# § 9 Numerisches Lösungsverfahren für Diff.gleichungssysteme 1. Ordnung

Gegeben: System von Diff.gl'en 1. Ordn.

Ziel: Verfahren zur näherungsweisen Lsg per Computer ("numerisches Verfahren", im Gegensatz zu "exaktes Lösungsverfahren")

L. Euler

1707- 1783

$e$

$\pi$

Differential- u. Integralrechng.

Algebra

Hydrodynamik

Elastizität

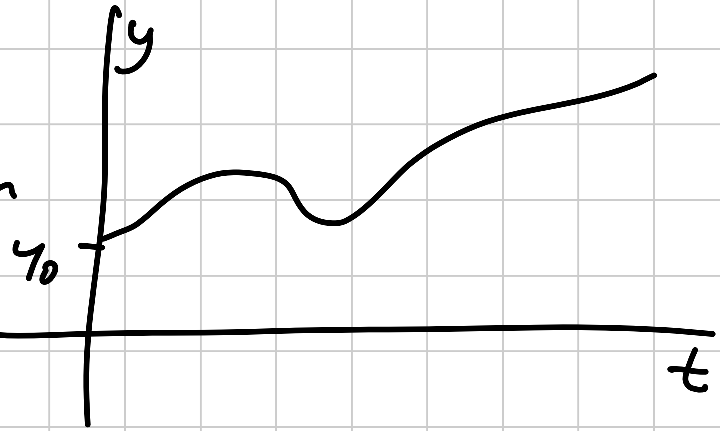
Wirtschaftsmathe

Schach, "Sudoku" (lateinische Quadrate)

Licht, Bau v. Teleskopen, ...

a) einzelne Diff.gl. 1. Ordn.

$$(*) \quad \begin{cases} y'(t) = f(y(t)) \\ y(0) = y_0 \end{cases}, \quad f, y_0 \text{ gegeben}$$



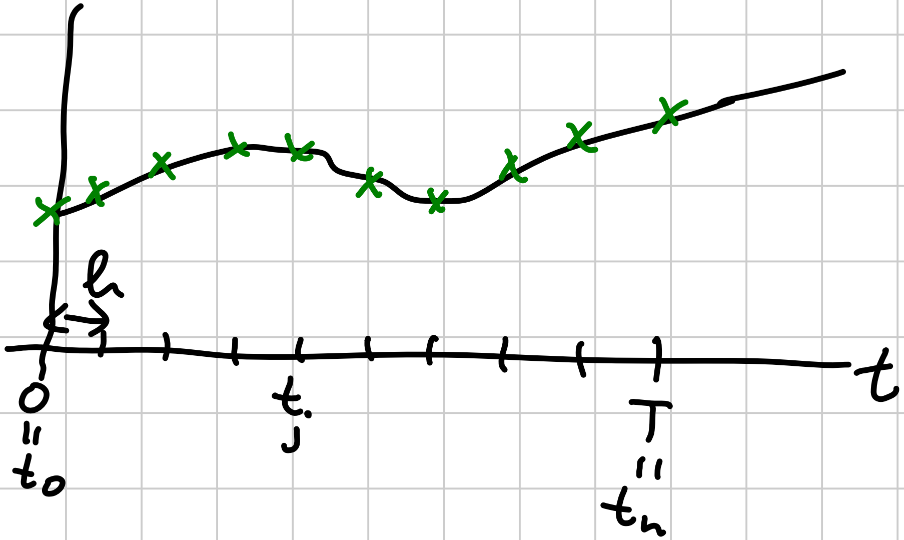

## 1. Zeitachse diskretisieren

Teile Zeitintervall $0 \leq t \leq T$ in $n$ Intervalle der Länge $h = T/n$ ("Schrittweite"), $h$ "hinreichend klein", und betrachte die Zeitpunkte

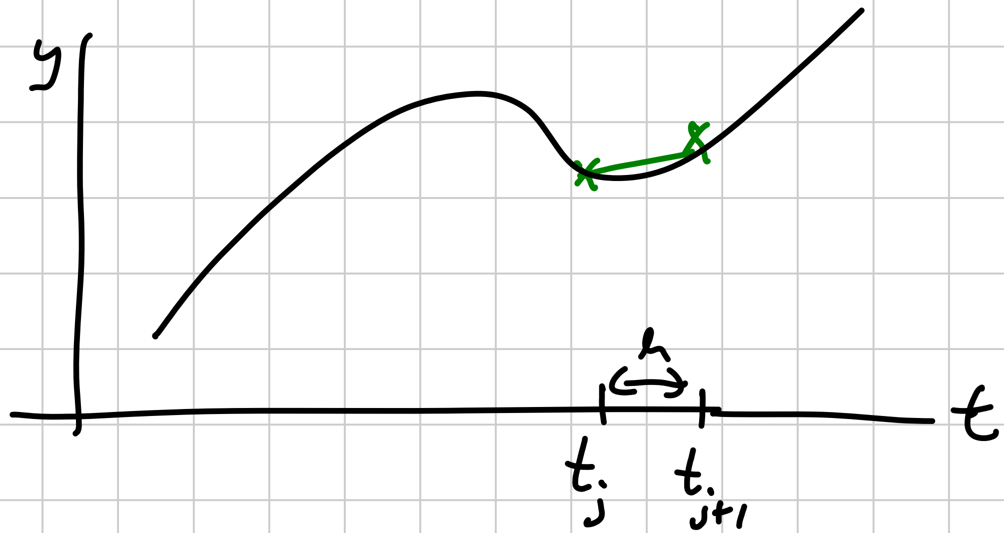
$$t_j = j \cdot h, \qquad j = 0, 1, \ldots, n,$$

sowie die zugehörigen $y$-Werte

$$y(t_j), \qquad j = 0, 1, \ldots, n.$$



## 2. Ableitung durch Differenzenquotienten approximieren




$$y'(t_j) \approx \frac{y(t_j+h) - y(t_j)}{h}$$ (Steigung grüne Strecke)

$$= \frac{y(t_{j+1}) - y(t_j)}{h}$$

Einsetzen in Dgl. $y'(t) = f(y(t))$ bei $t = t_j$

$$\Rightarrow \quad \frac{y(t_{j+1}) - y(t_j)}{h} \approx f(y(t_j))$$

3. Auflösen nach $y(t_{j+1})$ $\quad | \cdot h$

$$\Rightarrow \quad y(t_{j+1}) - y(t_j) \approx h \cdot f(y(t_j)) \quad | + y(t_j)$$

$$\Rightarrow \quad \boxed{y(t_{j+1}) \approx y(t_j) + h \cdot f(y(t_j))} \quad (*)_h$$

- Aus Kenntnis von $y(t_j)$ lässt sich $y(t_{j+1})$ (näherungsweise) bestimmen
- Durch wiederholtes Anwenden von $(*)$ lässt sich $y(t_j)$ (näherungsweise) für alle Zeitpunkte $t_j = j \cdot h$, $j = 0, 1, 2, \ldots$, bestimmen.

Numerisches Verfahren (Explizites Euler-Verfahren):

$$y_h :\equiv \text{exakte Lösung von } \begin{cases} (*)_h \\ y_h(t_0) = y_0 \end{cases}$$

(Lsg von $(*)$ löst $(*)_h$ näherungsweise
$\Updownarrow$
Lsg von $(*)_h$ löst $(*)$ näherungsweise)

Genauer:

$$y_h(t_0) = y_0 \qquad \text{Anfangsbed.}$$

$$y_h(t_1) = y_h(t_0) + h \cdot f(y_h(t_0))$$

$$y_h(t_2) = y_h(t_1) + h \cdot f(y_h(t_1))$$

$$\left(= y_0 + h \cdot f(y_0) + h \cdot f\left(y_0 + h \cdot f(y_0)\right)\right)$$

usw.

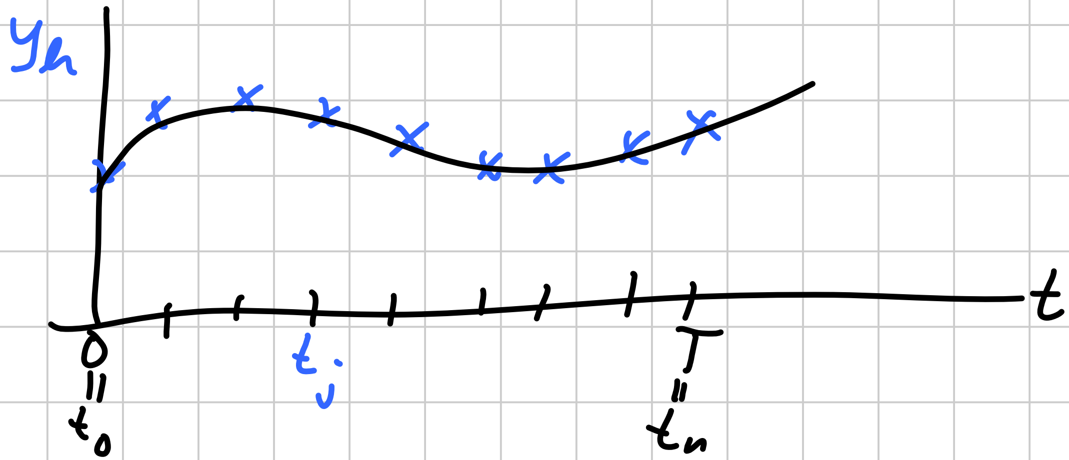
Übliche Schreibweise:

$$\boxed{\begin{cases} y_h(t_0) = y_0 \\ y_h(t_{j+1}) = y_h(t_j) + h \cdot f(y(t_j)), \quad j = 0, 1, \dots, n-1 \end{cases}}$$

(Explizites Euler-Verfahren.)

Datenstruktur:

Die Lösung $y_h$ des Euler-Verfahrens ist keine kontinuierliche Fkt. der Zeit, sondern:

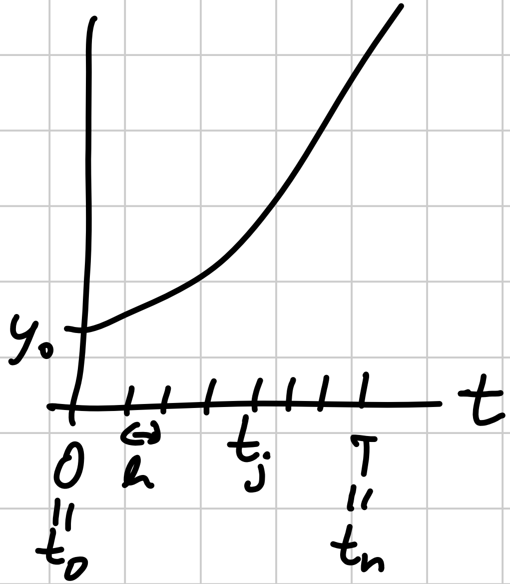



$$y_h = (y_h(t_0), y_h(t_1), y_h(t_2), \dots, y_h(t_n))$$

= geordnete Liste von Zahlen

= <u>Vektor</u>

**Bsp** $f(y) = y$ $\qquad \begin{cases} y'(t) = y(t) \\ y(0) = y_0 \end{cases}$




Lösung in diesem Fall bekannt:

$$y(t) = y_0 e^t$$

Näherungslösung via Euler-Verfahren:

- Teile $0 \leq t \leq T$ in $n$ Teilintervalle auf,
  Schrittweite: $h = \frac{T}{n}$,
  $t_0 = 0,\ t_1 = h,\ \ldots,\ t_j = j \cdot h, \ldots, t_n = T, \qquad j = 0, \ldots, n$

- $y_h(t_0) = y_0 \qquad (0)$

$$y_h(t_1) = y_h(t_0) + h \cdot f(y_h(t_0))$$
$$= y_0 + h \cdot y_0 \qquad (\text{denn } f(y_h(t_0)) = y_h(t_0))$$
$$= (1+h) \cdot y_0 \qquad (1)$$

$$y_h(t_2) = y_h(t_1) + h \cdot f(y_h(t_1))$$

$$= (1+h)\cdot y_h(t_1) \quad \underbrace{\phantom{xxxxx}}_{= y_h(t_1)}$$

$$\underset{(1)}{=} (1+h)\cdot(1+h)\cdot y_0 = (1+h)^2\cdot y_0$$

usw.

$$\Rightarrow \quad y_h(t_j) = (1+h)^j y_0,$$

$$\Rightarrow \quad y_h(T) \underset{T=t_n}{=} (1+h)^n \cdot y_0$$

$$= \left(1+\frac{T}{n}\right)^n \cdot y_0$$

Spezialfall:

$T=1, y_0=1$

$$y_h(1) = \left(1+\frac{1}{n}\right)^n$$

Exakte Lösung der Dgl. $y'=y$, $y(0)=1$:

$$y(1) = e^1 = e$$

Tatsache (Euler, Bernoulli, ...): $\lim_{n\to\infty} \left(1+\frac{1}{n}\right)^n = e$

(Euler'sche Zahl)

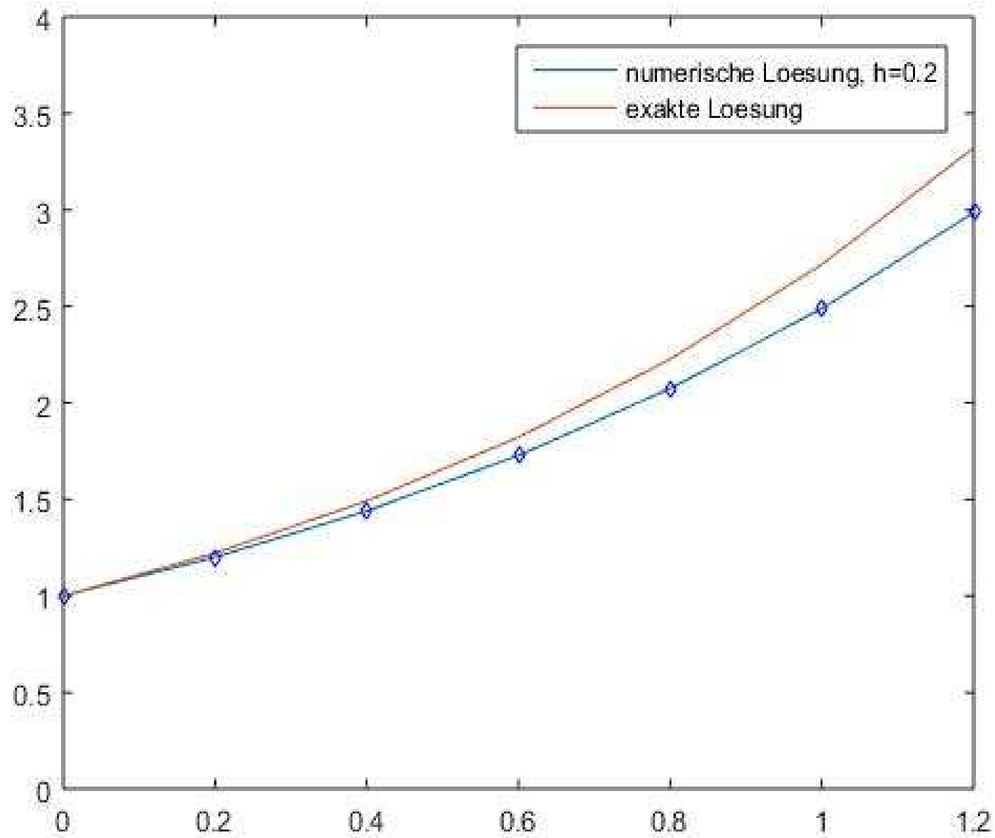
Allgemeiner: Für eine große Klasse von Funktionen $f$ gilt

$$\lim_{h \to 0} y_h(T) = y(T)$$

Lösung des Euler-Verfahrens mit Schrittweite $h$

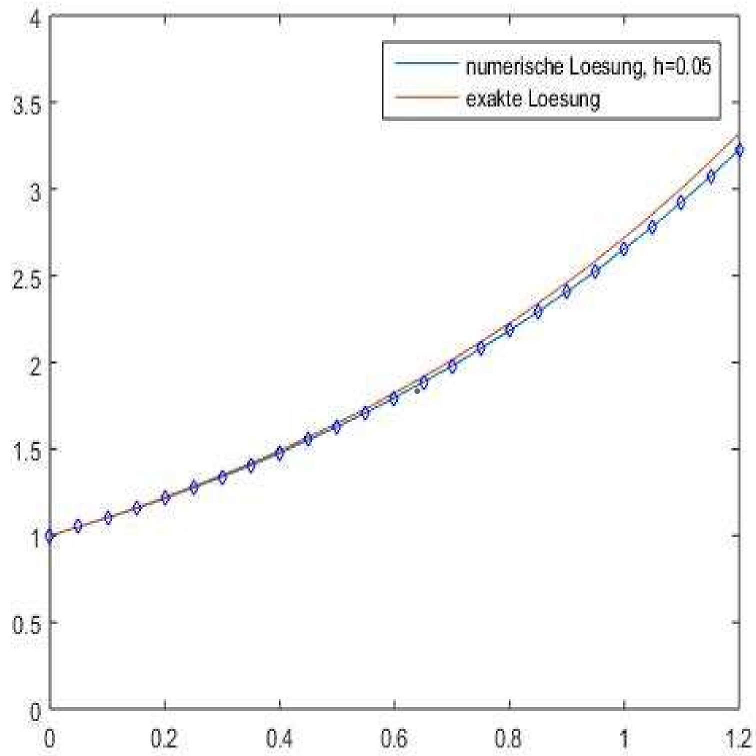
Lösung der Diff.gleichung

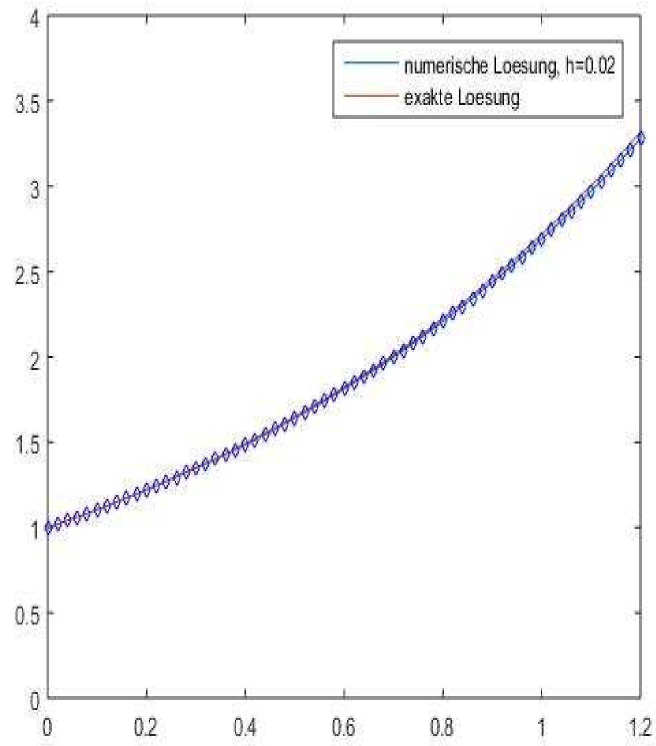
Bsp: Numerische Lsg von $y' = y$, $y(0) = 1$ via explizites Euler-Verfahren (siehe MATLAB-worksheet)




Bei großer Schrittweite $(h=0{,}2)$ sieht man deutliche Abweichungen der numerischen von der exakten Lösung.

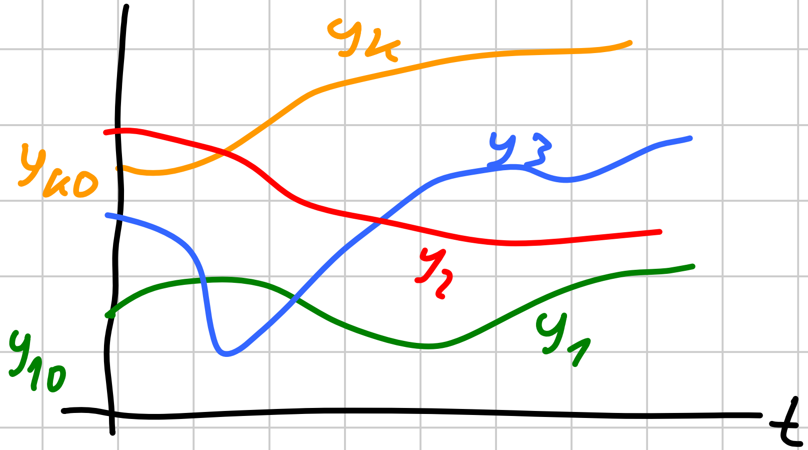
Bei kleiner Schrittweite ($h=0.05$ bzw. $0.02$) ist die numerische Lsg eine sehr gute Näherung der exakten Lsg.



## b) Systeme von Diff.gleichungen 1. Ordnung

$$\begin{cases} y_1'(t) = f_1(y_1(t), \dots, y_k(t)) \\ y_2'(t) = f_2(y_1(t), \dots, y_k(t)) \\ \vdots \\ y_k'(t) = f_k(y_1(t), \dots, y_k(t)) \end{cases}$$




$$y_1(0) = y_{10}$$

$$\vdots$$

$$y_k(0) = y_{k0}$$

**Vektorschreibweise:**

$$y(t) = \begin{pmatrix} y_1(t) \\ \vdots \\ y_k(t) \end{pmatrix}$$

Vektor (= geordnete Liste von Zahlen)

$$f(y(t)) = \begin{pmatrix} f_1(y_1(t), \dots, y_k(t)) \\ \vdots \\ f_k(y_1(t), \dots, y_k(t)) \end{pmatrix}$$

Vektoren werden komponentenweise abgeleitet und addiert

$$y(t) = \begin{pmatrix} y_1(t) \\ \vdots \\ y_k(t) \end{pmatrix}, \quad z(t) = \begin{pmatrix} z_1(t) \\ \vdots \\ z_k(t) \end{pmatrix} \Rightarrow y(t) + z(t) = \begin{pmatrix} y_1(t) + z_1(t) \\ \vdots \\ y_k(t) + z_k(t) \end{pmatrix}$$

$$y'(t) = \begin{pmatrix} y_1'(t) \\ \vdots \\ y_k'(t) \end{pmatrix}$$

<u>Bsp</u> $y(t) = \begin{pmatrix} e^t \\ t^2 \end{pmatrix} \Rightarrow y'(t) = \begin{pmatrix} e^t \\ 2t \end{pmatrix}$

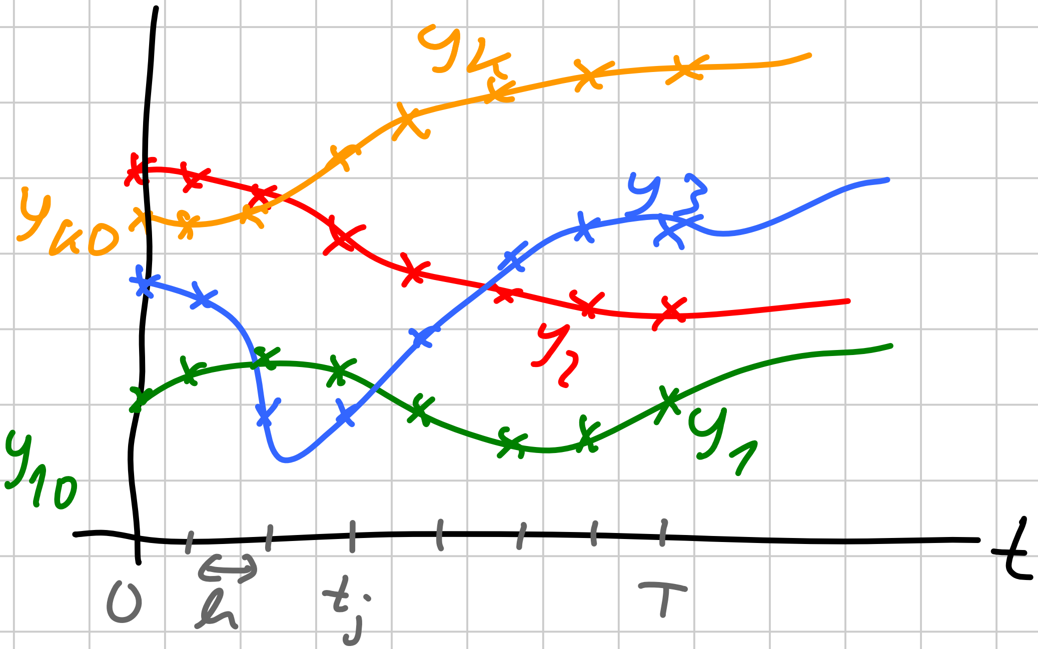
$\rightsquigarrow$ $y'(t) = f(y(t))$

$y(0) = y_0, \quad y_0 = \begin{pmatrix} y_{10} \\ \vdots \\ y_{k0} \end{pmatrix}$ Startvektor

<u>Eulerverfahren</u>: $t_j = h \cdot j$, $j = 0, \dots, n$, $h = \frac{T}{n}$ Schrittweite

$$y^{(h)}(0) = y_0 \qquad \text{(Startvektor)}$$

$$y^{(h)}(t_{j+1}) = y^{(h)}(t_j) + h\, f\big(y^{(h)}(t_j)\big), \quad j = 0, \dots, n-1$$




$$y^{(h)} = \begin{pmatrix} y_1^{(h)}(t_0) & y_1^{(h)}(t_1) & \cdots & y_1^{(h)}(t_n) \\ \vdots & & & \vdots \\ y_k^{(h)}(t_0) & y_k^{(h)}(t_1) & \cdots & y_k^{(h)}(t_n) \end{pmatrix}$$

Lösung $y^{(a)}$ des Euler-Verfahrens
= 2D Liste von Zahlen = Daten-Array = Daten-Matrix

Zeilen = Näherungslösg. für einzelne Dgl'en, aber alle Zeiten
Spalten = Näherungslsg. für alle Dgl'en zu festen Zeitpkt

Bsp: Numerische Lsg via Euler-Verfahren, Enzymkinetik-System aus VL10 (siehe Matlab Worksheet)