

Hochschule München Fakultät 04 Redaktion: Kahl / Paster / Rauh Stand SS 2010

Praktikum Numerik Versuch 4: "Differenzialgleichungen"

Bearbeiter: Dennis Kunz

Datum: 20.06.2012

Testat:

Abgrenzung des Themas: Wie in der Vorlesung beschränken wir uns auf Anfangswertaufgaben gewöhnlicher DGLen. Wir wählen einfache DGLen, für die wir mit exakten Lösungen vergleichen können. Außerdem arbeiten wir mit den einfacheren Lösungsverfahren, weil auch bei diesen schon die wesentlichen Aspekte der numerischen Lösung gezeigt werden können.

1 Genauigkeit bei verschiedenen Verfahren und Schrittweiten

Von Ihrem Dozenten erhalten Sie eine zu integrierende DGL $y' = \text{dgl}(t,y)$, die zugehörige Anfangsbedingung und die Formel für die exakte Lösung $f_{\text{soll}}(t)$.

$\text{dgl}(t,y) := e^{-t^2+2} - t \cdot y$ $t_a := 0$ $y_a := 0$ Anfangswerte **wird vom Dozenten vorgegeben (1)**

$t_e := 2$ Ende des Integrationsbereiches $f_{\text{soll}}(t) := t \cdot e^{-t^2+2}$ exakte Lösung

$n_g := 20$ $n_f := 2 \cdot n_g$ Schrittzahl grob bzw. fein $h_f := \frac{t_e - t_a}{n_f}$ $h_g := 2 \cdot h_f$ Schrittweite fein bzw. grob

$i := 0..n_f$ $t_{f_i} := i \cdot h_f + t_a$ $y_{sf_i} := f_{\text{soll}}(t_{f_i})$ exakte Funktionswerte im Abszissengitter fein bzw. grob

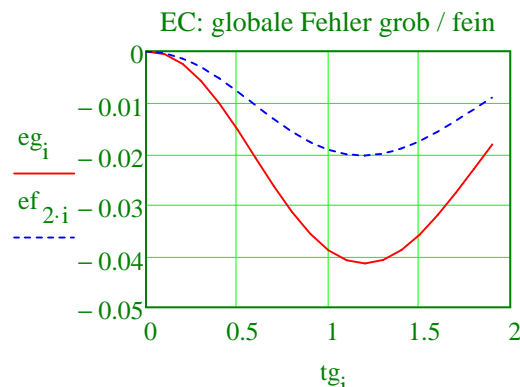
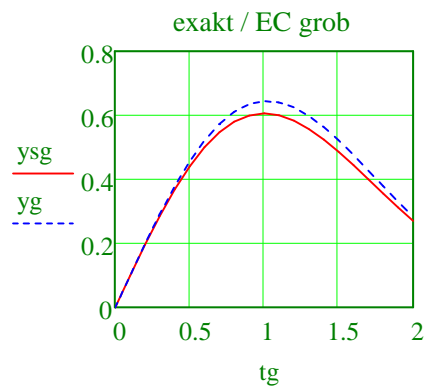
$i := 0..n_g$ $t_{g_i} := t_a + i \cdot h_g$ $y_{sg_i} := f_{\text{soll}}(t_{g_i})$

1.1 Euler-Cauchy-Verfahren (EC)

Selbst dieses einfachste Verfahren zeigt uns schon viel über das Verhalten der numerischen Lösung.

EC-Schritte fein: $y_{f_0} := y_a$ $i := 0..n_f - 1$ $y_{f_{i+1}} := y_{f_i} + h_f \cdot \text{dgl}(t_{f_i}, y_{f_i})$ $ef := y_{sf} - y_f$ Fehler fein

EC-Schritte grob $y_{g_0} := y_a$ $i := 0..n_g - 1$ $y_{g_{i+1}} := y_{g_i} + h_g \cdot \text{dgl}(t_{g_i}, y_{g_i})$ $eg := y_{sg} - y_g$ Fehler grob



Wie groß ist die lokale Schrittfehlerordnung und wie groß die globale Fehlerordnung des EC-Verfahrens? Um welchen Faktor vergrößert sich der globale Fehler ungefähr bei Verdopplung der Schrittweite? Wird das durch unsere Rechnung bestätigt?

Lokal $p = 2$, global $p = 1$. Verdopplung der Schrittweite -> Verdopplung des Fehlers

1.2 Heun-Verfahren

Mit jetzt zwei Aufrufen der DGL pro Schritt können wir eine deutliche Steigerung der Genauigkeit erzielen. Aus Ihrer Vorbereitung: Wie groß ist die lokale und globale Fehlerordnung des Heun-Verfahrens?

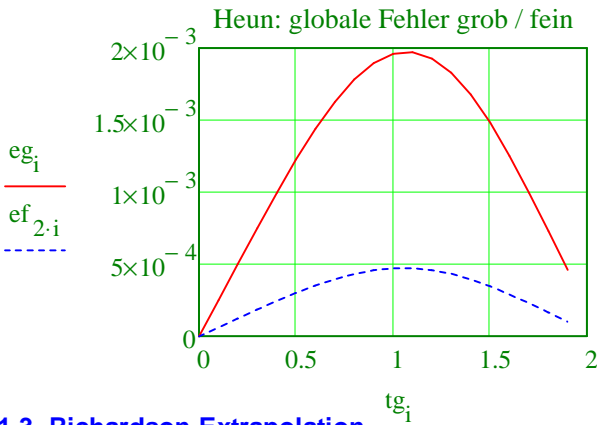
Lokal $p = 3$, global $p = 2$

Ergänzen Sie neben stehende Prozedur zur Berechnung eines Heun-Schritts der Weite h vom Punkt (t, y) aus:

$$\text{heun}(\text{dgl}, t, y, \text{hr}) := \begin{cases} \text{abl1} \leftarrow \text{dgl}(t, y) \\ \text{abl2} \leftarrow \text{dgl}(t + \text{hr}, y + \text{hr} \cdot \text{abl1}) \\ \text{yneu} \leftarrow y + \text{hr} \cdot (\text{abl1} + \text{abl2}) \div 2 \end{cases}$$

Heun-Schritte fein: $yf_0 := y_a \quad i := 0..nf - 1 \quad yf_{i+1} := \text{heun}(\text{dgl}, \text{tf}_i, yf_i, \text{hf}) \quad \text{ef}_{\text{ww}} := yf - yf \quad \text{Fehler fein}$

Heun-Schritte grob: $yg_0 := y_a \quad i := 0..ng - 1 \quad yg_{i+1} := \text{heun}(\text{dgl}, \text{tg}_i, yg_i, \text{hg}) \quad \text{eg}_{\text{ww}} := ysg - yg \quad \text{Fehler grob}$



Um welchen Faktor vergrößert sich der globale Fehler ungefähr bei Verdopplung der Schrittweite? Wird das durch unsere Rechnung bestätigt?

Verdopplung der Schrittweite -> Vervierfachung des Fehlers

1.3 Richardson-Extrapolation

Die Richardson-Extrapolation ist für die Praxis wichtig, denn sie liefert zwei sehr nützliche Informationen.

- 1) ein verbessertes Ergebnis (y_{rich}) (mit Fehler e_{rich})
- 2) eine Fehlerschätzung für das Heun-Verfahren (e_{heun})

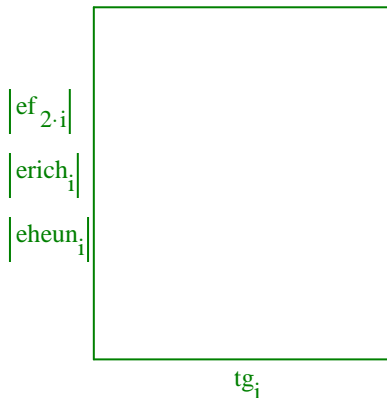
Warum wird die Richardson-Extrapolation beim EC-Verfahren nicht angewendet?

Weil Fehlerordnung $p > 1$ sein muss. EC hat die Fehlerordnung $p = 1$

$$i := 0..ng \quad y_{\text{rich}_i} := (4 \cdot yf_{2,i} - yf_i) \div 3 \quad e_{\text{rich}_i} := ysg_i - y_{\text{rich}_i} \quad e_{\text{heun}_i} := (yf_{2,i} - yf_i) \div 3$$

$i := 1..ng$ Null ausblenden wegen log-Skala

Wirkung Rich.-Etrapol.



$$\text{hg} = 0.1$$

$$\text{hf} = 0.05$$

Um welchen Faktor wird das Ergebnis hier verbessert?

$$ef/erich > 3$$

1.4 Schrittweitensteuerung

Wir geben Ihnen hier eine Prozedur (adapt) zur adaptiven Lösung einer DGL und demonstrieren ihre Anwendung an einer DGL, die Sie von Ihrem Dozenten erhalten.

$\text{hcartare} := 0.1$ Anfangs-Schrittweite

wird vom Dozenten vorgegeben (2)

$\text{errtol} := 10^{-3}$ Toleranz des lokalen Fehlers je Schritt

$\text{ns} := 43$ Zahl der (i.A.nicht äquidistanten) Schritte; durch Probieren gefunden

$k := 1..ns$

$$Z^{(0)} := \begin{pmatrix} \text{ta} \\ \text{hcartare} \\ 0 \\ y_a \end{pmatrix}$$

```

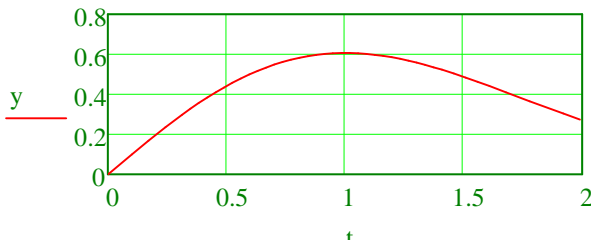
adapt(dgl, errtol, z) :=
  hr ← 2·z1
  dy ← 2·errtol
  while |dy| > errtol
    hr ← 0.5·hr
    a1 ← dgl(z0, z3)
    ym ← z3 + 0.5·hr·a1
    am ← dgl(z0 + 0.5·hr, ym)
    yf ← ym + 0.5·hr·am
    dy ← yf - (z3 + hr·a1)
  yrich ← yf + dy
  hneu ← hr if |dy| > 0.2·errtol
  hneu ← √2·hr otherwise
  return (z0 + hr hneu dy yrich)T
    
```

$$Z^{(k)} := \text{adapt}(dgl, \text{errtol}, Z^{(k-1)})$$

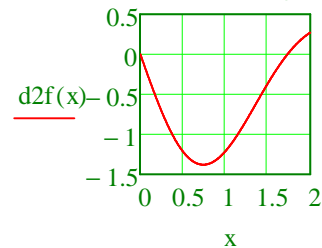
$\underline{\text{watt}} := Z^T$
 $t := \text{watt}^{(0)} \quad t_i$
 $y := \text{watt}^{(3)} \quad \text{numerische Lösung}$
 $\underline{\text{hr}} := \text{watt}^{(1)} \quad \text{Schrittweite}$
 $\text{elok} := \text{watt}^{(2)} \quad \text{Schätzung lok. Fehler}$
 $y_{\text{exakt}k} := f_{\text{so}}(t_k) \quad \text{exakte Lösung}$
 $\text{eglob} := y_{\text{exakt}} - y \quad \text{globaler Fehler}$

$$d2f(x) := \frac{d^2}{dx^2} f_{\text{so}}(x) \quad \begin{matrix} 2. \text{ Abl.} \\ \text{exakte} \\ \text{Lösung} \end{matrix}$$

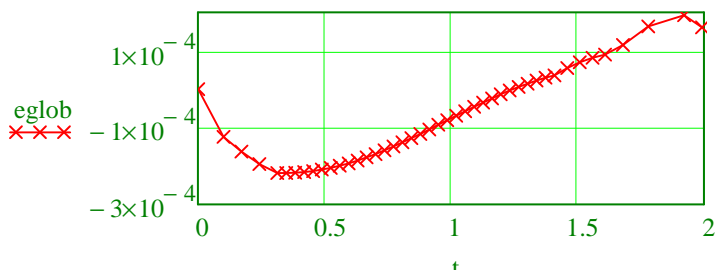
Funktionsverlauf numerisch



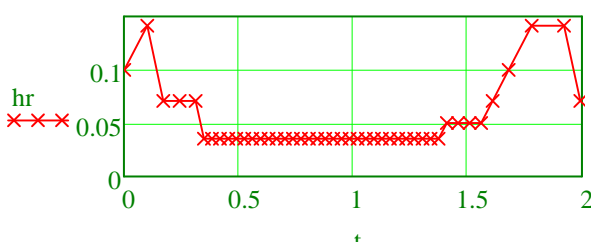
2. Ableitung



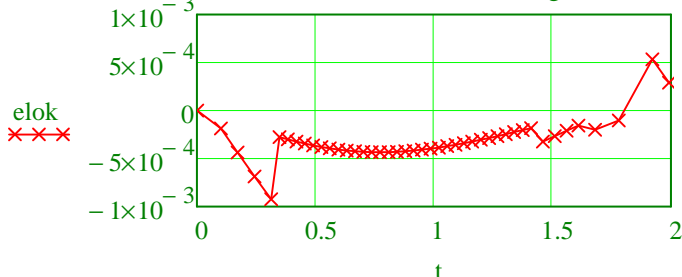
tats. globaler Fehler



Schrittweite



lokale Fehlerschätzung



Diskutieren Sie den Verlauf der erhaltenen Kurven.

Zunächst große Schrittweiten, dann kleinere um TOL zu erfüllen. Zuletzt wieder größere Weiten.

Im Bereich um $x = 1$ wechselt das Vorzeichen der Steigung. Es werden geringere Schrittweiten benötigt um den Tangentenverlauf zu folgen.

An den Randbereichen ist die Steigung annähernd konstant. Größere Schrittweiten sind möglich.

Nach dem ersten Schritt sind große Weiten möglich. Um im TOL-Bereich zu bleiben, wird die Schrittweite immer weiter erniedrigt.

Verkleinerung der SW mit $\cdot 1/2$
Vergrößerung der SW mit $\cdot \text{sqrt}(2)$

2 Systeme von Differenzialgleichungen

Verfahren zur numerischen Integration von DGL-Systemen lassen sich aus den bekannten Verfahren für Einzel-DGLen ableiten, indem wir von der skalaren Darstellung von DGL und Lösungsfunktion auf eine vektorielle Darstellung übergehen. In jedem Lösungsschritt sind dann alle Komponenten eines Vektors quasi parallel zu bearbeiten. Mathcad macht die Erweiterung von Skalaren auf Vektoren i.A. automatisch: Der Ergebnistyp eines Terms wechselt ggf. von Skalar auf Vektor, wenn als Argumente Vektoren anstatt Skalare eingesetzt werden. Von Ihrem Dozenten erhalten Sie ein einfaches DGL-System aus zwei DGLen 1. Ordnung mit einer bekannten exakten Lösung. Die beiden von t abhängigen Variablen werden als Komponenten y_0 und y_1 eines Vektors y aufgefasst.

$$\text{Dgl}(t, y) := \begin{pmatrix} -2 \cdot \pi \cdot y_1 \\ 2 \cdot \pi \cdot y_0 \end{pmatrix} \quad \underline{\underline{y_a}} := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{Vektor der Anfangswerte für } \underline{\underline{y}} \text{ wird vom Dozenten vorgegeben (3)}$$

$$\underline{\underline{f_soll}}(t) := \begin{pmatrix} \cos(2 \cdot \pi \cdot t) \\ \sin(2 \cdot \pi \cdot t) \end{pmatrix} \quad \text{exakte Lösung vektoriell} \quad n := 1080 \quad \underline{\underline{t_a}} := 0 \quad \underline{\underline{t_e}} := 3$$

$$Z := 0 \quad \text{rücksetzen} \quad \text{hr} := \frac{t_e - t_a}{n} \quad i := 0..n \quad t_i := t_a + i \cdot \text{hr}$$

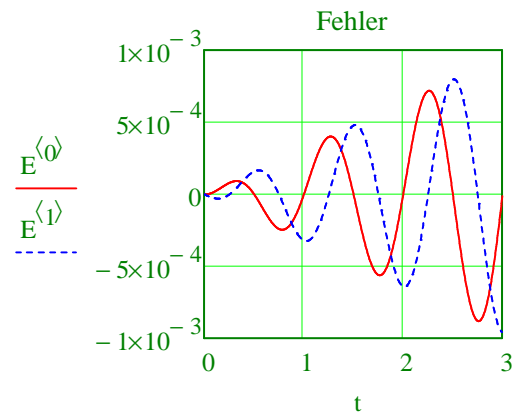
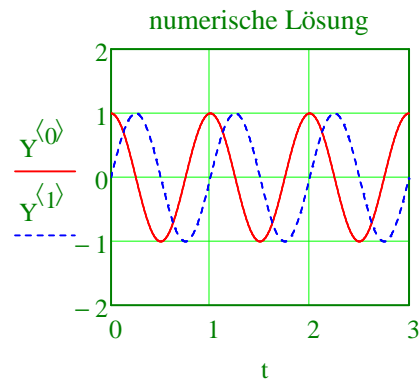
Mit einer von früher bekannten Programmiertechnik speichern wir jeweils den Ergebnisvektor eines Schritts als neue Spalte einer Matrix Z. Im gleichen Format speichern wir die Sollwerte der Lösung.

$$\underline{\underline{Z}}^{(0)} := \underline{\underline{y_a}} \quad i := 0..n-1 \quad \underline{\underline{Z}}^{(i+1)} := \text{heun}(\text{Dgl}, t_i, \underline{\underline{Z}}^{(i)}, \text{hr}) \quad \text{heun wird mit Vektor aufgerufen und liefert Vektor}$$

$$i := 0..n \quad \underline{\underline{Zsoll}}^{(i)} := \underline{\underline{f_soll}}(t_i)$$

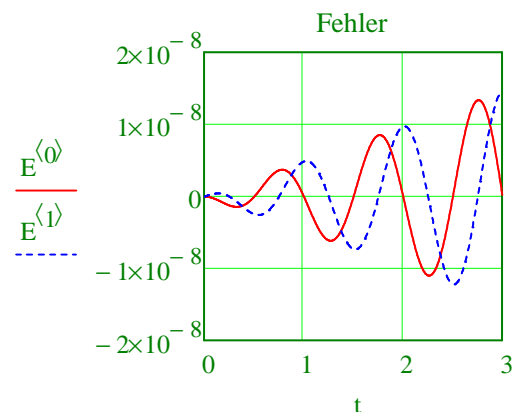
Nach dem Transponieren stehen die Abtastwerte der beiden zeitabhängigen Variablen sowie Ihre Fehler als Spaltenvektoren in den entsprechenden Matrizen zur Verfügung:

$$Y := Z^T \quad Y_{\text{soll}} := Z_{\text{soll}}^T \quad E := Y_{\text{soll}} - Y$$



Zum Vergleich berechnen wir die Lösung mit dem Runge-Kutta-Verfahren mit der Mathcad-Funktion rkfest. Wie groß ist der Rechenaufwand im Vergleich zum Heun-Verfahren? ungefähr doppelt so hoch

$$Z := \text{rkfixed}(\underline{\underline{Z}}^{(0)}, t_a, t_e, n, \text{Dgl}, \underline{\underline{f_soll}}^{(0)}) \quad \underline{\underline{E}}^{(0)} := Y_{\text{soll}}^{(0)} - Z^{(1)} \quad \underline{\underline{E}}^{(1)} := Y_{\text{soll}}^{(1)} - Z^{(2)}$$



3 Numerische Stabilität

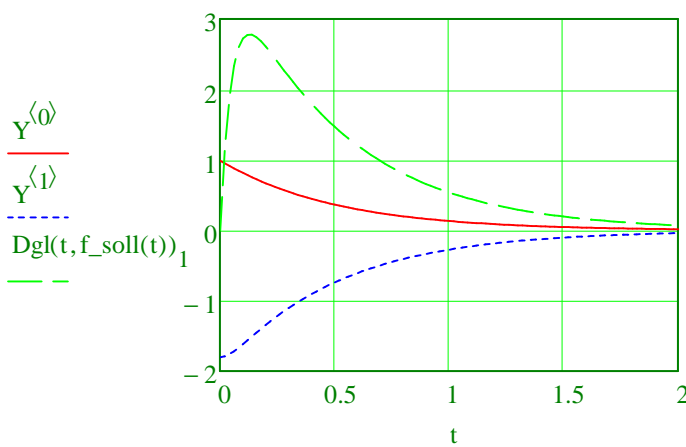
Wir demonstrieren nun die numerische Lösung eines relativ "steifen" DGL-Systems, dessen Lösung sich aus langsam veränderlichen und aus schnell veränderlichen Komponenten zusammensetzt. Das System und seine exakte Lösung lauten:

$$\underset{\text{Dgl}}{\text{Dgl}}(t, y) := \begin{pmatrix} y_1 \\ -40 \cdot y_0 - 22 \cdot y_1 \end{pmatrix} \quad y_a := \begin{pmatrix} 0.99 \\ -1.8 \end{pmatrix} \quad \text{Anfangsbedingung für } t=0$$

$$\underset{\text{f_soll}}{\text{f_soll}}(t) := \begin{pmatrix} \exp(-2 \cdot t) - 0.01 \cdot \exp(-20 \cdot t) \\ -2 \cdot \exp(-2 \cdot t) + 0.2 \cdot \exp(-20 \cdot t) \end{pmatrix}$$

Sollverlauf grafisch dargestellt:

$$t := 0 \quad \underset{\text{Z}}{\text{Z}} := 0 \quad i := 0..100 \quad \underset{\text{tj}}{t_j} := 0.02 \cdot i \quad \underset{\text{Z^{i}}}{\text{Z}} := \text{f_soll}(t_j) \quad Y := Z^T$$



In den beiden Lösungsfunktionen ist der "schnelle" Summand ($\exp(-20 \cdot t)$) nach kurzer Zeit bereits praktisch abgeklungen, so dass sie danach nur von der "langsamen" e-Funktion ($\exp(-2 \cdot t)$) bestimmt werden. Trotzdem darf er bei der numerischen Lösung zu keiner Zeit außer Acht gelassen werden, wie wir gleich sehen werden. Wir lösen jetzt das DGL-System numerisch nach dem Heun-Verfahren mit verschiedenen Schrittweiten:

$$\underset{\text{te}}{t_e} := 1.6 \quad \text{Endwert}$$

$$h_{\text{fein}} := 0.04 \quad h_{\text{mittel}} := 0.08 \quad h_{\text{grob}} := 0.16$$

wird vom Dozenten vorgegeben (4)

Integration mit h_{fein} : $\underset{\text{hr}}{h_r} := h_{\text{fein}} \quad \underset{\text{n}}{n} := \frac{t_e}{h_r} \quad n = 40 \quad t := 0 \quad Z := 0 \quad Z_{\text{soll}} := 0 \quad \text{rücksetzen}$

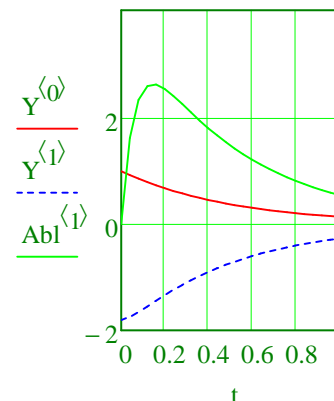
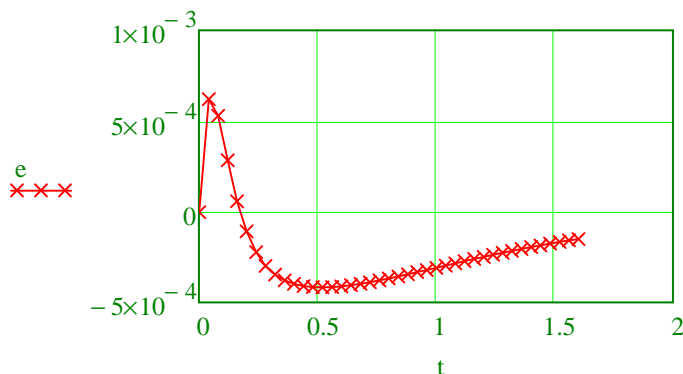
$$\underset{\text{Z^{0}}}{\text{Z}} := y_a \quad i := 0..n \quad \underset{\text{tj}}{t_j} := i \cdot h_r \quad \underset{\text{Z_{soll}^{i}}}{\text{Z}_{\text{soll}}} := \text{f_soll}(t_j) \quad i := 0..n-1 \quad \text{Z}^{i+1} := \text{heun}(\text{Dgl}, t_j, \text{Z}^{i}, h_r)$$

Nach Angabe Ihres Dozenten untersuchen Sie nun den Zeitverlauf eines der beiden globalen Fehler.

$$Y := Z^T \quad Y_{\text{soll}} := Z_{\text{soll}}^T \quad E := Y_{\text{soll}} - Y \quad \underset{\text{e}}{e} := E^{(0)}$$

wird vom Dozenten vorgegeben (5)

$$\text{Abl}_{i,1} := \text{Dgl}\left[0, \left(Y^T\right)^{(i)}\right]_1 \quad h_r = 0.04$$

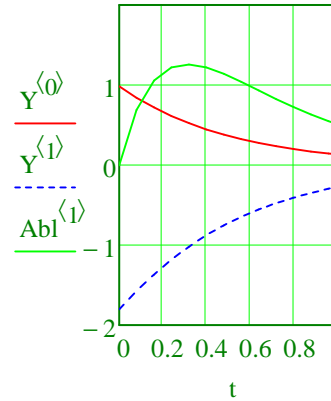
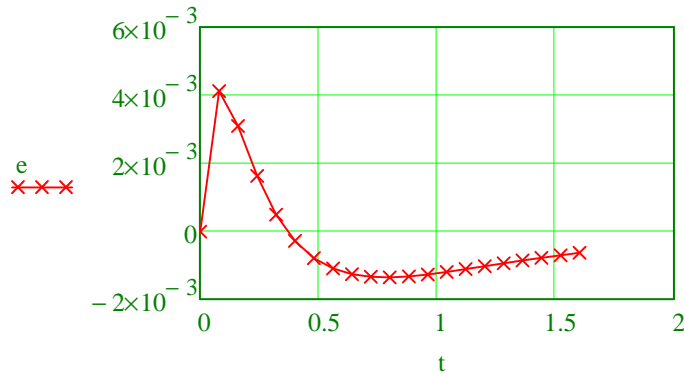


Integration mit h_{mittel} : $h_{\text{mittel}} := h_{\text{mittel}} \quad n := \frac{t_e}{h_{\text{mittel}}} \quad n = 20 \quad t := 0 \quad Z := 0 \quad Z_{\text{soll}} := 0$ **rücksetzen**

$Z_{\text{mittel}}^{(0)} := y_a \quad i := 0..n \quad t_{\text{mittel}} := i \cdot h_{\text{mittel}} \quad Z_{\text{soll}}^{(i)} := f_{\text{soll}}(t_i) \quad i := 0..n-1 \quad Z^{(i+1)} := \text{heun}(Dgl, t_i, Z^{(i)}, h_{\text{mittel}})$

$Y := Z^T \quad Y_{\text{soll}} := Z_{\text{soll}}^T \quad E_{\text{mittel}} := Y_{\text{soll}} - Y \quad e_{\text{mittel}} := E^{(0)}$ **wie oben** $h_{\text{mittel}} = 0.08$

$$\text{Abl}_{i,1} := \text{Dgl}[0, (Y^T)^{(i)}]_1$$

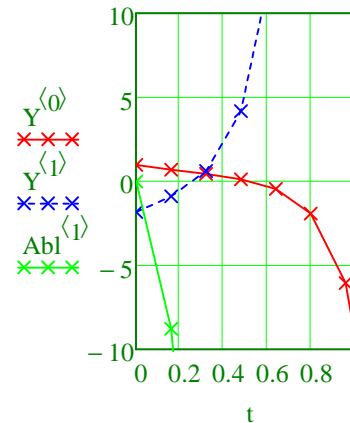
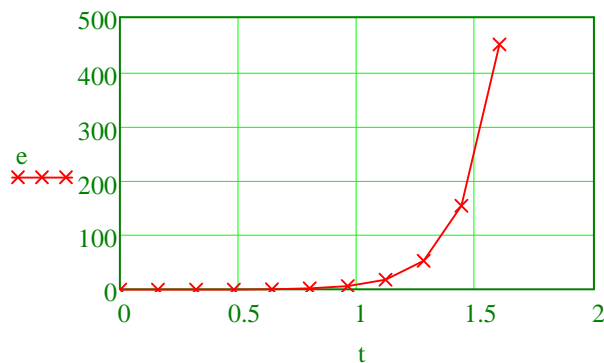


Integration mit h_{grob} : $h_{\text{grob}} := h_{\text{grob}} \quad n := \frac{t_e}{h_{\text{grob}}} \quad n = 10 \quad t := 0 \quad Z := 0 \quad Z_{\text{soll}} := 0$ **rücksetzen**

$Z_{\text{grob}}^{(0)} := y_a \quad i := 0..n \quad t_{\text{grob}} := i \cdot h_{\text{grob}} \quad Z_{\text{soll}}^{(i)} := f_{\text{soll}}(t_i) \quad i := 0..n-1 \quad Z^{(i+1)} := \text{heun}(Dgl, t_i, Z^{(i)}, h_{\text{grob}})$

$Y := Z^T \quad Y_{\text{soll}} := Z_{\text{soll}}^T \quad E_{\text{grob}} := Y_{\text{soll}} - Y \quad e_{\text{grob}} := E^{(0)}$ **wie oben** $h_{\text{grob}} = 0.16$

$$\text{Abl}_{i,1} := \text{Dgl}[0, (Y^T)^{(i)}]_1$$



Diskutieren Sie ausführlich die drei berechneten Fehlerverläufe.

h_{fein} : Die Funktionen werden gut angenähert

h_{mittel} : Die Funktionen werden nicht mehr so gut angenähert.

Fehler steigt zunächst an und geht dann gegen Null. Da die Konvergenzordnung 2 ist, ist der Fehler 4-mal so groß

h_{grob} : Schrittweite liegt außerhalb des Stabilitätsbereiches. Der "Integrationsfehler" des Verfahrens bei nur einem Verbesserungsschritt ist zu groß.