Induktion: $\text{z}: \; y_k = \sum_{j=0}^{k} \frac{(At)^j}{j!} y_0$

DON'T: $y_1 = \left(\sum_{j=0}^{1} \frac{(At)^j}{j!}\right) y_0, \; y_2 = \left(\sum_{j=0}^{2} \frac{(At)^j}{j!}\right) y_0$

Also gilt

$$y_k = \sum_{j=0}^{k} \frac{(At)^j}{j!} y_0$$

DO: $y_1 = \left(\sum_{j=0}^{1} \frac{(At)^j}{j!}\right) y_0 \quad \checkmark$

Augenommen die Gleichung gilt für k, dann gilt

$$y_{k+1} = \ldots \underset{\uparrow}{=} \sum_{j=0}^{k+1} \frac{(At)^j}{j!} y_0$$

induktionsschritt

MATLAB

```matlab
%Integrand
integrand = @(x) x.*cos(x);
integrand = @(x) (x-a).^(7/3);

%Referenzloesung
refsol = integral(integrand,a,b);

%Anzahl von Subintervallen
N = 2.^(0:6);

%Alloziere Speicherplatz
approxsol = zeros(1,length(N));

for ii = 1:length(N)

    %Bestimme approximierte Loesung mit N(ii) Subintervallen
    approxsol(ii) = quadratur(integrand,a,b,N(ii));

end

%Bestimme den absoluten Fehler
err = abs(approxsol - refsol);

%loglogplot
figure;
loglog(N,err,'-*')

%Bestimme die Konvergenzrate
p = polyfit(log(N), log(err), 1);
fprintf('Die Konvergenzordnung lautet %1.2f \n',-p(1))
```

TF? →

Wenn man über Konv. ordnung / rate spricht betrachten wir:

$$\underbrace{|Q(I) - I|}_{\text{Fehler / err}} \leq C N^{\alpha}, \quad (\text{für } N \text{ gross genug})$$

Dazu arbeiten wir log logs-plots, dies gibt:

$$\log(|Q(I) - I|) \underset{\approx}{\leq} \log(C N^{\alpha}) = \log(C) + \alpha \log(N) \quad (= mx + q)$$

Das ist ein lineares Polynom in Variable $\log(N)$. $(\alpha x + \log(C))$

Also rufen wir polyfit auf $\log(N)$ und $\log(|Q(I) - I|)$ auf und kriegen

$$(\alpha, \log(C))$$

Runge-Kutta-Methoden

- ist populär
- NM zur Integration gewöhnlicher DGL.

Idee: f bei sorgfältig ausgewählten Stellen auswerten $\longrightarrow$ Approximation zu kriegen, die gleich gut ist wie high-order Taylor

OHNE Ableitungen zu berechnen.

s-stufiges RK:

$$x^{j+1} = x^j + h\sum^{s} b_i k_i \quad , \quad k_i = f(t_j + c_i h, y_j + h\sum_{\ell=1}^{s} a_{i,\ell} k_\ell)$$

$$y^{j+1} = y^j + h\sum_{i}^{s} b_i k_i$$

Bis jetzt waren die meisten NM one-step RK methoden

Die generelle n-step RK Methode:

$$\sum_{j=0}^{n} \alpha_j x^{k+j} = h\sum_{j=0}^{n} \beta_j f(t_{k+1}, x^{k+j}) \quad , \alpha_n \neq 0$$

Falls $\beta_n = 0$, dann erhält man x^{k+n} explizit von den vorherigen x^j und $f(t_j, x^j)$

dann nennt man diese Methode explizit.

Bei $\beta_n \neq 0$, dann nennt man sie implizit.

Bsp

(i) Die four-step Adams-Bashforth Methode

$$x^{k+4} = x^{k+3} + \frac{(\Delta t)}{24}\left[55f\left(t_{k+3}, x^{k+3}\right) - 59f\left(t_{k+2}, x^{k+2}\right) + 37f\left(t_{k+1}, x^{k+1}\right) - 9f\left(t_k, x^k\right)\right]$$

ist ein Beispiel von einer expliziten four-step Methode;

(ii) Die two-step Adams-Moulton Methode

$$x^{k+2} = x^{k+1} + \frac{(\Delta t)}{12}\left[5f\left(t_{k+2}, x^{k+2}\right) + 8f\left(t_{k+1}, x^{k+1}\right) + f\left(t_k, x^k\right)\right]$$

ist ein Beispiel von einer impliziten two-step Methode.

RK: $x^{j+1} = x^j + h\sum^{s} b_i k_i$, $k_i = f\left(t_j + c_i h, y_j + h\sum_{l=1}^{s} a_{i,l} k_l\right)$

Butcher-Tableau (BT)

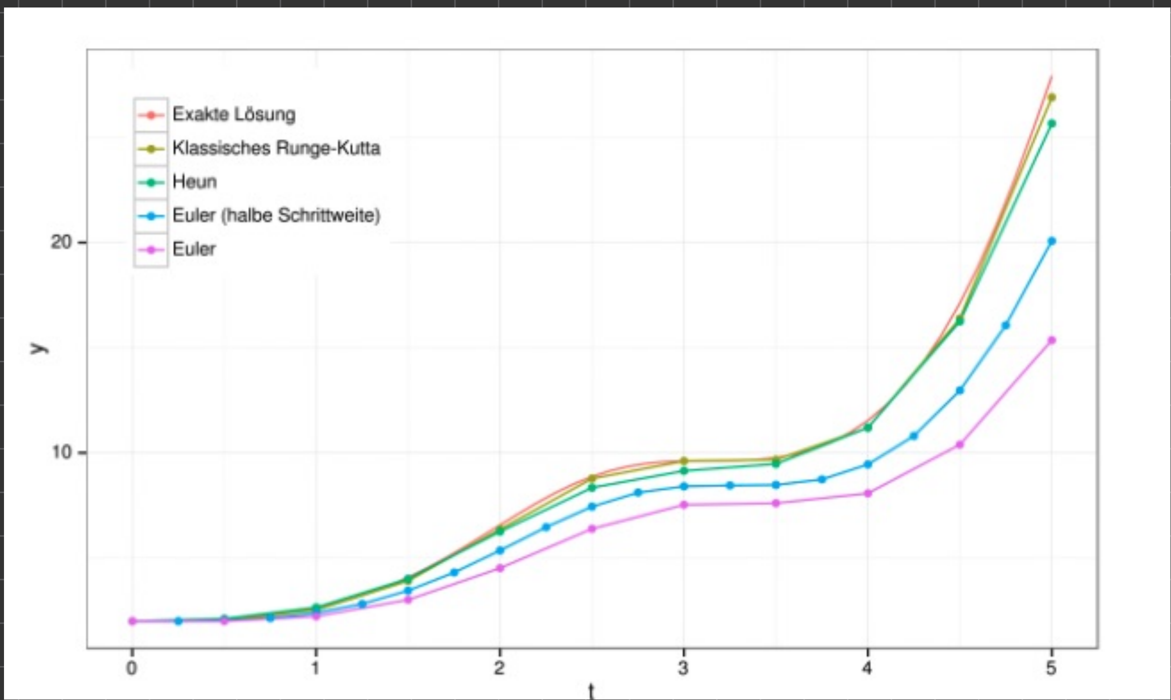
BT sind von der Form

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

Die klassische RK (The RK4)

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Vergleich



DGL: $y' = \sin(t)^2 y$