliche Strategien analysiert werden

Validierung durch Messung am realen Objekt

Spezifikation kontinuierlicher Probleme

- Vielfalt eher noch größer als bei diskreten Systemen
- Spannbreite von der Bibliothek numerischer Verfahren über Computeralgebrasysteme bis hin zu domänenspezifischen Simulationsumgebungen
- Matlab und zugehörige Bibliotheken hat sich an vielen Stellen als Standard etabliert (hat aber auch gravierende Nachteile)

Mindestanforderungen an Software zur kontinuierlichen Modellierung-Simulation (über die Eigenschaften moderner Programmiersprachen hinausgehend):

- Darstellung/Beschreibung von Differentialgleichungen
- Numerische Lösung von Differentialgleichungen
- Visualisierung der Resultate

Beispiel für eine FORTRAN-basierte Sprache ACSL: (für ld=0)

```
PROGRAM Schaukel
 INITIAL
  constant ...
   m = 70.0, g = 9.81, d = 0.5, l = 2.0, ...
   tend = 100, omega0 = 0, phi0 = 90
  cinterval cint = 0.05
 END $ „of  INITIAL"
 DYNAMIC
  DERIVATIVE
   phid = omega ;
   omegad = (d*ω/(m*l^2) – g*sin(φ)/l)
   phi = integ(phid, phi0) ;
   omega = integ(omegad, omega0) ;
  END $ „of DERIVATIVE"
  termt(t.ge.tend)
 END $ „of DYNAMIC"
END $ „of PROGRAM"
```

Deklaration der Konstanten

Definition der initialen Variablenbelegung

Definition der Simulationsdauer

Ausgabeintervall

Werte der Ableitungen

Integration

Vorgehen und Verwendung von ACSL:

- kommerzielle Programmiersprache, die in unterschiedlichen Programmierumgebungen verfügbar ist, dadurch heute
  - oft mit integrierter graphischer Ausgabeschnittstelle
  - teilweise mit graphischer Eingabeschnittstelle
- wird trotz des Alters und der FORTRAN-Struktur noch heute benutzt
- enthält sehr effiziente Lösungsverfahren
- wird i.d.R. nach FORTRAN übersetzt und mit einer Bibliothek gelinkt

Weitere ähnliche Simulationssprachen existieren z.B. Dare-P oder Desire

Gemeinsames Prinzip:

- Programmiersprachliche Beschreibung der Gleichungen
- Simulationsunterstützung durch integrierte Funktionen

Programmierung in MATLAB
durch Beschreibung der Modellgleichungen (hier für ld=0)

```
function dxdt Schaukel(t, x) ;
  m = 70.0 ;
  g  = 9.81 ;
  d = 0.5 ;
  l = 2.0 ;
  phi = x(1) ;
  omega = x(2) ;
  dphi_dt = omega ;
  domega_dt = -d*omega/(m*l*l) – g*sin(phi)/l ;
  dxdt = [phi, omega] ;
end ;
phi0 = 90 ;
omega0 = 0.0 ;
tend = 100.0 ;
[T, X] = ode45(`Schaukel´, [0, tend], [phi0, omega0]) ;
```

Modellparameter

Variablen definieren

Differentialgleichungen

Ergebnis

Startwerte

Simulationsdauer

Aufruf der Lösung

MATLAB erlaubt

- Modellbeschreibung sehr nahe beim mathematischen Modell durch direkte Eingabe der Gleichungen
- Nutzung einer Vielzahl vordefinierter Datenstrukturen
- Nutzung mathematischer Operationen auf komplexen Daten (z.B. Matrixmultiplikation, Matrixinvertierung etc.)
- Nutzung unterschiedlicher Lösung unterschiedlicher Lösungsalgorithmen
- graphische Ausgabe

Durch weite Verbreitung existieren eine Vielzahl von Erweiterungen mit Basismodellen aus unterschiedlichen Anwendungsgebieten

Nachteile von MATLAB:

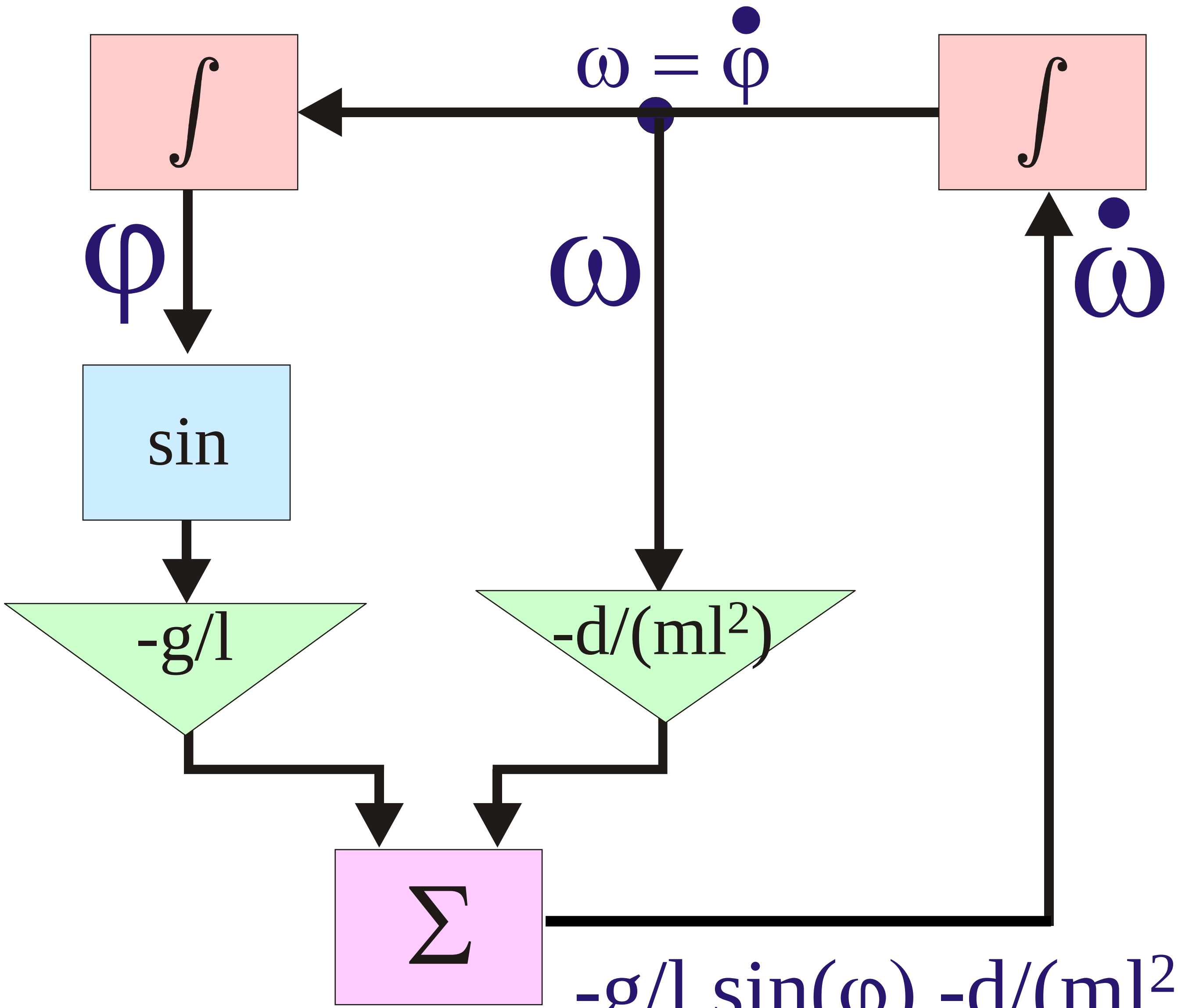
- Kommerziell und damit nicht frei nutzbar (freie Alternativen: Octave oder Scilab (weniger mächtig aber oft ausreichend))
- mangelnde Effizienz

Werkzeuge mit graphischer Eingabeschnittstelle:

- Blockorientierte Beschreibung der Abhängigkeiten zwischen Variablen und Funktionen
- Je nach Werkzeug Darstellung
  - der Variablen als Blöcke und textuelle Beschreibung der Funktionen
  - der Variablen und Funktionen als Blöcke
- Man kann unterscheidet
  - anwendungsspezifische Werkzeuge mit Symbolen der Anwendung (z.B. elek. Schaltkreise mit Symbolen für Drähte, Wiederstände, Kondensatoren etc)
  - allgemeine Darstellungen mit neutralen Symbolen

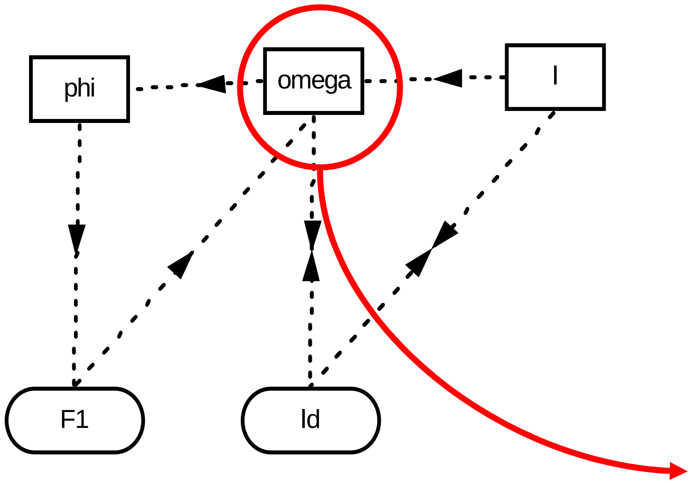
  Methodik zur Beschreibung existiert nur in einzelnen Anwendungsgebieten, keine allgemein akzeptierte Modellwelt
- Werkzeuge z.T. als Frontends für textuelle Werkzeuge (z.B. Simulink für MATLAB, Scicos für Scilab)

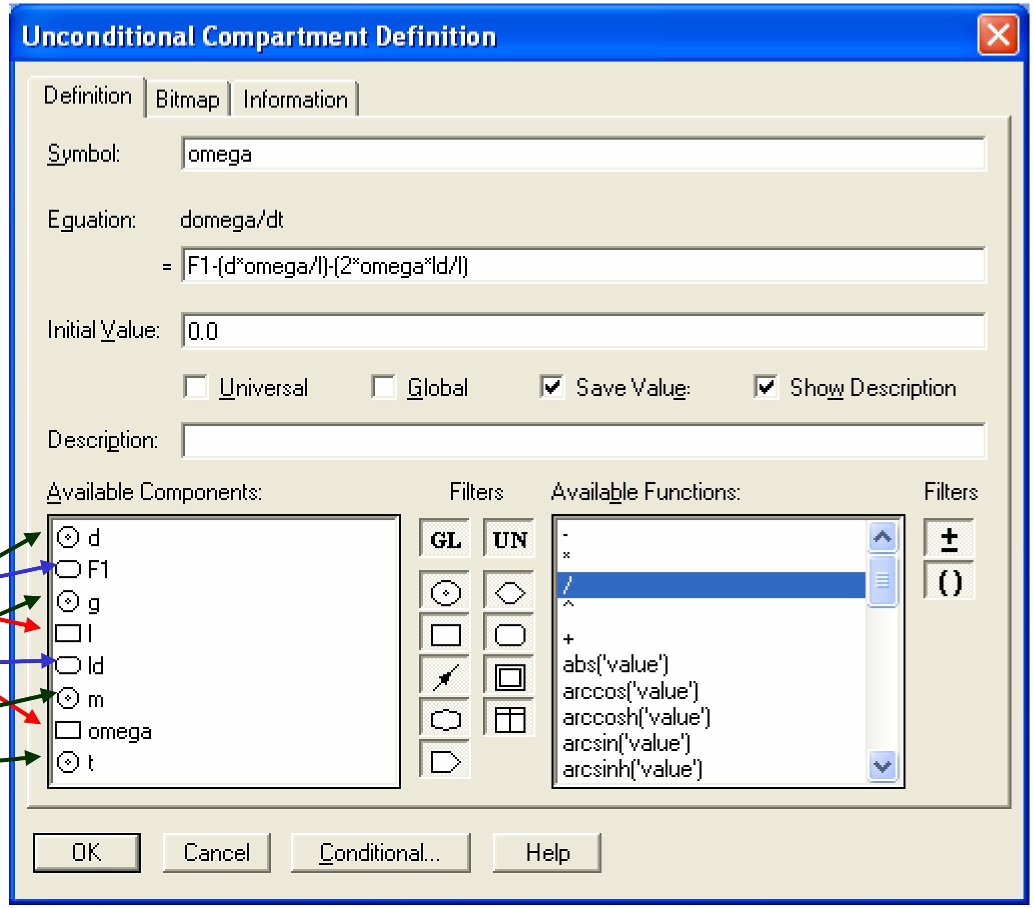
Funktionsorientierte Beschreibung in Simulink



$$-g/l \sin(\varphi) - d/(ml^2)\,\omega = \dot{\omega}$$

Variablenorientierte Beschreibung in ModelMaker





Differentialgleichungen

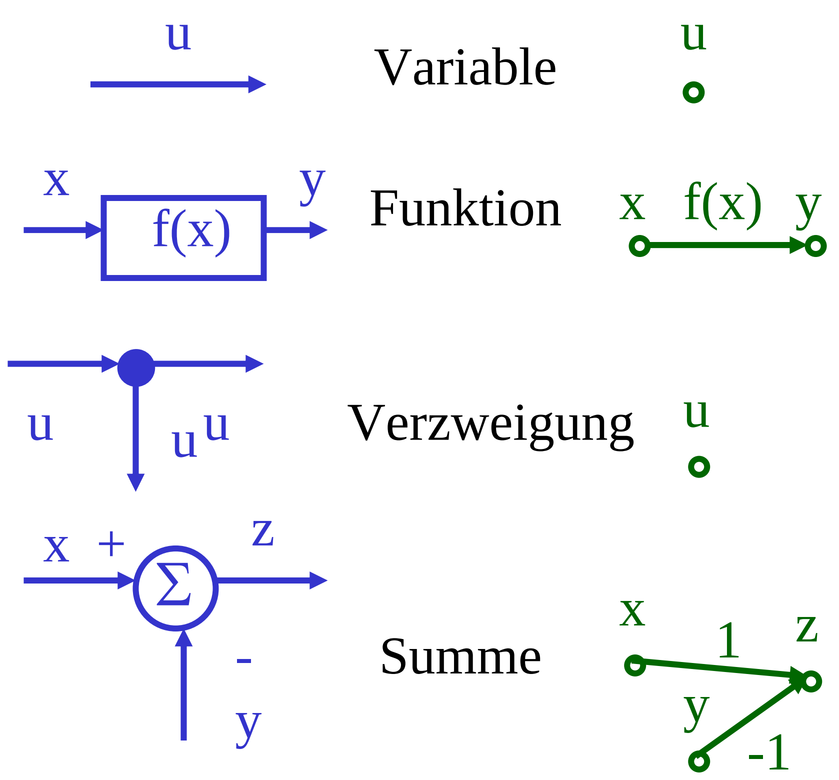
Variablen

Konstanten

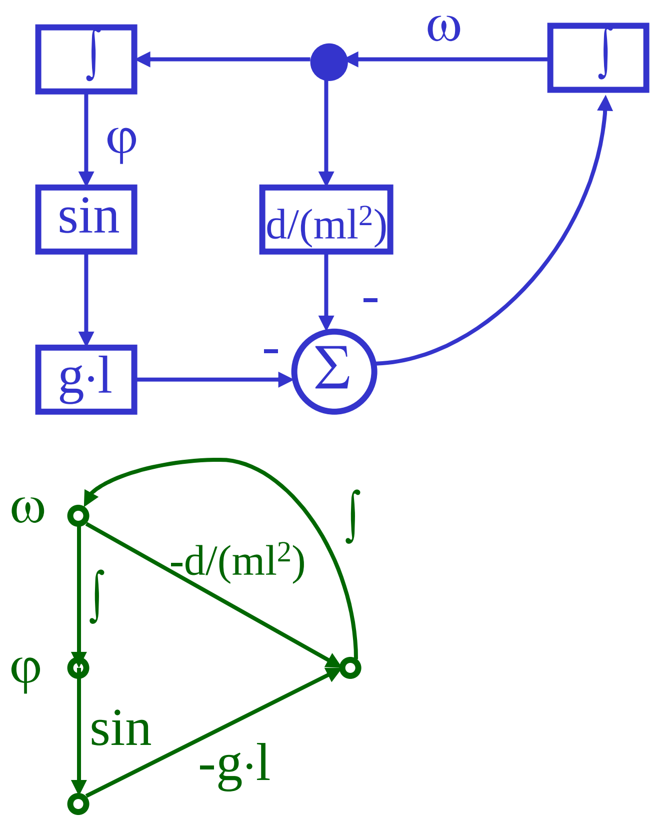
Sichtbarkeit auf Eingabegrößen beschränkt!

Basis der graphischen Beschreibungen

Blockdiagramme   Flussgraphen



Schaukel-Beispiel



- Basiselemente reichen zur Beschreibung einfacher Systeme aus
- Erweiterung der graphischen Modellierung durch Bond-Graphen (hier nicht behandelt

4.2 Analysetechniken für kontinuierliche Systeme

Allgemeine Form eines kontinuierlichen Systems: $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t)$

Wir betrachten den einfachen Sonderfall: $\dot{\mathbf{x}} = f(\mathbf{x})$

- Vektor **u** kann durch Erweiterung während der Lösung berücksichtigt werden
- Oft gilt für den Lösungsvektor **p** = g(**x**), so dass Beobachtung von **x** ausreicht

Beschreibung des Modells kann (und wird oft) allgemeiner sein, da

1. höhere Ableitungen vorkommen
2. die Zeit t explizit vorkommt
3. nur Beziehungen zwischen Variablen beschrieben werden

In diesen Fällen muss zuerst die Modellbeschreibung (möglichst automatisch) so transformiert werden, dass die obige Darstellung entsteht

Ersetzung höherer Ableitungen durch neue Variablen

Differentialgleichung der Ordnung n wird durch n Differentialgleichungen erster Ordnung ersetzt

Beispiel: $x^{(n)} = f(x, \dot{x}, \ddot{x}, \ldots, x^{(n-1)})$ mit $x^{(i)}(0) = C_i$

wird ersetzt durch

$$\left.\begin{aligned} \dot{x}_0 &= x_1 \\ \dot{x}_1 &= x_2 \\ &\vdots \\ \dot{x}_{n-2} &= x_{n-1} \\ \dot{x}_{n-1} &= f(x_0, \ldots, x_{n-2}) \end{aligned}\right\}$$

System erster Ordnung mit initialen Werten $x_i = C_i$

Darstellung der Zeit durch Einführung einer expliziten Variablen

$\dot{\mathbf{x}} = f(\mathbf{x}, t)$ mit n-1 Variablen $\Rightarrow$

Einführung von x_n mit $\dot{x}_n = \frac{dx_n}{dt} = 1$ und $x_n(0) = 0$

Modellbeschreibung (wie z.B. auf Folie 17-24) liefert eine Menge von Beziehungen zwischen den Variablen

- Beziehungen sind als Gleichungen (und nicht als Zuweisungen) zu verstehen
- In der Regel sind deutlich mehr Gleichungen als Variablen vorhanden

$\Rightarrow$ Gleichungen sind so zu transformieren, dass Differentialgleichungssystem in der gewünschten Form entsteht

Transformationsschritte (erst horizontale Sortierung):

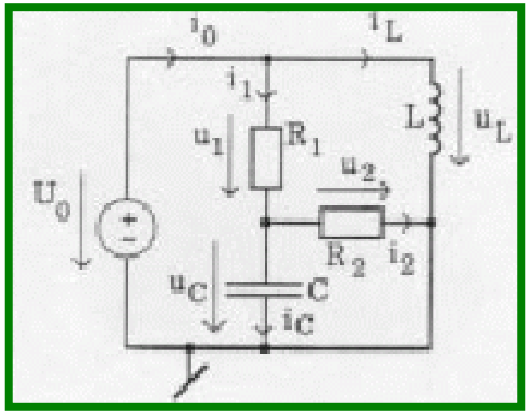
1. Zustandsvariablen (die in Differentialgleichungen auftauchen oder als Funktionen definiert sind) können als bekannt vorausgesetzt werden (ergeben sich später aus den Differentialgleichungen)
2. Variablen, die in Gleichungen vorkommen, die neben der Variable nur eine Zustandsvariable enthält werden aus dieser Gleichung berechnet
3. Variablen, die in nur einer Gleichung vorkommen, müssen aus dieser Gleichung berechnet werden

Schritte 1.-3. werden rekursiv angewendet

dann vertikale Sortierung

4. Gleichungen so sortieren, dass Variablen Werte zugewiesen werden, bevor sie benutzt werden
5. Falls noch nicht alle Werte bekannt $\Rightarrow$ symbolische Gleichungslösung

Bespiel (aus Cellier 1991):



5 Elemente in der Schaltung jeweils Berechnung von Strom und Spannung $\Rightarrow$
10 Werte sind zu berechnen $\Rightarrow$
10 Gleichungen werden benötigt

Vollständiger Satz von Gleichungen (redundante Beziehungen wurden bereits entfernt)

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$u_1 = R_1 \cdot i_1$	$i_1 = i_2 + i_C$
$u_2 = R_2 \cdot i_2$	$U_0 = u_1 + u_C$
$i_C = C \cdot du_C/dt$	$u_C = u_2$
$u_L = L \cdot di_L/dt$	$u_L = u_1 + u_2$

○ bekannt nach Regel 1.

○ berechnen nach Regel 2.

○ berechnen nach Regel 3.

◌ bekannt nach einmaliger Anwendung der Regel

Resultierende Gleichungen nach mehrfacher Anwendung der Regeln und Umformung der Gleichungen, so dass Zuweisungen definiert werden

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$i_1 = u_1 / R_1$	$i_C = i_1 - i_2$
$i_2 = u_2 / R_2$	$u_1 = U_0 - u_C$
$du_C/dt = i_C / C$	$u_2 = u_C$
$di_L/dt = u_L / L$	$u_L = u_1 + u_2$

vertikale Sortierung

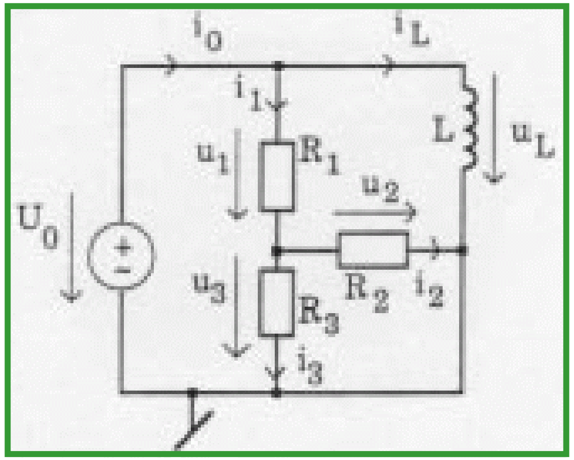
$U_0 = f(t)$	$i_2 = u_2 / R_2$
$u_1 = U_0 - u_C$	$i_C = i_1 - i_2$
$i_1 = u_1 / R_1$	$u_L = u_1 + u_2$
$i_0 = i_1 + i_L$	$du_C/dt = i_C / C$
$u_2 = u_C$	$di_L/dt = u_L / L$

Resultierendes Differentialgleichungssystem mit 2 Variablen (i_L und u_C), die restlichen Variablenwerte ergeben sich aus den Konstanten.

$$
\begin{aligned}
du_C/dt &= i_C / C \\
&= (i_1 - i_2) / C \\
&= i_1 / C - i_2 / C \\
&= u_1 / (R_1 \cdot C) - u_2 / (R_2 \cdot C) \\
&= (U_0 - u_C) / (R_1 \cdot C) - u_C / (R_2 \cdot C)
\end{aligned}
$$

$$
\begin{aligned}
di_L/dt &= u_L / L \\
&= (u_1 + u_2) / L \\
&= u_1 / L + u_2 / L \\
&= (U_0 - u_C) / L + u_C / L \\
&= U_0 / L
\end{aligned}
$$

Ein weiteres Bespiel (aus Cellier 1991):



5 Elemente in der Schaltung jeweils Berechnung von Strom und Spannung ⇒ 10 Werte sind zu berechnen ⇒ 10 Gleichungen werden benötigt

Vollständiger Satz von Gleichungen (redundante Beziehungen wurden bereits entfernt)

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$u_1 = R_1 \cdot i_1$	$i_1 = i_2 + i_3$
$u_2 = R_2 \cdot i_2$	$U_0 = u_1 + u_3$
$u_3 = R_3 \cdot i_3$	$u_3 = u_2$
$u_L = L \cdot di_L/dt$	$u_L = u_1 + u_2$

○ Bekannte Variablen nach rekursiver Anwendung der Regeln

○ zu berechnen nach Regel 3.

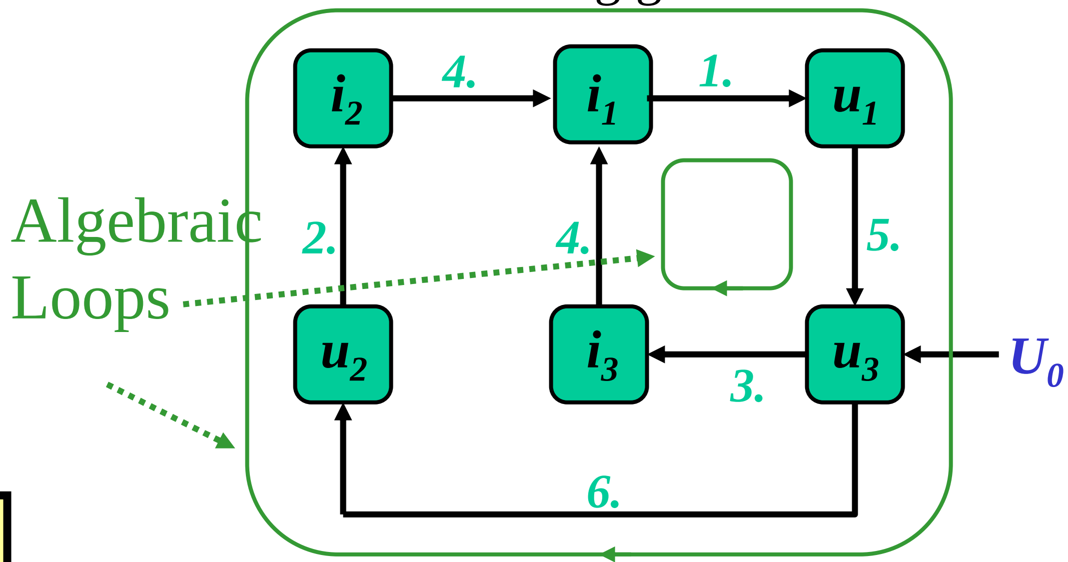
- alle Gleichungen enthalten mindestens zwei unbekannte Variablen und
- jede unbekannte Variable kommt in mindestens zwei Gleichungen vor

„algebraic loop“

Verbleibende Gleichungen

1. $u_1 = R_1 \cdot i_1$	4. $i_1 = i_2 + i_3$
2. $u_2 = R_2 \cdot i_2$	5. $U_0 = u_1 + u_3$
3. $u_3 = R_3 \cdot i_3$	6. $u_3 = u_2$

Struktur der Abhängigkeit



Variablen freistellen (jede Variable einmal auf linker Seite)

1. $u_1 = i_1 / R_1$	4. $i_1 = i_2 + i_3$
2. $i_2 = u_2 / R_2$	5. $u_3 = u_1 + U_0$
3. $i_3 = u_3 / R_3$	6. $u_2 = u_3$

Zyklen können nicht durch Umordnen der Variablen gebrochen werden!

Auflösung der Schleifen durch symbolische Gleichungslösung

Hier konkret:

- Auswahl einer Variablen, die auf beiden Schleifen vorkommet (z.B. i_1)
- Annahme, dass diese Variable bekannt sei
- Auflösung des restlichen Gleichungssystems
- Berechnung der ausgewählten Variablen aus der Gleichungslösung

Lösung des Gleichungssystems bei bekanntem i_1:

$u_1 = R_1 \cdot i_1$	$u_2 = u_3$
$u_3 = U_0 - u_1$	$i_2 = u_2 / R_2$
$i_3 = u_3 / R_3$	

Berechnung von i_1:

$$
\begin{aligned}
i_1 &= i_2 + i_3 \\
&= u_2 / R_2 + u_3 / R_3 \\
&= u_3 / R_2 + u_3 / R_3 \\
&= ((R_2 + R_3) / (R_2 \cdot R_3)) \cdot u_3 \\
&= ((R_2 + R_3) / (R_2 \cdot R_3)) \cdot (U_0 - u_1) \\
&= ((R_2 + R_3) / (R_2 \cdot R_3)) \cdot (U_0 - R_1 \cdot i_1)
\end{aligned}
$$

$$\boxed{i_1 = \frac{R_2 + R_3}{R_1 R_2 + R_1 R_3 + R_2 R_3} \cdot U_0}$$

Vorgehen im allgemeinen Fall :

- Algorithmen (z.B. von Tarjan) existieren,
  - um Schleifen zu identifizieren und
  - Variablen entsprechend umzuordnen
- Lösung der resultierenden Gleichungssysteme
  - für lineare Gleichungssysteme mit wenigen Variablen symbolische Lösung
  - für lineare Gleichungssysteme mit vielen Variablen erfolgt eine numerische Lösung (mittels Gauß-Elimination oder iterativen Verfahren)
  - für nichtlineare Systeme numerische Lösung oft mittels Newton-Iteration

Kontinuierliche Simulation zur Berechnung von $\mathbf{x}(t)$
auf Basis der Information $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t)$ und $\mathbf{x}(0) = \mathbf{x}_0$

Einfachster Lösungsansatz Euler-Verfahren,
d.h. Diskretisierung mit fester Schrittweite Δt:

Sei $\mathbf{x}_k = \mathbf{x}(k \cdot \Delta t)$ und $\mathbf{u}_k = \mathbf{u}(k \cdot \Delta t)$ sowie $t = K \cdot \Delta t$
(damit ist $\mathbf{x}_K$ zu berechnen)
Die Approximation

$$\int_{k \cdot \Delta t}^{(k+1) \cdot \Delta t} f(\mathbf{x}(t), \mathbf{u}(t), t) dt \approx \Delta t \cdot f(\mathbf{x}_k, \mathbf{u}_k, k \cdot \Delta t)$$

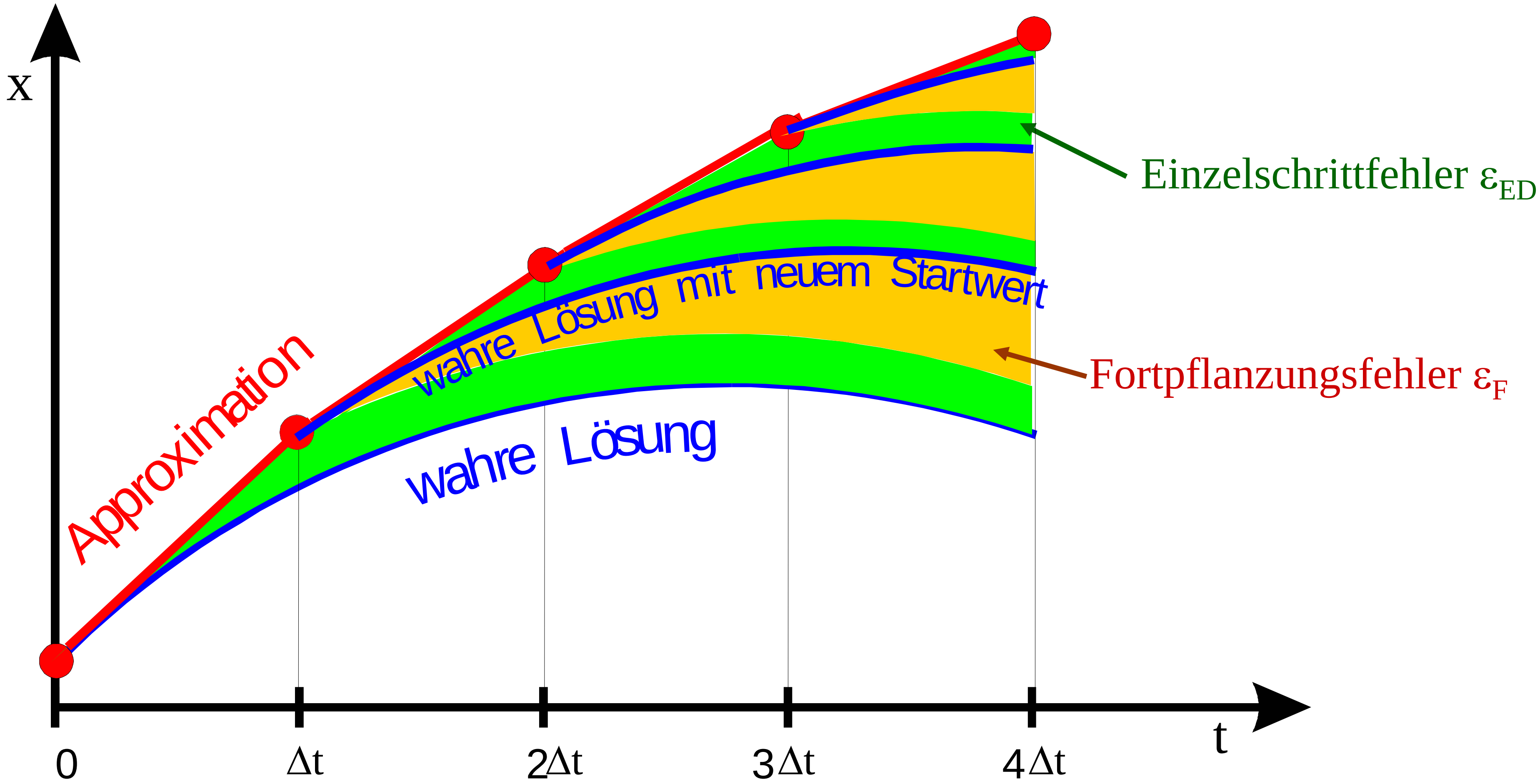
liefert

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot f(\mathbf{x}_k, \mathbf{u}_k, k \cdot \Delta t)$$ (Linearisierung der Funktion)

Ergebnis kann einfach in K Schritten berechnet werden
Fehler durch die Linearisierung der Funktion

Freiheitsgrad: Wahl von Δt

Fehler der Methode (veranschaulicht für ein eindimensionales Problem)



$$\varepsilon_{ED}(k\cdot\Delta t) = |x((k+1)\cdot\Delta t) - x(k\cdot\Delta t) - \Delta t\cdot f(x(k\cdot\Delta t), u(k\cdot\Delta t), k\cdot\Delta t)|$$

$$\varepsilon_D = \varepsilon_{ED}(0) + \varepsilon_{ED}(\Delta t) + \ldots + \varepsilon_{ED}((n-1)\cdot\Delta t)$$

mit $n = 1/\Delta t$ Diskretisierungsfehler

x(t) ist der exakte Wert

Für das Euler Verfahren gilt $\varepsilon_D, \varepsilon_F \in O(\Delta t)$

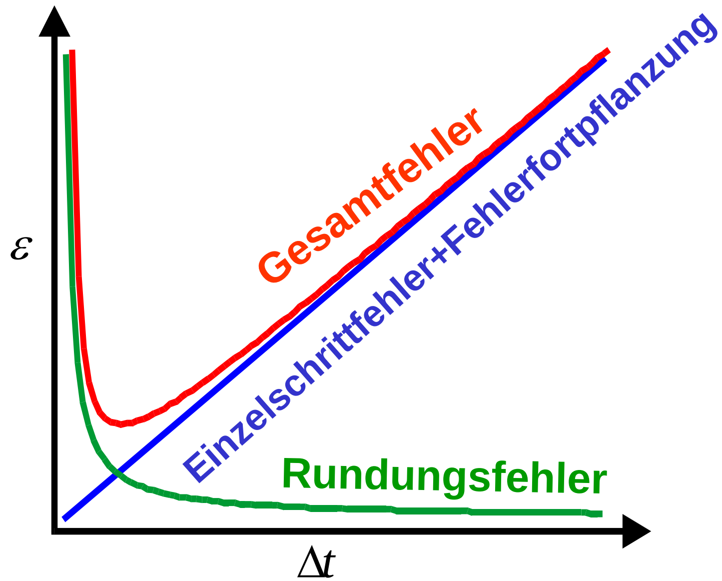
??

$\Rightarrow$ Δt kann so klein gewählt werden, dass der Fehler verschwindet

Offensichtlich nicht, da Rechner nur mit endlicher Genauigkeit rechnen!

Beispiel 32 Bit Zahlendarstellung einfache Genauigkeit:

- 8 Bit Exponent
- 24 Bit Mantisse $\Rightarrow$ erreichbare Genauigkeit $2^{-24} \approx 10^{-6}$ (ε_R Rundungsfehler)
- damit gilt für den Gesamtfehler $\varepsilon \leq \varepsilon_D + \varepsilon_F + \varepsilon_R$



I.d.R. ist der Rundungsfehler deutlich kleiner als der Verfahrensfehler, zumindest wenn mit doppelter Genauigkeit gearbeitet wird