Anwendung Stetiger Runge-Kutta Verfahren auf die Pantographengleichung

Hans Irl

LMU Munich, Germany

Hütte, den 14.12.2012

Runge-Kutta Verfahren für Gewöhnliche DGL

Es soll ein Runge-Kutta Verfahren (A, b)

$$\left\{ \begin{array}{l} K_{n+1}^i = f\left(t_{n+1}^i,\ y_n + h_{n+1} \sum_{j=1}^{s} a_{ij} K_{n+1}^j \right), \quad i = 1, \ldots, s \\ \\ y_{n+1} = y_n + h_{n+1} \sum_{i=1}^{s} b_i K_{n+1}^i \end{array} \right\}$$

mit $t_{n+1}^i = t_n + c_i h_{n+1}$ auf eine Gewöhnliche DGL

$$\left\{ \begin{array}{l} y'(t) = f(t, y(t)) \quad \text{für} \quad t \geq t_0 \\ y(t_0) = y_0 \end{array} \right\}$$

angewendet werden.

Auf welcher Idee beruhen Runge-Kutta Verfahren?

Das Lösen des Anfangswertproblems einer Gewöhnlichen DGL

$$\left\{ \begin{array}{ll} y'(t) = f(t, y(t)) & \text{für} \quad t \geq t_0 \\ y(t_0) = y_0 & \end{array} \right\}$$

ist äquivalent zum Lösen der Integralgleichung vom Volterra'schen Typ

$$y(t) = y_0 + \int_{t_0}^{t} f(s, y(s)) \mathrm{d}s.$$

Diskretisierung der Integralgleichung führt auf

$$y_1 = y_0 + \left(\sum_{i=1}^{s} b_i K_1^i \right) h_1.$$

Runge-Kutta Verfahren für Gewöhnliche DGL

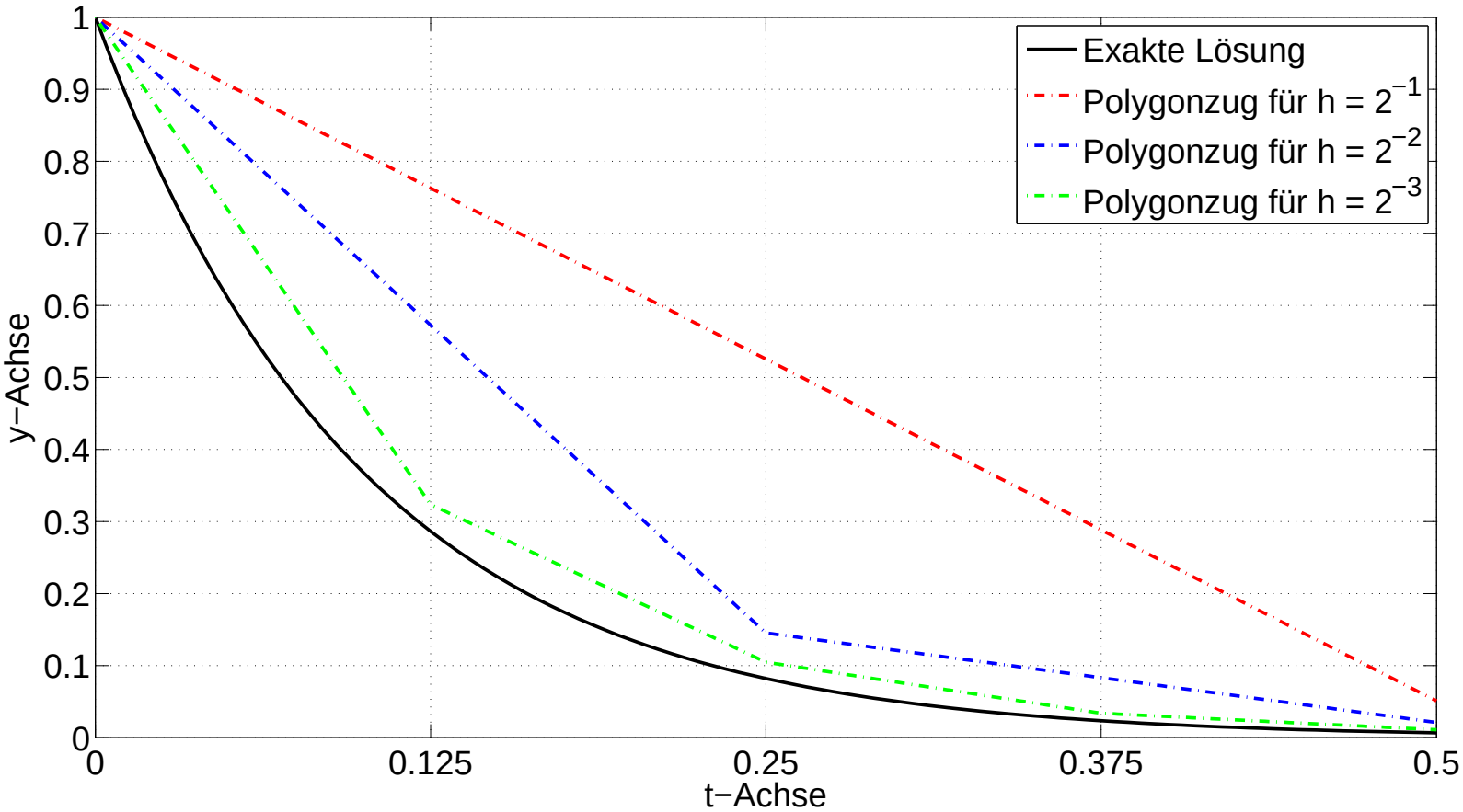

Abb.. *Konvergenz numerischer Lösungen von* $y' = -10y$

Stetige RKV für Gewöhnliche DGL

Es wird nun ein **Stetiges Runge-Kutta Verfahren** $(A, b(\theta))$

$$\left\{ \begin{array}{l} K_{n+1}^i = f\left(t_{n+1}^i,\ y_n + h_{n+1} \sum_{j=1}^{s} a_{ij} K_{n+1}^j \right), \quad i = 1, ..., s \\ \\ \eta(t_n + \theta h_{n+1}) = y_n + h_{n+1} \sum_{i=1}^{s} b_i(\theta) K_{n+1}^i \end{array} \right\}$$

mit $i = 1, ..., s$, $t_{n+1}^i = t_n + c_i h_{n+1}$ und $0 \leq \theta \leq 1$ auf eine GDGL

$$\left\{ \begin{array}{l} y'(t) = f(t, y(t)) \quad \text{für} \quad t \geq t_0 \\ y(t_0) = y_0 \end{array} \right\}$$

angewendet.

Stetige Runge-Kutta Verfahren für Gewöhnliche DGL

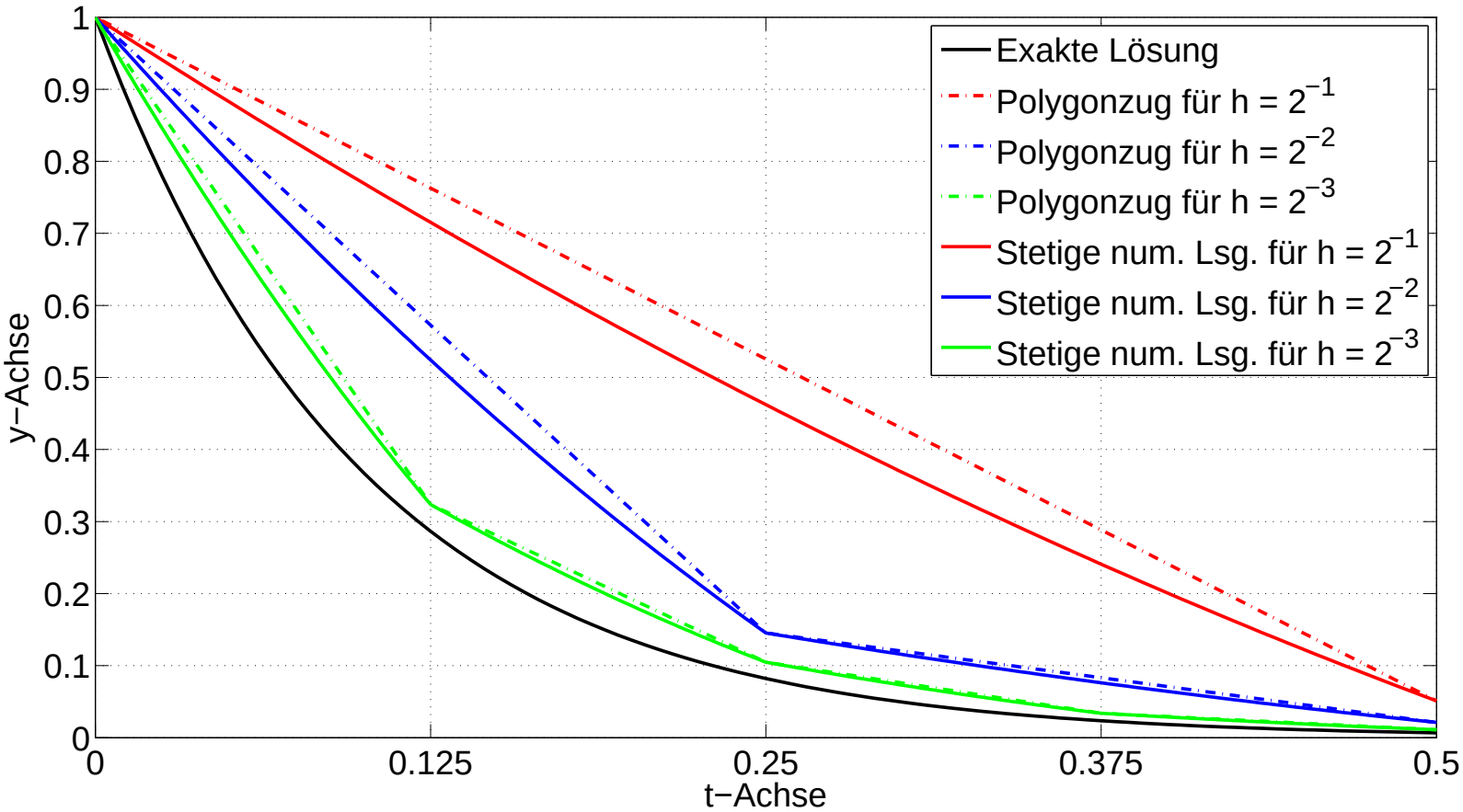

Abb.. *Konvergenz stetiger numerischer Lösungen für* $y' = -10y$

Runge-Kutta Verfahren für Gewöhnliche DGL - Fehler

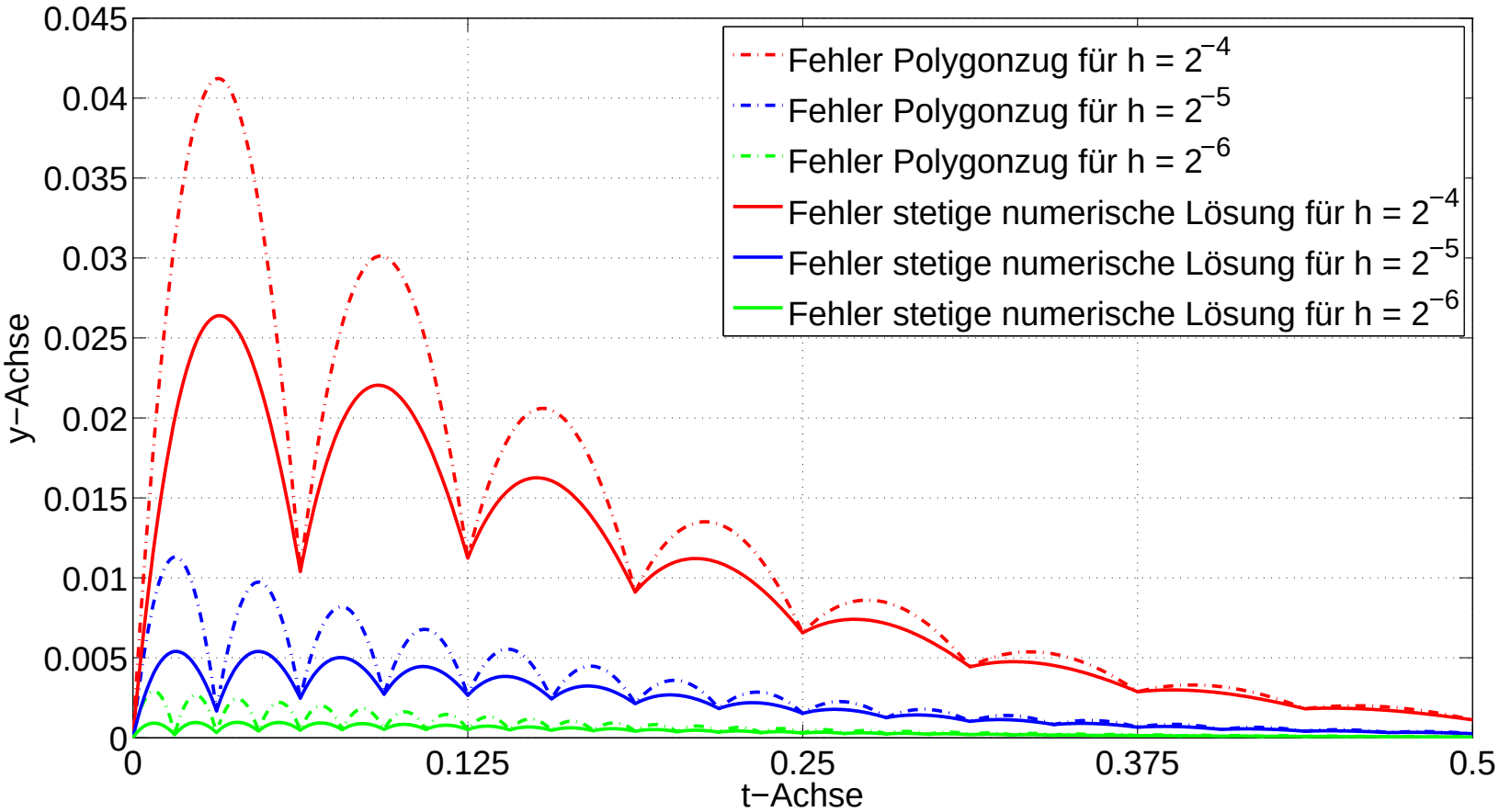

Abb.. *Konvergenzfehler numerischer Lösungen für* $y' = -10y$

Runge-Kutta Verfahren für Gewöhnliche DGL - Fehler

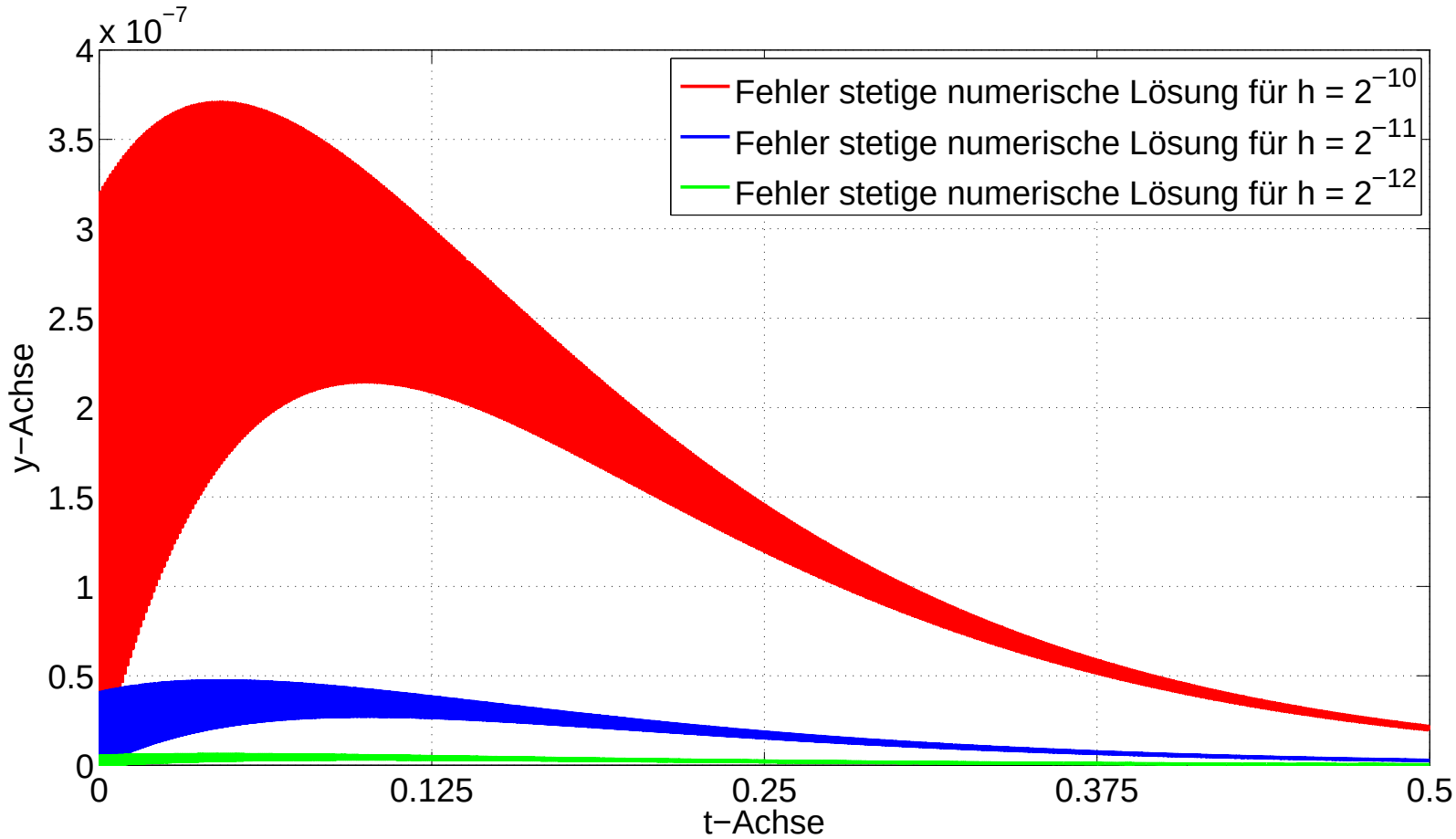

Abb.. *Konvergenzfehler numerischer Lösungen für* $y' = -10y$

Verifizierung der Konvergenzordnung

Nimmt man an, dass

$$\frac{\left|\text{Fehler}\left(h=2^{i-1}\right)\right|}{\left|\text{Fehler}\left(h=2^{i}\right)\right|} \approx \frac{1}{2^{p}} = 2^{-p},$$

so gilt für die Differenzen der logarithmierten Fehlerbeträge

$$-\log_2\left(\left|\text{Fehler}\left(h=2^{i-1}\right)\right|\right) - \left(-\log_2\left(\left|\text{Fehler}\left(h=2^{i}\right)\right|\right)\right) \approx p.$$

Verifizierung der Konvergenzordnung via `MatLab`:

$-\log_2(h)$	10.0000	11.0000	12.0000	13.0000
$-\log_2\left(\left\|\text{Fehler}\left(h=2^{i-1}\right)\right\|\right)$	22.1536	25.1232	28.1078	31.1000
Δ		2.9695	2.9846	2.9923

Konvergenz in einem Schritt (Konsistenz)

Wird nur der Fehler im ersten Integrations bei verschiedenen Schrittweiten betrachtet, so ergibt sich via `MatLab`:

$-\log_2(h)$	10.0000	11.0000	12.0000	13.0000
$-\log_2\left(\left\|\text{Fehler}\left(h=2^{i-1}\right)\right\|\right)$	27.4031	31.3656	35.3467	39.3371
Δ		3.9625	3.9811	3.9905

Anwendung Stetiger RKV auf die Pantographengleichung

Es wird nun ein **Stetiges Runge-Kutta Verfahren** $(A, b(\theta))$

$$\left\{ \begin{array}{l} K_{n+1}^i = f\left(t_{n+1}^i,\ y_n + h_{n+1} \sum_{j=1}^{s} a_{ij} K_{n+1}^j \right), \quad i = 1, \ldots, s \\ \eta(t_n + \theta h_{n+1}) = y_n + h_{n+1} \sum_{i=1}^{s} b_i(\theta) K_{n+1}^i \end{array} \right\}$$

mit $t_{n+1}^i = t_n + c_i h_{n+1}$ und $0 \leq \theta \leq 1$ auf die

Verallgemeinerte Pantographengleichung

$$\left\{ \begin{array}{l} y'(t) = Ly(t) + My(rt) + Ny'(rt) \quad \text{für} \quad t \geq 0 \\ y(0) = y_0 \end{array} \right\}$$

mit $r \in (0,1)$, $L, M, N \in \mathbb{C}^{d \times d}$ und $y_0 \in \mathbb{C}^d$ angewendet.

Geometrisches Gitter für $r = 0.5,\ t_f = 2,\ h_1 = 0.25$

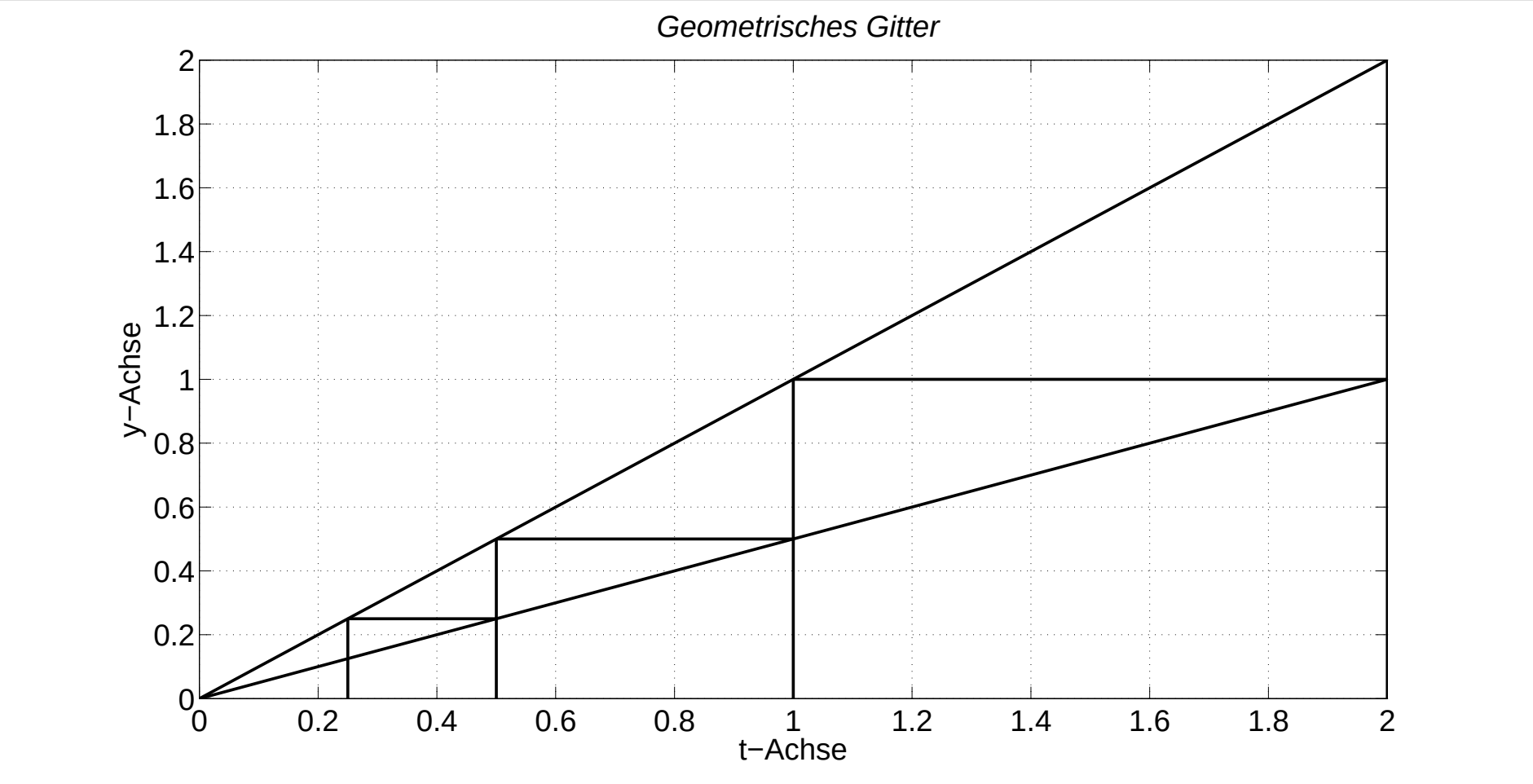

Abb.. *Geometrisches Gitter mit $M = 3$ Makrointervalle*

Quasi-Geometrisches Gitter für $r = 0.5$, $t_f = 2$, $h_1 = 0.25$

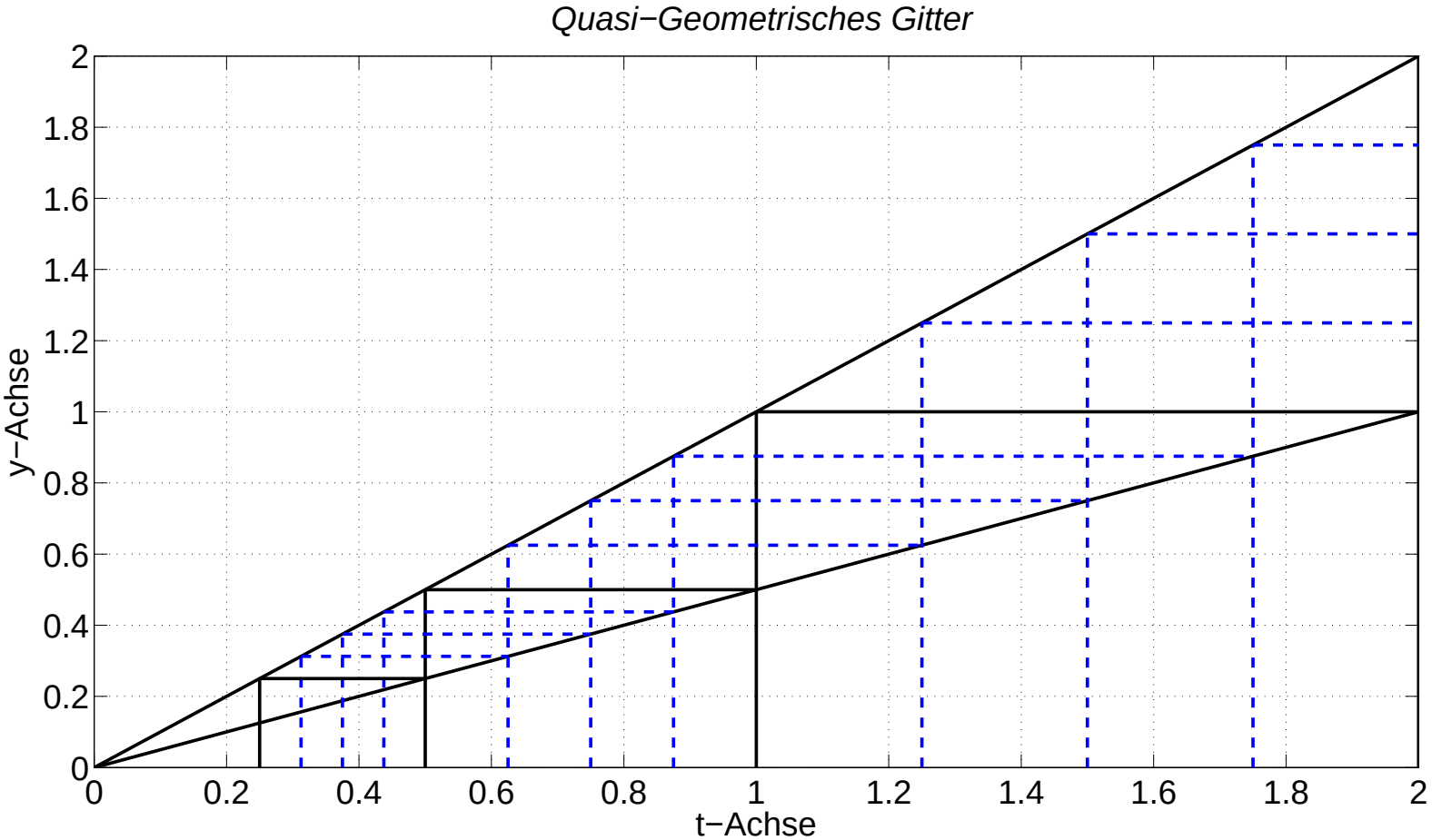

Abb.. *QGG mit $M = 3$ Makro- und je $m = 4$ Mikrointervalle*

Integration über Mikrointervall $[t_n,\ t_{n+1}]$ für $n \geq 1 + m$

Auf diesem Gitter schreibt sich das Stetiges RKV wie folgt:

$$K_{n+1}^i = L\left(y_n + h_{n+1}\sum_{j=1}^{s} a_{ij}K_{n+1}^j\right) + M\left(y_{n-m} + h_{n-m+1}\sum_{j=1}^{s} b_j(c_i)K_{n-m+1}^j\right) + N\left(\sum_{j=1}^{s} b_j'(c_i)K_{n-m+1}^j\right) \quad \text{für} \quad i = 1, ..., s,$$

$$y_{n+1} = y_n + h_{n+1}\sum_{i=1}^{s} b_i K_{n+1}^i.$$

Erster Integrationsschritt (Initialisierungsschritt)

$$K_1^i = L\left(y_0 + h_1 \sum_{j=1}^{s} a_{ij} K_1^j\right) + M\left(y_0 + h_1 \sum_{j=1}^{s} b_j(?) K_1^j\right) + N\left(\sum_{j=1}^{s} b_j'(?) K_1^j\right) \quad \text{für} \quad i = 1, ..., s,$$

$$y_1 = y_0 + h_1 \sum_{i=1}^{s} b_i K_1^i.$$

Integration über erstes Makrointervall

$$K_2^i = L\left(y_1 + h_2 \sum_{j=1}^{s} a_{ij} K_2^j\right) + M\left(y_0 + h_1 \sum_{j=1}^{s} b_j(?) K_1^j\right) + N\left(\sum_{j=1}^{s} b_j'(?) K_1^j\right) \quad \text{für} \quad i = 1, ..., s,$$

$$y_2 = y_1 + h_2 \sum_{i=1}^{s} b_i K_2^i.$$

Konvergenzordnung des Verfahrens für die PGL

KonvO via `MatLab` für $L = -100$, $M = -99$, $N = 0$, $r = 0.5$:

$-\log_2(h)$	12.0000	13.0000	14.0000	15.0000
$-\log_2\left(\left\|\text{Fehler}\left(h = 2^{i-1}\right)\right\|\right)$	16.1273	19.0706	22.0416	25.0270
Δ		2.9434	2.9710	2.9853

KonvO via `MatLab` für $L = -100$, $M = -99$, $N = 0.65$, $r = 0.5$:

$-\log_2(h)$	12.0000	13.0000	14.0000	15.0000
$-\log_2\left(\left\|\text{Fehler}\left(h = 2^{i-1}\right)\right\|\right)$	10.9058	13.5120	16.1421	18.7792
Δ		2.6062	2.6301	2.6371

Konvergenz in einem Schritt (Konsistenz)

Wird nur der Fehler im ersten Integrationsschritt bei $t = h$ für verschiedene Schrittweiten h betrachtet, so ergibt sich via `MatLab`:

$-\log_2(h)$	12.0000	13.0000	14.0000	15.0000
$-\log_2\left(\left\|\text{Fehler}\left(h = 2^{i-1}\right)\right\|\right)$	17.4360	20.3352	23.2838	26.2578
Δ		2.8991	2.9486	2.9740

Besseres Verfahren „BMT“ liefert

$-\log_2(h)$	12.0000	13.0000	14.0000	15.0000
$-\log_2\left(\left\|\text{Fehler}\left(h = 2^{i-1}\right)\right\|\right)$	20.9477	24.8342	28.7759	32.7464
Δ		3.8865	3.9417	3.9705

Konvergenzordnung des Verfahrens für die PGL

Wird nur der maximale Fehler im ersten Integrationsintervall $[0,\ h]$ für verschiedene Schrittweiten h betrachtet, so ergibt sich via `MatLab`:

$-\log_2(h)$	12.0000	13.0000	14.0000	15.0000
$-\log_2\left(\left\|\text{Fehler}\left(h=2^{i-1}\right)\right\|\right)$	16.6777	19.6146	22.5820	25.5654
Δ		2.9369	2.9674	2.9834

Problem: im ersten Makrointervall wird auf diese „schlechten“ Werte zurückgegriffen.
Eine mögliche Lösung: mehrere Initialisierungsschritte
Problem: teuer - warum?
Andere Lösung: besseres Verfahren für ersten Integrationsschritt verwenden.

Vergleich zweier Verfahren

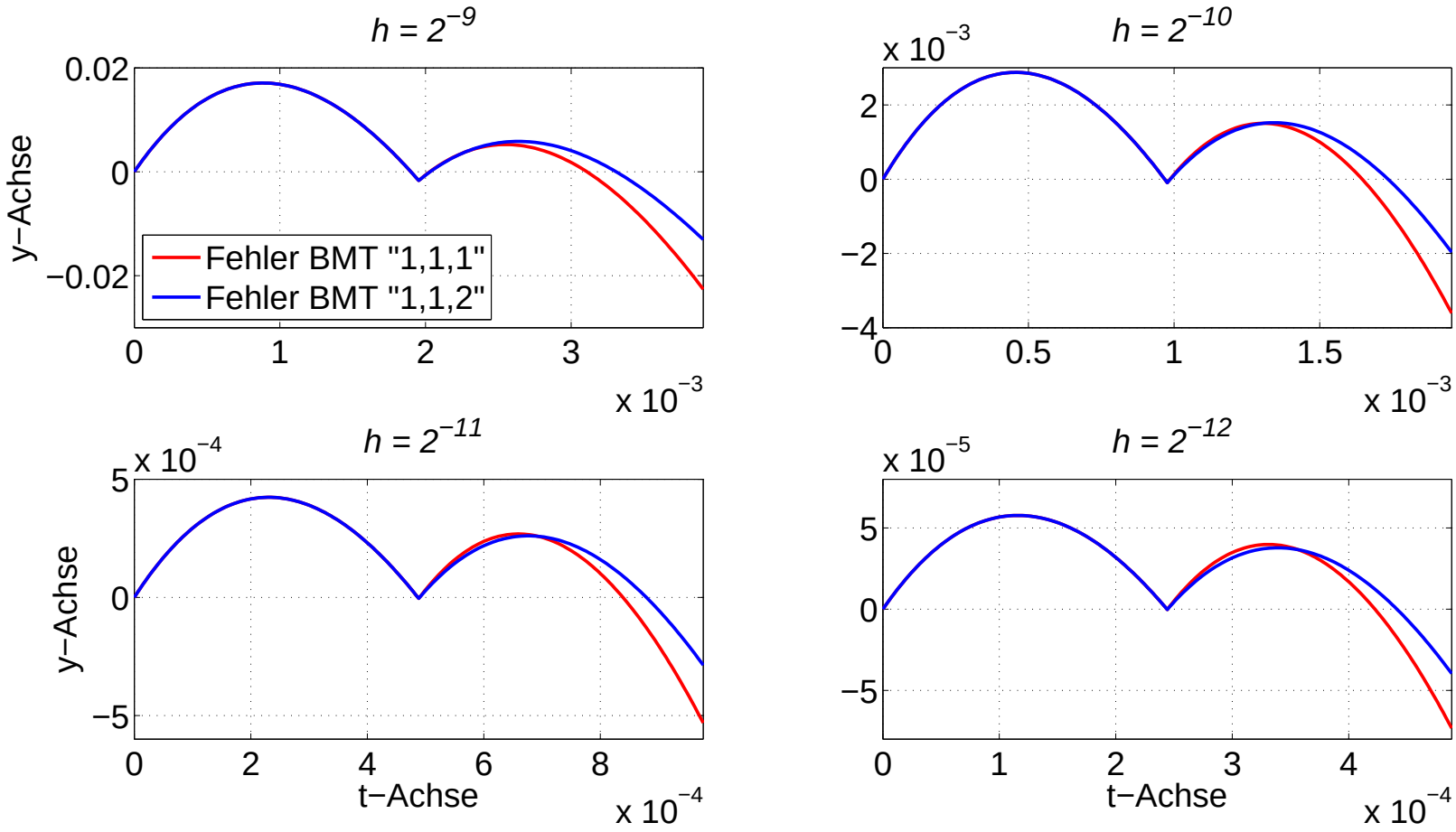

Abb.. *Fehler bei Verwendung von* $\eta'_{LobIIIA}$ *bzw.* η'_{Ehle}

Mehrere Initialisierungsschritte

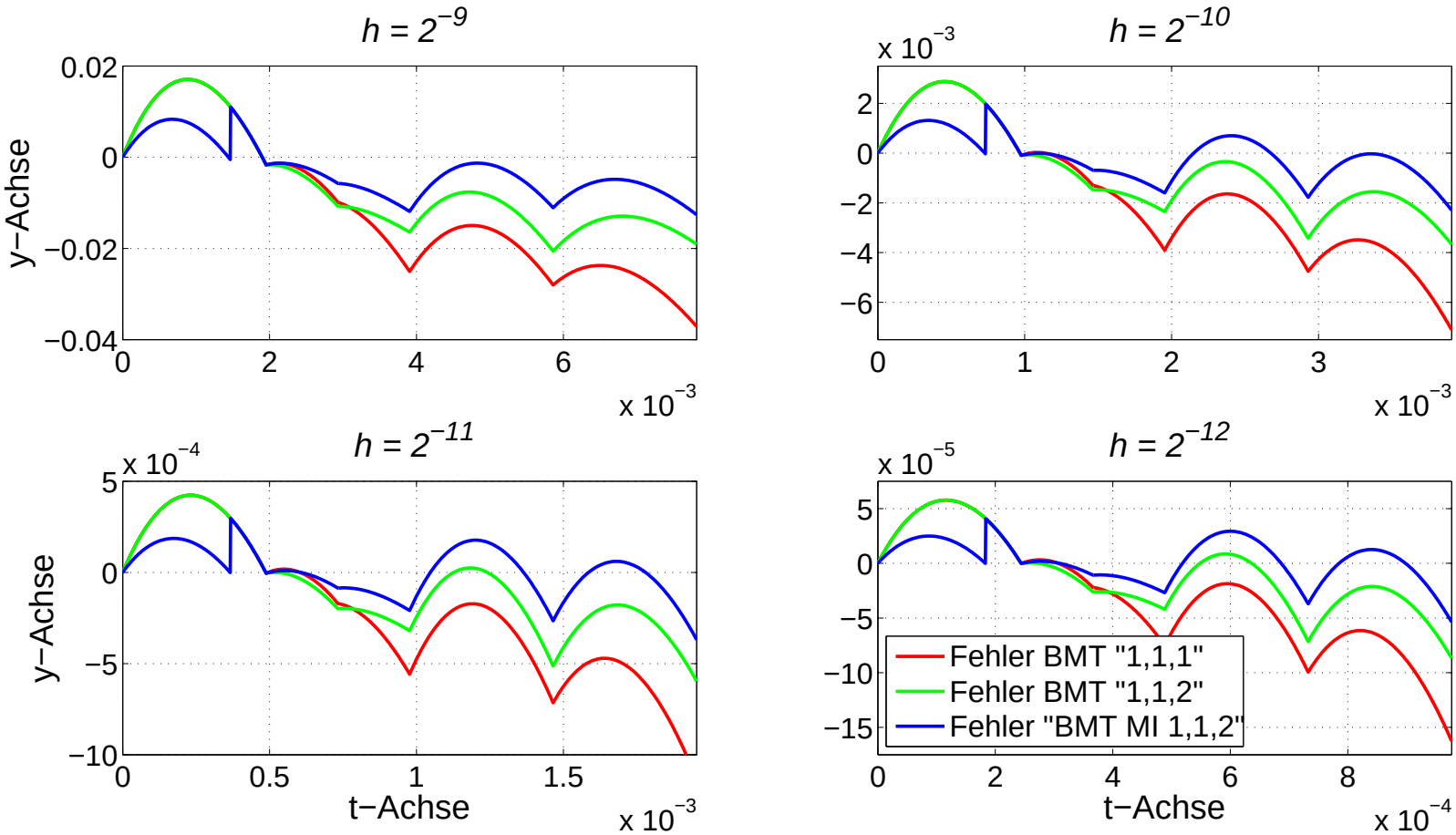

Abb.. *Fehler der „1,1,1“-, „1,1,2“- und „MI 1,1,2“-Verfahren*