10. Kontinuierliche Simulation - Eine Skizze

Erinnerung: Abschnitte 1 + 2

- **System** Menge Objekte,
  je mit Attributen,
  mit Zusammenhängen / Abhängigkeiten
  von Objekten (bzw Attributen)

- **Systemanalytiker** interessiert an
  Einfluß von kontrollierbaren + unkontrollierbaren
  (Einfluß-)Größen (C + U)
  auf beobachtbare (Auswirkungs-)Größen (P)

  in Form von Funktion $f: (W_C, W_U) \rightarrow W_P$
  oder Relation $f \quad W_C \times W_U \times W_P$

  als Basis von Beurteilungs- ("Güte"-) Kriterien $V = \{v_i\}$,
  welche aus Beobachtungsgrößen errechenbar
  $v_i = g_i(w_P) \qquad i=1,2,\ldots,n$

- Verfolgung der Aufgabe ("Technik") mithilfe von
  Modell Ersatzsystem

  - einfacher zu handhaben als System

  - Einflüsse → Auswirkungen ("f")
    reproduzierend
    (damit notwendig Zweck/Ziel - abhängig)

- Festlegung: **nicht-materielle** Modellklassen

  darin
  - closed box (black box) - Modelle:
    nur (C,U) $\leftrightarrow$ P - Zusammenhang
    beschreibend
  - open box (white box) - Modelle:
    Mechanik / Zustandekommen
    des (C,U) $\rightarrow$ P - Zusammenhangs
    beschreibend

- weitere Interesseneingrenzung:
  dynamische Modelle / Systeme

  - closed box: zeitabhängiger Verlauf von Einflüssen und Auswirkungen
  - open box: zeitabhängige Mechanik über **Zustand** ("variiert über der Zeit")

  dabei Charakterisierung
  zeit- / zustands- diskreter / kontinuierlicher
  Systeme / Modelle
  als Betrachtungsentscheidung ("Zweck / Ziel")
  s. Bild 10.0.01

- und (exklusive) Konzentration auf:
  ereignisorientierte Modelle / Systeme

  - zustandsdiskret oder zustandskontinuierlich
  - zeitkontinuierlich,
    aber mit "sprunghaften" Zustandsänderungen
    zu Zeit-**Punkten** (in "verschwindender" Zeit)

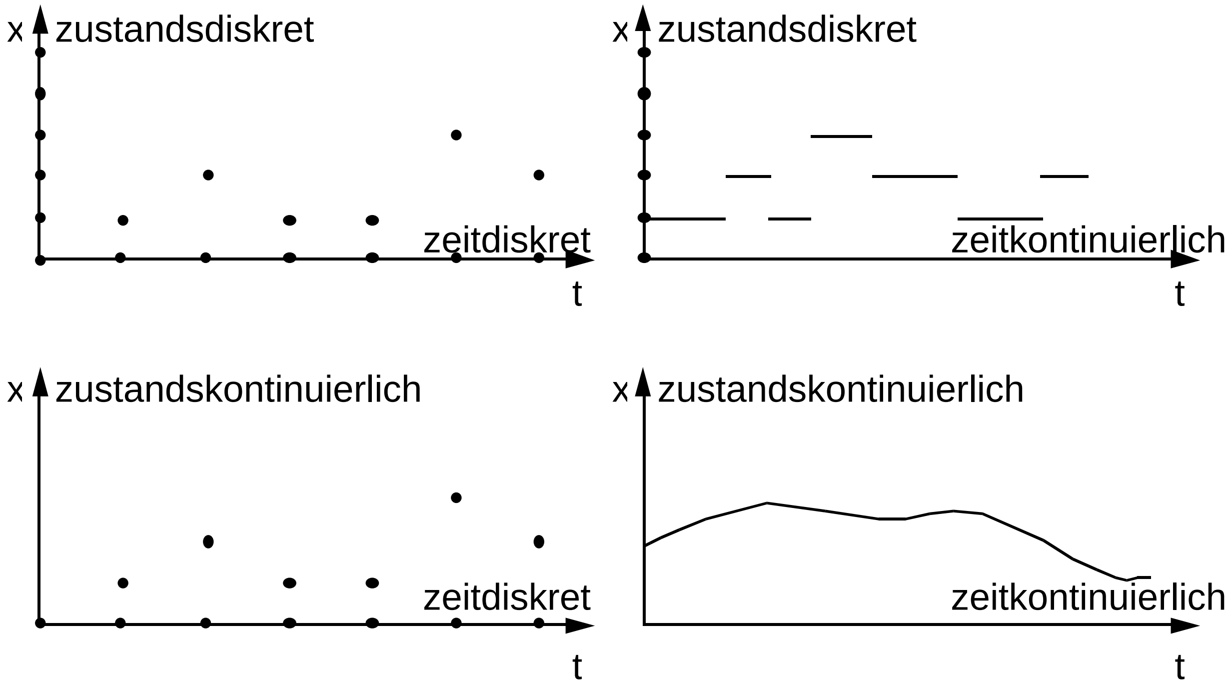


Bild 10.0.1: zeit- / zustands- diskrete / kontinuierliche Trajektorien

Ereignisorientiertheit

- ist (simulations-) effiziente Vorstellung
- trägt auch "ein Stück weit" in Richtung zustands-/zeit-kontinuierlicher Systeme
- versagt aber "im allgemeinen" für letztere

Beispiele von Systemen / Modellen
/ Analysen (→Simulationen)
"am anderen Ende":

Beispiel 10.0.2: Häuser und Klimaanlagen

- Annahmen:
  Bau / Verkauf von Häusern (in Bereich)
  proportional zu
  H (=const) Zahl Familien ohne Haus, mit Hauswunsch

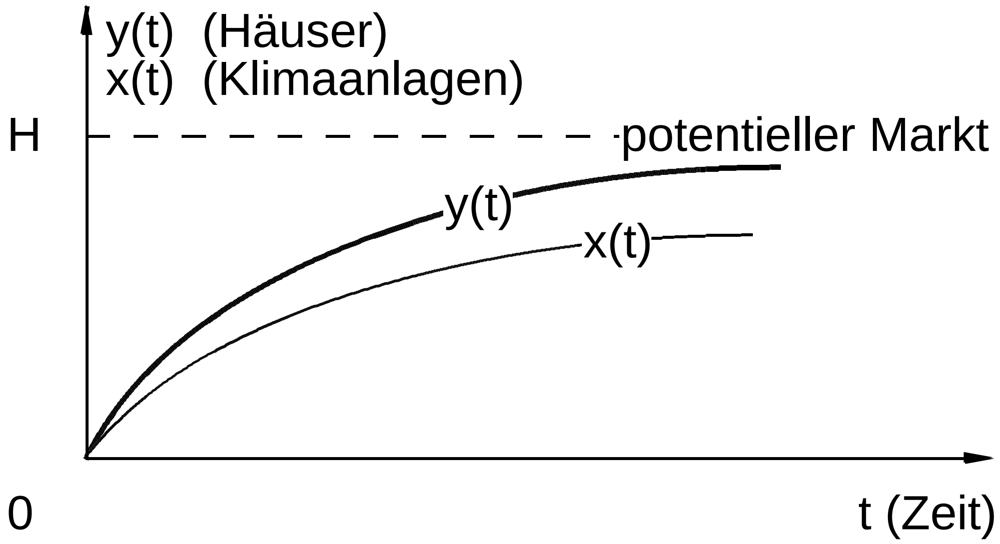
- formale Fassung:
  $y(t)$ bis t (seit t=0) gebaute Häuser
  $$\frac{dy(t)}{dt} = k_1 \left(H - y(t)\right) \qquad y(0) = 0$$
  kurz auch
  $$\dot{y}(t) = k_1 \left(H - y(t)\right) \qquad y(0) = 0$$
  $$\dot{y} = k_1 \left(H - y\right) \qquad y = 0 \text{ für } t = 0$$

- weiter angenommen:
  Verkauf von Klimaanlagen
  proportional zu neugebauten Häusern ohne Klimaanlage
  (alle alten haben schon, alle neuen wollen)

- formale Fassung:
  $x(t)$ bis t (seit t=0) verkaufte Klimaanlagen
  $$\dot{x} = k_2 \left(y - x\right) \qquad x = 0 \text{ für } t = 0$$

  System von **Differentialgleichungen** beschreibt
  y(t), x(t)-Verlauf
  in diesem Fall explizit lösbar (→ Funktionen y(t), x(t))

Erahnter Verlauf
(zeit- + zustandskontinuierliche Trajektorien):



Typ der DGl:

- "gewöhnlich": Ableitungen ausschließlich nach einer (der unabhängigen) Variablen "Zeit"
  sonst: + partielle Ableitungen, oft: "Raum"

- "linear": unabh.Var. + Ableitungen nur 1. Potenz, keine Produkte (Variablen, Ableitungen)

- "1. Ordnung": (höchste) Ableitungen 1. Ordnung

- "mit konstanten Koeffizienten"
  sonst: zeitabhängige Koeffizienten

(hier:) explizite Lösung verfügbar:

$$y(t) = H \left(1 - e^{-k_1 t}\right)$$

In komplexeren Fällen

zB Marktgrenze H nicht konstant
wg Populationswachstum,
ökonomischen Bedingungen

zB Proportionalitätskoeffizienten nicht konstant
wg marketing-Kampagnen,
Konkurrenzmärkten (Mietwohnungen,…)

- explizite Lösbarkeit uU nicht gegeben
(auch: Kenntnisproblem!)

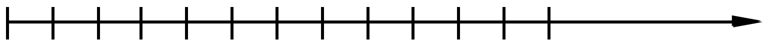
- "Simulation" vorstellbar
("nur wenns nicht anders geht"!)

Verfolgung des Zustands über der Zeit,
durch Imitation (zustandsabhängiger) Zustandswechsel,
gemäß "Dynamik" des Systems (≡ D'Gl-System !)

- Typische Methode:

(sogar notwendigerweise:
Im Digitalrechner prinzipiell
Zustand + Zeit nur diskret änderbar):

Statt kontinuierlicher Zeit
nun diskrete Zeit betrachtet



t (Zeit)

diskrete Zeit mit Zeitpunkten

$t_0, t_1, \ldots, t_i = t_{i-1} + (\Delta t)_i, \ldots$

wo oft (nicht notwendig)

$(\Delta t)_i \equiv \Delta t$

und

statt Zuständen zu kontinuierlicher Zeit
nun Zustände zu diskreten Zeitpunkten

ermittelt (zB)
durch Festhalten (Konstanz)
der Zustandsänderungs"raten" ("Ableitungen")
über (t_i, t_{i+1})-Intervallen

Diskretisierung von **Differentialgleichungssystem**
zu **Differenzengleichungssystem**

Im Beispiel:

- Veränderung y im Intervall (t_i, t_{i+1})

  $\Delta y_i = const_i \cdot \Delta t$

  mit (zB) Wahl von $const_i$ "wie zu Anfang des Intervalls"

  $const_i = k_1 \cdot (H - y_i)$

  und folglich

  $y_{i+1} = y_i + const_i \cdot \Delta t = y_i + k_1 \cdot (H - y_i) \cdot \Delta t$

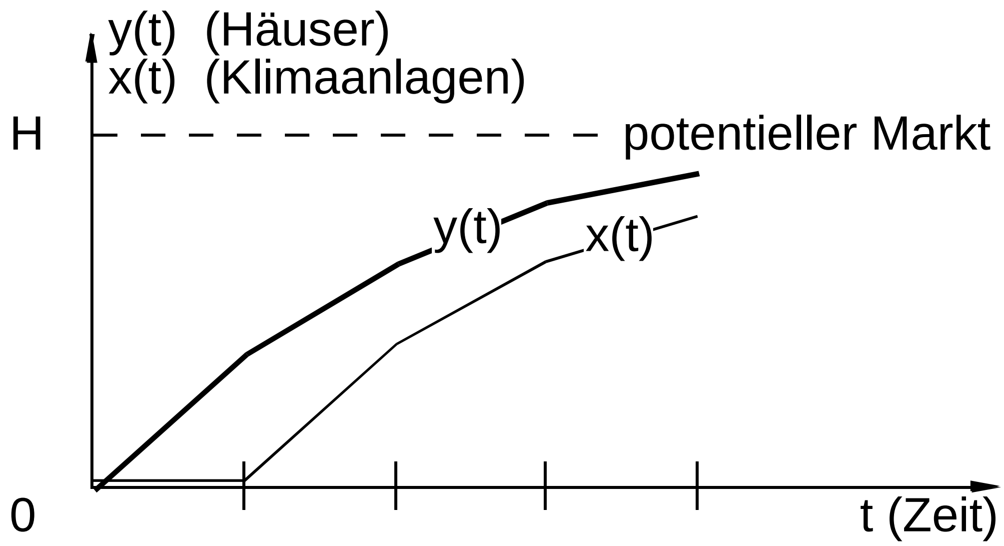
- (entsprechend:)
  Veränderung x im Intervall (t_i,t_{i+1}) gemäß
  $x_{i+1} = x_i + k_2 \cdot (y_i - x_i) \cdot \Delta t$

- und Simulationsvorgang insgesamt:

  Initialisierung:
  $y_0 = 0 \qquad \Delta y_0 = k_1 \cdot H \cdot \Delta t$
  $x_0 = 0 \qquad \Delta x_0 = 0$

  Iteration i=1,2,… :
  $y_i = \ldots \qquad x_i = \ldots$
  $\Delta y_i = \ldots \qquad \Delta x_i = \ldots$

- und "tatsächlich verfolgte" Trajektorien:



  natürlich nur "Approximation" !

Entwickelt: (sehr einfache)
Methode der **numerischen Integration**

Es gibt bessere (genauere) Methoden (s. später)

Beispiel 10.0.3: Rad-Auslenkung (vertikal)

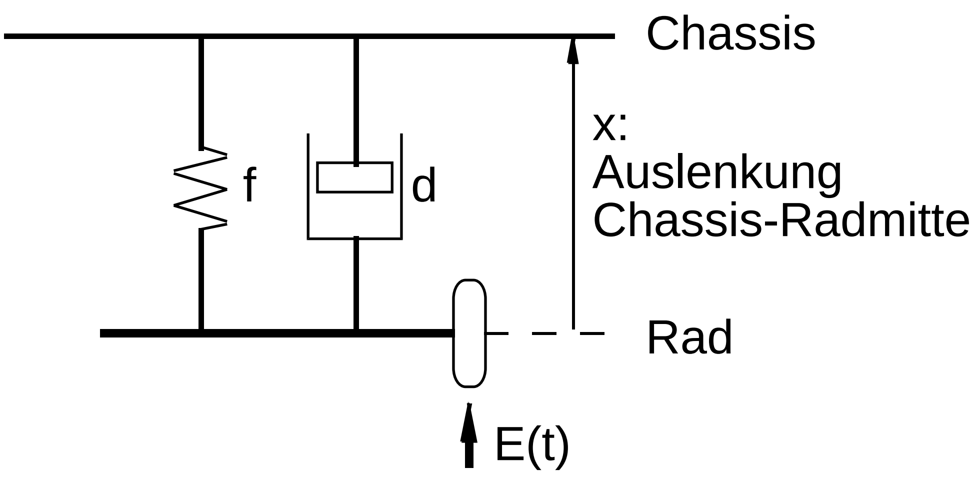
- Annahmen:

  Automobil in Bewegung

  (bewegungs- →)
  zeitabhängige Krafteinwirkung auf Rad E(t)

Rad mit	Masse	m
Radauslenkung vertikal gedämpft durch		
Feder,	Federkonstante	f
Stoßdämpfer,	Dämpfungskonstante	d

- Skizze



- physikalische → mathematische Gesetze

  Weg / Geschwindigleit / Beschleunigung

  Kraft = Masse · Beschleunigung

Kräfte

- externe "Störung": $E(t)$
- Federkraft, Annahme:
  proportional Auslenkung $-f \cdot x(t)$
- Dämpfung, Annahme:
  proportional Geschwindigkeit $-d \cdot \dot{x}(t)$

$$E(t) - f \cdot x(t) - d \cdot \dot{x}(t) = m \cdot \ddot{x}(t)$$

$$m \cdot \ddot{x} + d \cdot \dot{x} + f \cdot x = E(t)$$

- lineare Differentialgleichung 2. Ordnung
  mit konstanten Koeffizienten

  ggf, abhängig von E(t), explizit lösbar
  Anfangsbedingungen, $x_0, \dot{x}_0$, erforderlich
  → x(t) "gedämpfte Schwingung"

- in komplexeren Fällen:
  kontinuierliche Simulation

  einfacher Zugang (analog Bsp.10.0.02):
  - Einführung "Zustandsgrößen"
    (Ableitungen, i.Vgl. zu Vorkapiteln "künstlich")

    $$x_0 := x$$
    $$x_1 := \dot{x}$$

  - liefert DGl-System 1. Ordnung

    $$\dot{x}_0 = x_1$$
    $$\dot{x}_1 = \frac{1}{m} \left(E(t) - f \cdot x_0 - d \cdot x_1\right)$$

    welches über der Zeit "schrittweise" vefolgbar

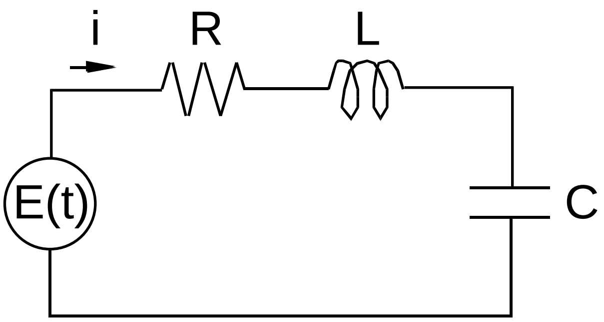
Beispiel 10.0.4: Elektrischer Schwingkreis

- Annahmen:

  Elektrisches Netzwerk, Reihenschaltung aus

-	Ohmscher	Widerstand	R
-	Spule	Induktivität	L
-	Kondensator	Kapazität	C
+	zeitabhängige (Spannungs-)Anregung		E(t)

- Skizze



- physikalische → mathematische Gesetze

  Kirchhoffsche Gesetze
  Spannung / Strom an Schaltelementen

- mit Ladung des Kondensators $q(t)$
  ergibt sich Zusammenhang:

$$LC \;\ddot{q} + RC\;\dot{q} + q = E(t)$$

- lineare Differentialgleichung 2. Ordnung
  mit konstanten Koeffizienten

  ggf, abhängig von E(t), explizit lösbar
  Anfangsbedingungen, q_0, $\dot{q}_0$, erforderlich

  → q(t) "gedämpfte Schwingung"

- in komplexeren Fällen:
  kontinuierliche Simulation

- aber auch weiterer Weg zur Lösung aufscheinend:

  - DGl formal identisch zu jener aus Beispiel 10.0.03 !!

  - elektrischer Schaltkreis (auch allgemeinere)
    "am Laborplatz" aufbaubar
    betreibbar
    beobachtbar / meßbar
    → Lösung q(t) aufzeichenbar

  - Lösung auf mechanische Problem übertragbar !!

- ehemals breit eingesetzt: **Analogrechner**

  entsprechend DGl (-System) zusammengeschaltete
  Menge von
  (Grund-)Elementen

  → "**analoges Modell**"

- Grund-Schaltungs-Elemente Analogrechner

  - Integratoren (Signalintegration über der Zeit)
  - Skalierer (Multiplikation mit Faktor)
  - Addierer (Addition)
  - Inverter (Signalumkehr)
  - …

- Blockschaltbilder (aus Schaltungselementen) entsprechend DGl (Modell)

  Rezept:

  - Auflösung DGl (aller DGl eines DGl-Systems) nach (jeweils) höchster Ableitung

  - Blockschaltbild zeichnen (abzeichnen)

  → Analogrechner / Blockdiagramm
  auch "für sich" brauchbar
  als Beschreibung
  siehe 10.1

  erreicht Genauigkeiten ca 10^{-3}, uU 10^{-4}
  (als Hinweis auf erforderliche Präzision
  "ungenauer" numerischer Integration)

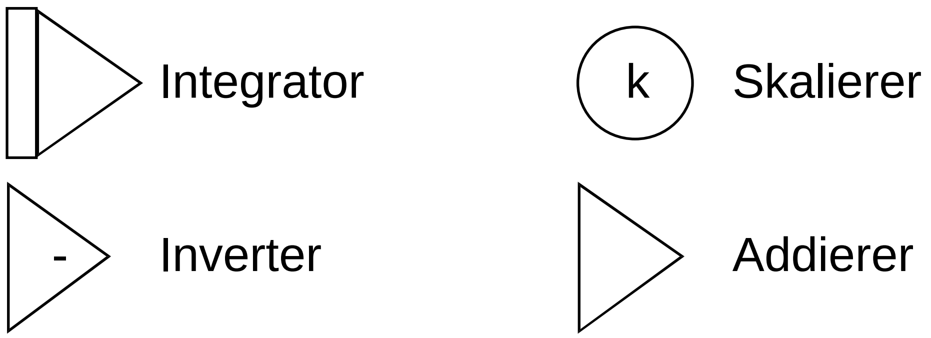
- im Beispiel Radauslenkung
  wird aus

  $$m \ddot{x} + d \dot{x} + f x = E(t)$$

  aufgelöst

  $$\ddot{x} = \frac{1}{m} \left(E(t) - d \dot{x} - f x\right)$$

Schaltungssymbole (nicht völlig einheitlich)



Schaltung für $\ddot{x} = \frac{1}{m}\left(E(t) - d\,\dot{x} - f\,x\right)$

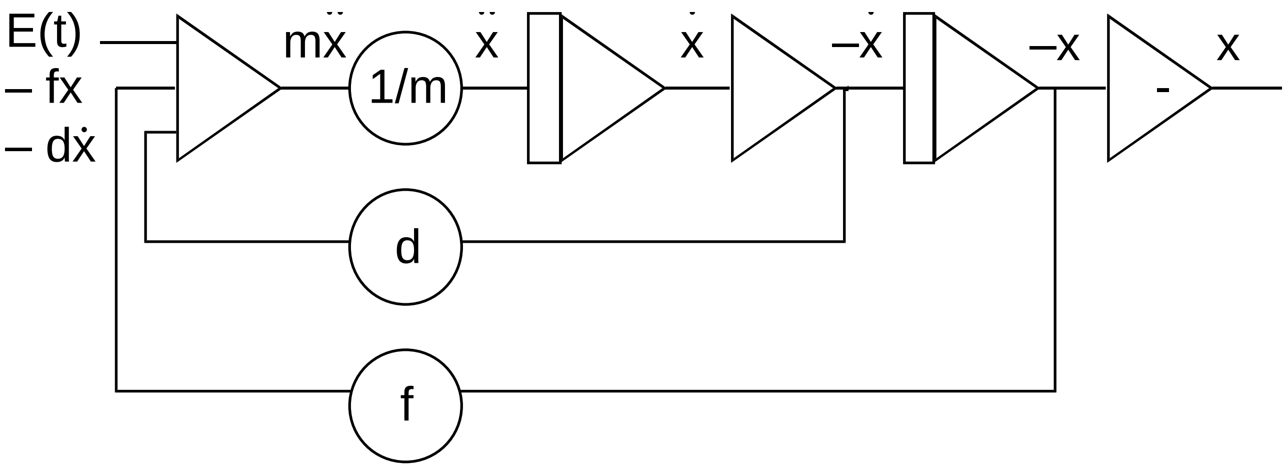


Bild 10.0.5: Schaltung DGl Radauslenkung

Insgesamt genauer zu betrachten

- **Beschreibungsmittel** für
  Spezifikation kontinuierlicher Simulatoren

- **Lösungsverfahren** bei
  Diskretisierung und numerischer Integration

10.1 Spezifikation von kontinuierlichen Modellen

Beschreibungsmittel

- allgemein,
  an berücksichtigte Gleichungsformen angepaßt:

  lineare, nichtlineare, gewöhnliche, partielle, …
  DGl + DGl-Systeme

- an Nutzungsbereich / Benutzerkreis angepaßt
  (Erinnerung: "Szenario")

  sozio-ökonom., technischer, medizinischer, … Bereich

Konzentration (hier)

- auf Systeme
  - gewöhnlicher Differentialgleichungen

- auf wenige Nutzungsbereiche
  - nur beispielhaft

Im Gegensatz zu ereignisorientierten Systemen
bietet klassische Mathematik (Analysis)

für zeit-/zustands-kontinuierliche Systeme

- ausgearbeitete Beschreibungstechniken
- ausgefeilte Lösungs-Kalküle
  (ggf hinsichtlich expliziter Lösungen versagend:
  Ausweg: Simulation)

Modellformulierung "startet" daher

- typischerweise bei mathematischer Formulierung

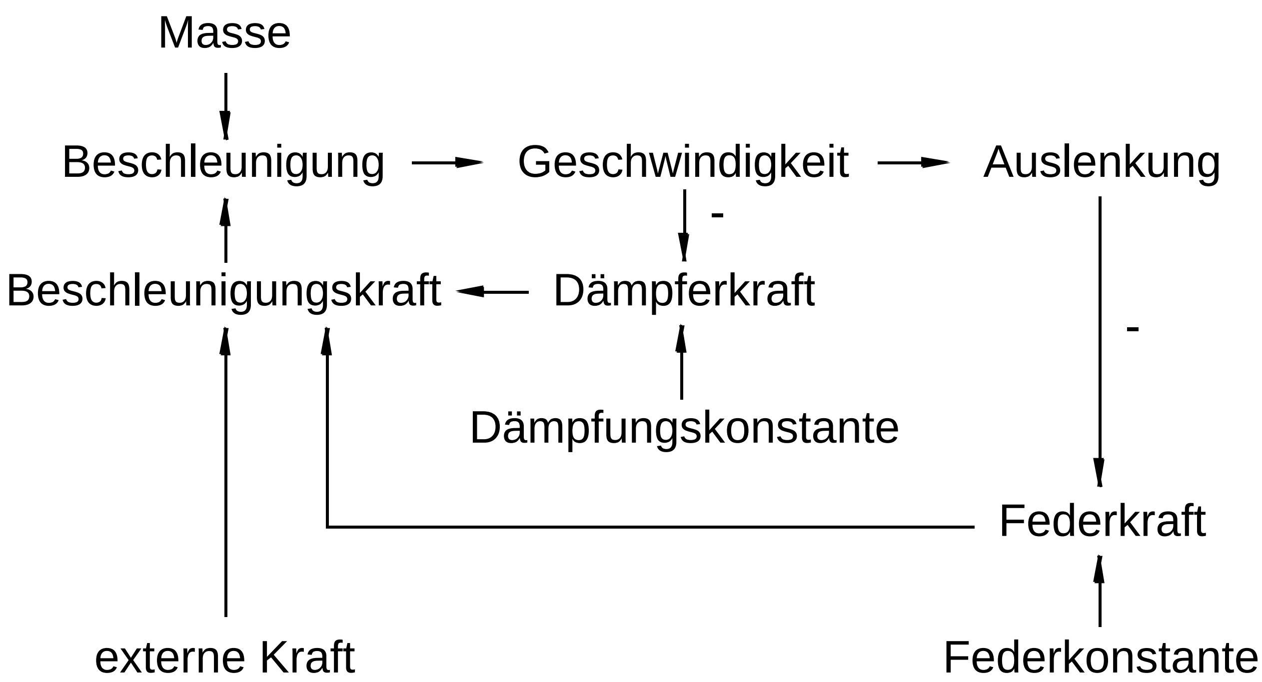
- welche notwendigerweise auf "technischem" Verständnis zu analysierenden Systems beruht
  (zB Mechanik, Regelungstechnik, …, BWL)

Verschiedene initiale Modell"formen",
(soweit vollständig formalisiert:)
untereinander gleichwertig

Es gibt (teilformalisierte) "Vorstufen", zB

- **Wirkungsdiagramme**
  stellen prinzipielle Abhängigkeiten / Einflüsse dar

im Bsp 10.0.03 (hier bis auf weiteres genutzt)



voll formalisierte Formen

- gewöhnliche lineare DGl n.Ordnung
  ("wie gehabt", nicht immer erreichbar)

$$m \ddot{x} + d \dot{x} + f x = E(t)$$

- Sytem linearer DGl 1.Ordnung
  ("allgemein", für Analyse sehr angenehm)

  über Einführung von "Zustandsvariablen",
  entsprechend spezifischen Bezeichnern
  - für "eigentliche" abhängige Variablen
  - für deren Ableitungen nach der Zeit

  im Bsp:

  - 1 abh. Variable, Ordnung 2
    2 Zustandsvariable x_0, x_1

    $x_0 := x$
    $x_1 := \dot{x}$

  - System von DGl 1.Ordnung

    $\dot{x}_0 = x_1$
    $\dot{x}_1 = {}^1\!/_m \left(E(t) - d\, x_1 - f\, x_0 \right)$

    erste Gleichung(en) per def
    letzte Gleichung gemäß DGl n.Ordnung

- System von Integralgleichungen
  ("allgemein", Entsprechung Blockdiagramm)

  durch formale Integration des DGl-Systems 1. Ordnung

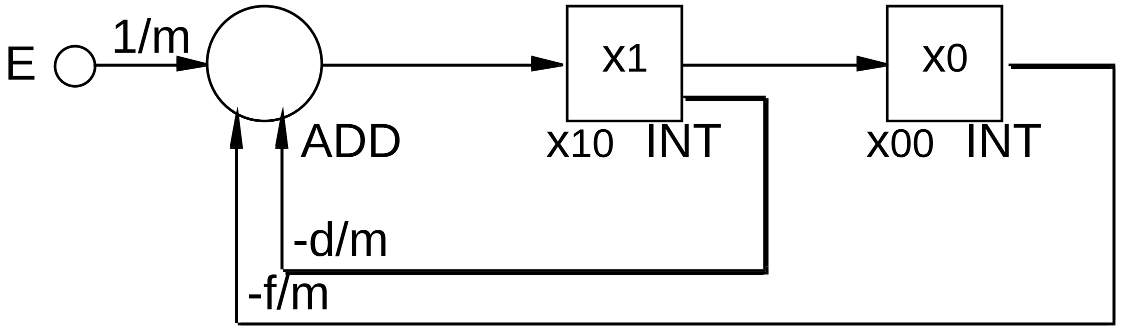
  $$\dot{x}_0 = x_1$$
  $$\dot{x}_1 = {}^1\!/_m \left(E(t) - d\,x_1 - f\,x_0\right)$$

  $\rightarrow$

  $$x_0 = \int_0^t x_1\,dt$$
  $$x_1 = \frac{1}{m}\int_0^t E(t)\,dt - \frac{d}{m}\int_0^t x_1\,dt - \frac{f}{m}\int_0^t x_0\,dt$$

- Blockdiagramm
  (verschiedene Ausprägungen)

  dem Integralgleichungssystem
  (bzw System DGl 1.Ordnung) entsprechend,
  oft "direkt" formuliert



vgl auch Bild 10.0.05

- Programmsprache

  (eine Möglichkeit natürlich:
  in beliebiger HLL ausprogrammieren,
  incl expliziter Programmierung Lösungsverfahren)

  diverse Ansätze von Sprachen:
  - deklarativ als Differenzengleichungssystem,
  - uU incl Lösungsverfahren,
  - direkt vom DGl-System

```
beschl = (E - d*geschw - f*auslenk)/masse
geschw = geschw + DELTA*beschl
auslenk = auslenk + DELTA*geschw
```

  "Reihenfolge" der Anweisungen wesentlich

"problemangepaßte" Formulierungen
lehnen sich an "passende" obiger Alternativen an

große Szenario-Bereiche

- sozio-ökonomische Probleme
  (vgl Beispiel DYNAMO)

- technische Probleme
  (vgl Beispiel CSMP)

- biologisch / medizinische Probleme
  (vgl Beispiele)

- …

BEISPIEL DYNAMO

entwickelt im Zusammenspiel mit Forrester's
System / Industrial / Urban / World Dynamics

aufgrund breit diskutierter (Welt-)Modelle
mit beträchtlicher Bekanntheit
"Club of Rome"

Fähigkeiten / Modelltypen (im Vergleich)
eher begrenzt: Systeme linearer DGl 1.Ordnung
aber Schwergewicht (beabsichtigterweise)
eher bei Prinzip von Wechselwirkungen
als bei quantitativen Zusammenhängen

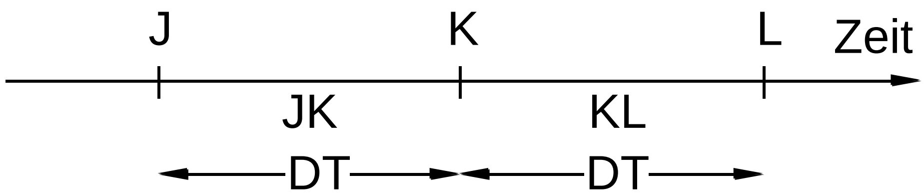
Modellspezifikation auf 2 Ebenen

- spezielle blockorientierte Graphik
  nicht voll formalisiert: à la Wirkungsdiagramme

  führte zur hohen Attraktivität des "x" Dynamics Ansatzes

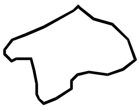
- sprachliche Notation, FORTRAN-basiert,
  deklarativ,
  mit speziellen Notationshilfen für Differenzen-Gl-Systeme

  Zeit strukturiert in 3 Zeitpunkte (+ 2 Zeitintervalle)
  vorher: J jetzt: K nachher: L
  dazwischen: JK bzw KL

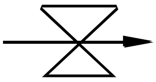


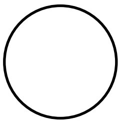
DT konstant, Nutzer-gesetzt

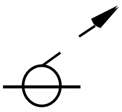
- Symbole der DYNAMO-Diagramme:


Quelle bzw Senke


level (Niveau):Zustandsvariable


rate (Veränderungsrate)


auxiliary (Hilfsvariable)


constant (Konstante)

flow (Veränderungsfluß)

cause-and-effect (Einwirkung)

siehe Beispiel

- Syntaxprinzip DYNAMO-Programme:

```
L  y.K = y.J + (DT)(yrate.JK)
R  yrate.KL = (k1)(H-y.K)
```

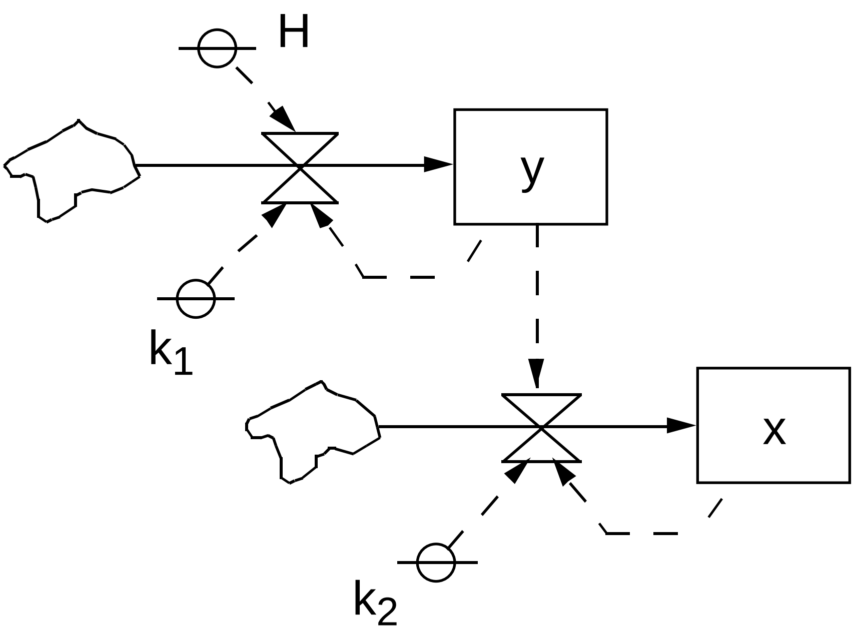
Kennung Gleichungstyp (L: level, R: rate)
bei level Bezug auf Zeitpunkt
bei rate Bezug auf Zeitintervall (Sortierungsproblem beseitigt)
siehe Beispiel

- Verwendungsbeispiel
  am Beispiel 10.0.02: Häuser: y(t)
  + Klimaanlagen: x(t)

  DGI-System war

  $$\dot{y} = k_1 \; (H - y) \qquad y = 0 \text{ für } t = 0$$
  $$\dot{x} = k_2 \; (y - x) \qquad x = 0 \text{ für } t = 0$$

- zugehöriges DYNAMO-Diagramm



- zugehöriger Ausschnitt aus DYNAMO-Programm:

```
L  y.K = y.J + (DT)(yrate.JK)
L  x.K = x.J + (DT)(xrate.JK)
R  yrate.KL = (k1)(H-y.K)
R  xrate.KL = (k2)(y.K-x.K)
```

BEISPIEL CSMP

- eine von vielen
  Continuous System Simulation Languages (CSSLs)
  des technischen Bereichs
  diese idR deutlich mächtiger als (zB) DYNAMO

- historisch bedeutsam,
  hier wegen "typischen" Ansatzes skizziert,
  FORTRAN-geprägt

- folgt den Ideen

  - Nutzung von Zustandsvariablen gern in
    Standard-Notation: `x, xdot, x2dot`

  - in deklarativer Formulierung
    auf Basis Integralgleichungssystem
    mit Notation: `x = INTGRL(init,xdot)`

  - unter interner Setzung Lösungsverfahren

CSMP unterscheidet "statement"-Typen

- strukturelle statements
  definieren "eigentliches" Modell

- Daten-statements
  setzen Werte für Parameter, Konstanten, init-Bedingungen

- Steuerungs-statements
  bestimmen Übersetzung, Ausführung, Ausgaben

CSMP-Fassung Bsp 10.0.03
(mit Kommentaren {…} nicht in CSMP-Syntax)

```
title radauslenkung
*
param d = (5.61,11.82,…)   {Liste Werte}
*
   x2dot = (1.0/m) * (e - d*xdot - f*x)
   xdot = INTGRL(0.0,x2dot {Initialwert 0}
   x = INTGRL(0.0,xdot)    {Initialwert 0}
*
const m=2.0,e=1.0,f=400.0  {ext. Kraft const}
timer delt=0.005,fintim=1.5,…
                 {Diskretisierung, Abbruch,…}
print x,xdot,x2dot         {Ausgaben}
…                          {Ausgabenformat}
end
```

heute gebräuchliche Spezifikationen

- rein sprachliche: recht ähnlich CSMP
  (zB ACSL, DARE, DESIRE, …)

- Betonung Blockdiagrammdenken
  (dann: problemangepaßte Blocktypen
  häufig Regelungstechnik
  zB SIMULINK, …)

- auch in direkter Übersetzung aus Graphik

- widmen sich vermehrt der Modellstrukturierung
  (vgl analoge Überlegungen bei ereignisorient. Simulation)

vgl Spezialliteratur

BEISPIELBEREICH ÖKOSYSTEME

Beispiel "Wirte - Parasiten"

Wirkungsschema (→ Wirkungsdiagramm)

parasitäre Insekte legen Eier in Larven von Wirten
→ Parasiten ↑ , Wirte ↓
→ Parasiten ↓ , Wirte ↑
→ …

mit

W: Zahl Wirte
P: Zahl Parasiten
G: Geburtenüberschuß Wirte (bei normaler Sterberate)
B: Befallskoeffizient Wirte
S: Sterbequote Parasiten

ergibt sich "erstes" Modell

$$\dot{W} = W \quad (G - B\,P)$$
$$\dot{P} = P \quad (B\,W - S)$$

BEISPIELBEREICH BIOMECHANIK

mit sehr unterschiedlichen "Verzweigungen":

→ Prothetik

→ Roboter

BEISPIELBEREICH MEDIZIN

ua "Pharmakakinetik"
(Dynamik der Arzneimittelverteilung)

mit typischen Kompartiments-Modellen
(Organismus in "Kompartimente" aufgeteilt)

je Kompartiment Gleichung des Typs
Konzentrationsveränderung = Zufuhr - Abgabe
Abgabe proportional Konzentration

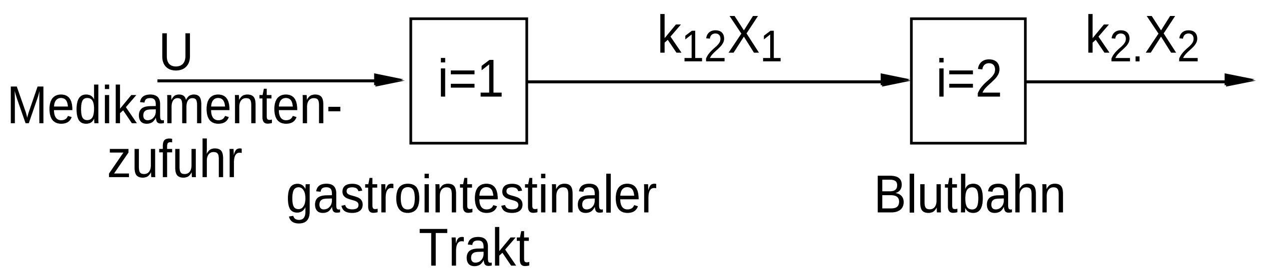
für Kompartiment i:

$$\dot{X}_i = Z_i - k \quad X_i$$

\+ Kopplung der Gleichungen
"gemäß Struktur Organismus"

→ DGl-System

Ausschnitt:



nochmals:

Blockdiagramme / Graphiken
offensichtlich an Anwendungsbereich anzulehnen !

10.2 Numerische Integrationsmethoden
"angerissen"

Zu lösen sei zB:

$dy/dx = f(x,y)$ mit Anfangsbedingungen
$x = x_0,\ y = y_0$

für uns immer $x \rightarrow t$

dh zu lösen (implizite Gleichung):

$$y = y_0 + \int_{x_0}^{x} f(t,y)\, dt$$

Unsere Lösungs-Ahnung ("numerische Integration") war

- unabhängige Variable diskretisieren
  $t_n = t_0 + n \cdot \Delta t$ $n=1,2,\ldots$

- Teilintegrale bilden pro Δt-Intervall
  auf Basis konstanter Steigung ("Rate")
  unter Verwendung der Anfangssteigung

ist bekannt als **Euler-Integration**
("wir waren nicht die ersten")

ist "Einschrittverfahren",
(Mehrschrittverfahren: "predictor-corrector"
versuchen Verbesserung)

ist Runge-Kutta-Verfahren 1.Ordnung
(RK höherer Ordnung versuchen Verbesserung)

- **Euler-Integration** algorithmisch für "Anfangswertproblem"
  $dy/dt = f(t,y)$ Anfangsbedingungen $t=t_0$, $y=y_0$

  - betrachtete Zeitpunkte
    $t_n = t_0 + n \cdot \Delta t$ $n=1,2,\ldots$

  - Initialisierung
    $t=t_0$
    $y=y_0$ $y_0' = f(t_0,y_0)$

  - 1. Schritt
    $t_1 = t_0 + \Delta t$
    $y_1 = y_0 + \Delta t \cdot y_0' = y_0 + \Delta t \cdot f(t_0,y_0)$
    $y_1' = f(t_1,y_1)$

  - 2. Schritt
    $t_2 = t_1 + \Delta t$
    $y_2 = y_1 + \Delta t \cdot y_1' = y_1 + \Delta t \cdot f(t_1,y_1)$
    $y_2' = f(t_2,y_2)$

  - …

Idee von Verfahren steigender Genauigkeit
(ua Runge-Kutta) liegt
in Taylor-Reihen-Entwicklung von y(t)
an Stützpunkten t_n

$$y(t+\Delta t) = y(t) + \Delta t\,\dot{y}(t) + \frac{1}{2}\Delta t^2\,\ddot{y}(t) + \\ + \ldots \frac{1}{k!}\Delta t^k\,y^{(k)}(t) + O(\Delta t^{k+1})$$

und Ermittlung der "nächsten" Veränderung Δy so daß
$y_{n+1} = y_n + \Delta y$

falls $y^{(1)}(t)$, $y^{(2)}(t)$, …, $y^{(n)}(t)$ bekannt,
liegt unmittelbar Approximationsverfahren "n. Ordnung" vor

falls Ableitungen nicht / nur für niedrige Ordnungen bekannt,
müssen höhere Ableitungen numerisch ermittelt werden
(im vorliegenden Fall nur 1.Ableitung exakt bekannt)

zwangsläufig erforderlich:
- bei Verfahren n.Ordnung
- mit k<n exakt bekannten Ableitungen
- ≥ n-k zusätzliche Funktionsberechnungen f(t,y)
  für numerische Ableitungsberechnung

(im vorliegenden Fall für Verfahren 2.Ordnung ≥ 1 Berechng)

zusätzliche Punkte + Verwendung derart daß
n erste Glieder Taylor-Entwicklung exakt reproduziert
(falls vorliegendes Problem Polynom n.Ordnung,
dann ist Verfahren exakt)

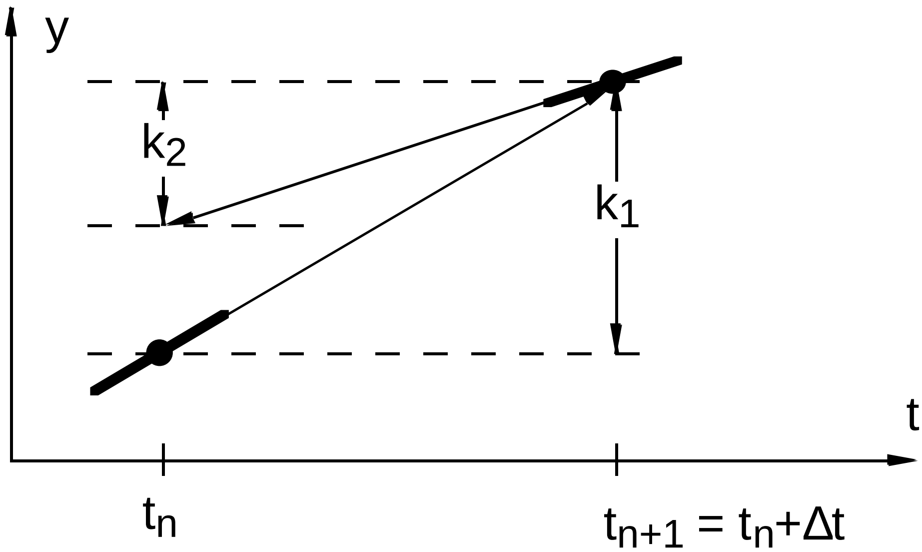
Viele Verfahren !
Skizze vgl später

Vergleichspunkte
- Stabilität
- Genauigkeit
- Effizienz

breite Literatur

bei zu verwendenden Paketen / tools
auf Spektrum implementierter Verfahren achten

Beispiel Runge-Kutta 2.Ordnung (Euler-Cauchy / Heun)



Verfahren

- bei vollem Euler-Schritt ergäbe sich
  $(\Delta y=)\ k_1 = \Delta t \cdot y_n' = \Delta t \cdot f(t_n, y_n)$

- Steigung im erreichten Punkt
  $f(\ t_n + \Delta t\ ,\ y_n + k_1)$

- bei dieser Steigung ergäbe sich
  $(\Delta y=)\ k_2 = \Delta t \cdot f(\ t_n + \Delta t\ ,\ y_n + k_1)$

- verwendet wird
  $\Delta y = 1/2 \cdot (k_1 + k_2)$
  dh
  $y_{n+1} = y_n + 1/2 \cdot (k_1 + k_2)$

Konstruktion Runge-Kutta-Verfahren höherer Ordnung
- aufwendig
- mit jeweils mehreren Lösungen / Ergebnissen

viel gebraucht Runge-Kutta 4.Ordnung

häufigste Varianten
sind

$$y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}$$

$$k_1 = \Delta t \; f(t_n, y_n)$$

$$k_2 = \Delta t \; f(t_n + \frac{\Delta t}{2}, y_n + \frac{k_1}{2})$$

$$k_3 = \Delta t \; f(t_n + \frac{\Delta t}{2}, y_n + \frac{k_2}{2})$$

$$k_4 = \Delta t \; f(t_n + \Delta t, y_n + k_3)$$

und

$$y_{n+1} = y_n + \frac{k_1}{8} + \frac{3k_2}{8} + \frac{3k_3}{8} + \frac{k_4}{8}$$

$$k_1 = \Delta t \; f(t_n, y_n)$$

$$k_2 = \Delta t \; f(t_n + \frac{\Delta t}{3}, y_n + \frac{k_1}{3})$$

$$k_3 = \Delta t \; f(t_n + \frac{2\Delta t}{3}, y_n + k_2 - \frac{k_1}{3})$$

$$k_4 = \Delta t \; f(t_n + \Delta t, y_n + k_3 - k_2 + k_3)$$

Runge-Kutta-Verfahren sind **Einschrittverfahren**

nutzen nur Information (Ableitungsberechnungen)
aus **einem** Intervall (Schritt) $[t_n, t_{n+1}]$

Mehrschrittverfahren
nutzen auch zurückliegende Information
(aus "vorigen" Intervallen)

Prädiktor-Korrektor-Verfahren (ebenfalls viel verwendet)
sind Mehrschrittverfahren

prinzipielles Schema:

- Prädiktor(teil)schritt:

  berechne y^P_{n+1} ("predicted")
  aus äquidistanten Punkten $t_n, t_{n-1}, t_{n-2}, \dots$
  deren Funktionswerten $y_n, y_{n-1}, y_{n-2}, \dots$
  und Ableitungswerten f(.) an diesen Punkten

- Korrektor(teil)schritt:

  berechne Steigung an diesem Punkt
  dh $f(t_{n+1}, y^P_{n+1})$
  und y^C_{n+1} ("corrected")
  aus zuvor genutzter Information
  $+ f(t_{n+1}, y^P_{n+1})$

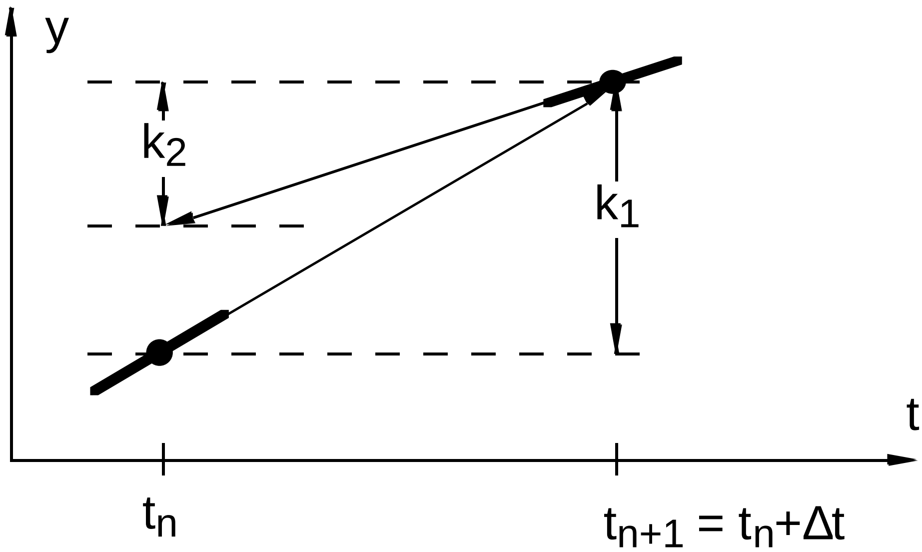
(- Schema ließe sich offensichtlich "iterieren":
lohnt nicht, dh nutze y^C_{n+1} als Basis nächsten Schritts)

Insgesamt wieder Familie von Verfahren

- unterschiedlicher Ordnung / Fehlerordnung
  (je nach "Zahl" genutzter Punkte)

- mit verschiedenen Alternativen / Varianten

nochmals Betrachtung
Beispiel Runge-Kutta 2.Ordnung (Euler-Cauchy / Heun)



Verfahren

- bei vollem Euler-Schritt ergab sich
  $(\Delta y=)\ k_1 = \Delta t \cdot y_n' = \Delta t \cdot f(t_n,y_n)$

  umgedeutet als Prädiktor(teil)schritt:
  $y^P_{n+1} = y_n + \Delta t \cdot f(t_n,y_n)$

  genutzte Information:
  t_n, y_n und Ableitungswert $f(t_n,y_n)$ an diesem Punkt
  (+ "nichts" vorher)

- Steigung im erreichten Punkt ergab sich zu
  $f(t_n+\Delta t\ ,\ y_n+k_1)$

  umgedeutet als Beginn Korrektor(teil)schritt:
  $= f(t_{n+1}\ ,\ y^P_{n+1})$

- bei dieser Steigung ergab sich
  $(\Delta y=)\ k_2 = \Delta t \cdot f(\ t_n+\Delta t\ ,\ y_n+k_1)$
  und verwendet wurde
  $\Delta y = 1/2 \cdot (k_1 + k_2)$
  $y_{n+1} = y_n + 1/2 \cdot (k_1 + k_2)$

  umgedeutet als Fortsetzung Korrektor(teil)schritt:
  $y^C_{n+1} = y_n + \Delta t/2 \cdot \{ f(t_n,y_n) + f(t_{n+1},y^P_{n+1}) \}$

  genutzte Information:
  "zuvor genutzte" + $f(t_{n+1},y^P_{n+1})$

  Heun-Verfahren ("Runge-Kutta 2.Ordnung")
  ist demnach "auch" Prädiktor-Korrektor-Verfahren

entsprechend:

- wo Einschrittverfahren (zB Runge-Kutta)
  im Intervall $[t_n,t_{n+1}]$ äquidistante Zwischenpunkte nutzen

- lassen sich uU durch Uminterpretation daraus
  Prädiktor-Korrektor-Verfahren formen

allgemein aber:

- größere Systematik bei Konstruktion
  von Prädiktor-Korrektor-Verfahren

- unter Einschluß von Fehlerabschätzungen

Nebengesichtspunkt:
Einschrittverfahren sind "selbststartend"
Mehrschrittverfahren benötigen separate Initialisierung

"viele Verfahren":

Klassifikation zB nach

Berücksichtigung zurückliegender Punkte
- Einschrittverfahren
- Mehrschrittverfahren

Einbeziehung zukünftiger Werte
- explizit
- implizit

Konvergenzordnung
- 1. 2. … Ordnung

Schrittweitensteuerung
- feste Schrittweite
- variable Schrittweite

Extrapolation

Ordnungssteuerung
- feste Ordnung
- variable Ordnung

vgl Spezialliteratur

und: auch hier ist der Fall stochastischer Modelle
- wesentlich ("Störungen" zB Wind, Strömung)
- und bearbeitet !

vgl Spezialliteratur