Grundlagen

Fehlerfortpflanzungsgesetz $|\Delta y| \le \sum_{i=1}^{n} (|\frac{\partial f}{\partial x_i}(\tilde{x})| * |\Delta x_i|$

Fehlerarten: Verfahrensfehler (Falsches numerisches Verfahren)
Rechnungsfehler (Rundungsfehler)
Eingangsfehler (Fehlerfortpflanzung)

Instabiler Algorithmus:

$y_{n+1} = a \cdot y_n + b \cdot y_{n-1}$ $\left|\frac{a}{2} \pm \sqrt{\frac{a^2}{4} + b}\right| < 1$ rekursiver Algorithmus und sein Stabilitätskriterium

$\Delta Y_n = -\varepsilon; \Delta y_{n-1} = \varepsilon$
$y_{n+1} = a(y_n - \varepsilon) + b(y_{n-1} + \varepsilon)$

Lineare Gleichungssysteme

CRAMER $x_i = \frac{\det A_i}{\det A}$

Frobenius-Matrix L für **LR Zerlegung** $l_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$

Nachiteration $A \cdot x = b$ Residium $r = b - A \cdot \tilde{x} \Rightarrow \Delta x = A^{-1} \cdot r$ $\tilde{\tilde{x}} = \tilde{x} + \Delta x$ besserer Wert – reicht oft schon 1Schritt

Iterationsverfahren(Jacobi & Gauss-Seidel) $A \cdot x = b \Longrightarrow \varphi(x) = C \cdot x + d$ $c_{ik} = -\frac{a_{ik}}{a_{ii}}, i \ne k, d_i = \frac{b_i}{a_{ii}}$ Betragsmäßig größten Werte müssen auf Hauptdiagonale liegen.

Gauss-Seidel: $x_i = \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{k=1 \\ k \ne i}}^{n} a_{ik} x_i\right)$ konvergiert wenn A symmetrisch und positiv $\Rightarrow$ $A \cdot x = b \Rightarrow A^T A \cdot x = A^T b$

Konvergenzkriterium: Zeilensummenkr., Spaltensummenkr., Kriterium für Symmetrische Matrizen

Nichtlineare Gleichungen

Wenn φ (x) in [a,b] für x∈ [a,b] und Zahl M∈]0,1[für $|\varphi'(x)| \le M$ dann konvergiert φ (x)=x für jeden Startwert in [a,b] genau gegen eine Lösung ξ

Fehlerabschätzung: $|x_n - \xi| \le \frac{M^n}{1-M} |x_1 - x_0|$ (a-priori) oder $|x_n - \xi| \le \frac{M}{1-M} |x_n - x_{n-1}|$ (a-posteriori)

Newtonverfahren: $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ Startwert x_o beliebig in der Nähe einer Nullstelle (lokal konvergent) Konvergenzbedingung: $\left|\frac{f''(x) f(x)}{(f'(x))^2}\right| \le M \le 1$

Möglichkeiten aus einem Nullstellenproblem ein Fixpunktproblem zu machen: $f(x) = 0$ Funktion wird nach einem beliebigem x der Funktion f(x) aufgelöst

Regula falsi: $x_1 = a - \frac{f(a)(b-a)}{f(b) - f(a)}$ $f(x_1) \cdot f(a) > 0 \Longrightarrow a = x_1$ $f(x_1) \cdot f(a) < 0 \Longrightarrow b = x_1$ global konvergent

Horner-Schema(erzeugen reduzierter Polynome): $f(x) = a_3 \cdot x^3 + a_2 \cdot x^2 + a_1 \cdot x + a_0 \quad (a_3 \ne 0)$ erste NS raten (z.B.: x_0=3)

Koeffizienten für neues Polynom

	a_3	a_2	a_1	a_0
x_0		$x_0 a_3$	$(a_2 + x_0 a_3) x_0$	$(a_1 + a_2 x_0 + a_3 x_0^2) x_0$
	a_3	$a_2 + a_3 x_0$	$a_1 + a_2 x_0 + a_3 x_0^2$	$a_0 + a_1 x_0 + a_2 x_0^2 + a_3 x_0^3$

b_3

b_1 b_0

$f(x_0)$(=0 bei Nullstellen)

Newton-Raphson: $\vec{F}(\vec{x}_k)$ $\rightarrow$ Jakobi-Matrix $J(\vec{x}) = \left(\frac{\partial f_i(\vec{x})}{\partial \vec{x}_k}\right)_{\substack{i=1..n \\ k=1..n}}$ $\rightarrow$ $J(\vec{x}_k)\Delta\vec{x} = -F(x_k)$ $\rightarrow$ $\vec{x}_{k+1} = \vec{x}_k + \Delta\vec{x}$ k=0,1..

$f_1(x,y) = \ldots = 0 \quad , f_2(x,y) = \ldots = 0$

Startwert : $z_0 = (x_0, y_0)$ $F(x_0, y_0) = \begin{pmatrix} f_1(x_0, y_0) \\ f_2(x_0, y_0) \end{pmatrix}$

JakobiMatrix : $J(x_0, y_0) = \begin{pmatrix} \frac{\partial f_1}{\partial x}|x_0 & \frac{\partial f_1}{\partial y}|y_0 \\ \frac{\partial f_2}{\partial x}|x_0 & \frac{\partial f_2}{\partial y}|y_0 \end{pmatrix} \Rightarrow J(x_0, y_0) = (\frac{\Delta x}{\Delta y}) = -\vec{F}(x_0, y_0) \Rightarrow x_1 = x_0 + \Delta x \quad , y_1 = y_0 + \Delta x$

Wichtig : x_1 , y_1 als neue Werte x_0, y_0 nehmen als neue Startwerte

Interpolation

Unmittelbarer Polynomansatz:
Stützpunkte durch ein Polynom:

…durch ein Polynom: geg. n+1 Stützpunkte (x_i,y_i) i=0..n –> $p(x)=\beta_0+\beta_1 x+\beta_2 x^2+\ldots+\beta_n x^n$ ⇒ $\begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}$

Vandermonde Matrix ist im allgemeinen schlecht konditioniert, andere Verfahren besser

Newton Interpolation $\boxed{\Rightarrow P_n(x) = c_0 + c_1 \cdot (x - x_0) + c_2 \cdot (x - x_0)(x - x_1) + \ldots + c_n \cdot (x - x_0)(x - x_1)\ldots(x - x_{n-1})}$

Differenzschema. Für n=3

$$\begin{matrix} y_0 & & & \\ & d_{1,0} = y_1 - y_0 & & \\ y_1 & & d_{2,0} = d_{1,1} - d_{1,0} & \\ & d_{1,1} = y_2 - y_1 & & d_{3,0} = d_{2,1} - d_{1,1} \\ y_2 & & d_{2,1} = d_{1,2} - d_{1,1} & \\ & d_{1,2} = y_3 - y_2 & & \\ y_3 & & & \end{matrix}$$

c_i entstehen aus

$$c_0 = y_0 \quad c_1 = d_{1,0} \quad c_2 = \tfrac{1}{2} d_{2,0} \quad c_3 = \tfrac{1}{6} d_{3,0}$$

Vorteil vom Newton gegenüber anderer Verfahren (z.B. **Lagrange Interpolation):** die Anzahl der Stürzpunkte kann beliebig vergrößert werden, ohne dass die Koeffizienten neu berechnet werden müssen. Nachteil: bei höherem Grad (>6) neigen die Polynome zu starkem Überschwingen.

Lagrange: $\boxed{P_n(x) = L_0(x) y_0 + L_1(x) y_1 + \ldots + L_n(x) y_n}$ $\boxed{L_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i}}$ Fehler $\frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)(x - x_1)\ldots(x - x_n)$

Nachteil: Soll ein weiterer Stützpunkt hinzugenommen werden, um den Grad des Näherungspolynoms um 1 zu erhöhen, so müssen sämtliche Koeffizientenfunktionen neu berechnet werden.

...durch ein trigonometrisches Polynom. Geeignet für periodische oder periodisch fortgesetzte Funktionen. Koeffizienten können unabhängig voneinander berechnet werden, da die Basisfunktionen summen-orthogonal sind

Splineinterpolation: Interpolationsfunktion S(x) durch Polynome niedrigeren Grades $S_k(x)$ die auf $[x_k,x_{k+1}]$ definiert sind, stückweise zusammensetzten.
Kubische Interpolation: $S_k(x_k)=a_{k0}+a_{k1}(x-x_k)+a_{k2}(x-x_k)^2+a_{k3}(x-x_k)^3$ für $x\in[x_k,x_{k+1}]$ und k=0..n-.

$\boxed{a_{k0} = y_k \quad a_{k1} = d_k - \frac{h_k(2m_k + m_{k+1})}{h_k} \quad a_{k2} = \frac{m_k}{2} \quad a_{k3} = y_k \frac{m_{k+1} - m_k}{6h_k}}$ $h_k = x_{k+1} - x_k \;/\; d_k = \frac{y_{k+1} - y_k}{h_k}$

Jedes Polynom $S_k(x)$ besitzt 4 unbekannte Koeffizienten ⇒ 4n Unbekannte
$S_k(x_k)=y_k$ für k=0..n (S nimmt Stützwerte an)⇒ n+1 Bedingungen
$S_k(x_{k+1})=S_{k+1}(x_{k+1})$ für k=0..n-2 (S ist stetig) ⇒n-1 Bedingungen
$S'_k(x_{k+1})=S'_{k+1}(x_{k+1})$ für k=0..n-2 (S ist glatt) ⇒ n-1 Bedingungen
Aus Randpunkt-Restriktionen folgt: $S''_k(x_k)=m_k$ & $S''_k(x_o)=m_o$⇒ insgesamt 4n Bedingungen

Approximation (Funktion bekannt aber zu schwer)

Taylorpolynom $\boxed{P_n(x) = \sum_{k=0}^{n} \frac{f^k(f(x_0))}{k!}(x - x_0)^k + \underbrace{\frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)^{n+1}}_{\text{Fehlerterm } R_n(x)}}$ mit $\xi \in \,]x_0, x[$ Abweichung p(x)-f(x) steigt i.a. stark mit dem Abstand $x-x_o$

Tschebyscheffsumme: f(x) wird angenähert durch eine endliche Summe von T.-Polynomen $\boxed{p(x) = c_0 T_o(x) + \ldots + c_n T_n(x)}$

$\boxed{T_{n+1} = 2xT_n - T_{n-1} \;/\; T_0 = 1 \;/\; T_1 = x}$ geeignet für das Integral [-1,1]. Erlaubt eine annähernd „gleichmäßige Approximation“ von f(x)
Approximation durch Interpolation: Werte speichern und Interpolieren
Approximationsfehler:
Grad 0= Treppenfunktion: $e=1/2 \cdot h \cdot f'$
Grad 1= lineare Interpolation: $e=1/8 \cdot h^2 \cdot f''$
Grad 2= quadratische Interpolation: $e=1/16 \cdot h^3 \cdot f'''$
Grad 3= kubische Interpolation: $e=1/24 \cdot h^4 \cdot f^{(4)}$

Ausgleichsrechnung

Funktion durch Messpunkte bekannt, Komponenten k_i gesucht

$\Delta = \sum_{i=1}^{N} (f(x_i) - y_i)^2$ muss minimal sein $\frac{\partial \Delta}{\partial k_i} = 0 \Rightarrow$ aus $\vec{y} = a + bx + c \cdot f(x)\ldots$ folgt: $\boxed{A = \begin{pmatrix} 1 & x_1 & f(x_1) \\ & \ldots & \\ 1 & x_n & f(x_n) \end{pmatrix} \quad \vec{y}\begin{pmatrix} y_1 \\ \ldots \\ y_n \end{pmatrix} \quad \vec{\gamma}\begin{pmatrix} a \\ b \\ c \end{pmatrix}}$

$\boxed{A^T \vec{y} = A^T A \cdot \vec{y} = M \cdot \vec{y}}$ M ist Quadratisch und Symmetrisch $(A^T A)^T = A^T A^{T^T} = A^T A$

Numerisch Integration

n	c_n	β_0	β_1	β_2	β_3	β_4	Restglied
1	$\frac{h}{2}$	1	1				$\frac{-h^3}{12} f''(\xi) \rightarrow$ Sehnen-Trapez
2	$\frac{h}{3}$	1	4	1			$\frac{-h^5}{90} f^{(4)}(\xi) \rightarrow$ Simpson
3	$\frac{3h}{8}$	1	3	3	1		$\frac{-h^5}{80} f^{(4)}(\xi) \rightarrow$ 3/8 Regel
4	$\frac{2h}{45}$	7	32	12	32	7	$\frac{-h^7}{945} f^{(6)}(\xi)$

n	c_n	β_1	β_2	β_3	Restglied
2	2h	1			$\frac{-h^3}{3} f''(\xi) \rightarrow$ Tangenten - Trapez Mittelpunktregel
3	$\frac{3h}{2}$	1	1		$\frac{-h^5}{90} f^{(4)}(\xi)$
4	$\frac{4h}{3}$	2	-1	3	$\frac{-h^5}{80} f^{(4)}(\xi)$

Newton-Cotes $\boxed{Q_n = c_n \sum_{k=0}^{n} \beta_k f(x_k)}$ $\boxed{\int_a^b f(x)dx = \int_a^b p_n(x)dx + R_n}$

Geschlossene Newton-Cotes-Formeln: Integrationsgrenzen a und b gehören zu den Stützstellen der Integrationsformel

$x_k = a + k \cdot h || k = 0..n || h = (b-a)/n || \xi \in [a,b]$

Grad $p_n(x)=n$

Offene Newton-Cotes-Formeln: $Q = c_n \sum_{k=0}^{n} \beta_k f(x_k)$ Integrationsformel kommt ohne a und b als Stützstellen aus

$x_k = a + k \cdot h || k = 1..n-1 || h = (b-a)/n || \xi \in [a,b]$

Grad $p_n(x)=n-2$

Summierte Integrationsformeln:

Sehnen-Trapez-Regel: auf Intervall $[x_k, x_{k+1}]$ anwenden k=0..n

$\boxed{\begin{matrix} h = (b-a)/n \\ x_k = a + k \cdot h \end{matrix}}$ $\boxed{\int_a^b f(x)dx = \frac{h}{2}\left(f(a) + 2\sum_{k=1}^{n-1} f(x_k) + f(b) \right) + R = T_n(h) + h^2 C}$ $\boxed{|R| \le \frac{b-a}{12} h^2 \max_{a \le x \le b} |f''(x)|}$

Simpson-Regel

auf Intervall $[x_k, x_{k+2}]$ anwenden k=0..2n (aus 2n folgt gerade Anzahl von Teilintervallen⇒2n+1 Stützstellen) Nachteil: nur anwendbar für eine Zerlegung in eine gerade Anzahl von (einfachen) Streifen, d.h. man benötigt stets eine ungerade Anzahl von Stützwerten.

$\boxed{\begin{matrix} h = (b-a)/2n \\ x_k = a + k \cdot h \end{matrix}}$ $\boxed{\int_a^b f(x)dx = \frac{h}{3}\left(f(a) + 2\sum_{k=1}^{n-1} f(x_{2k}) + 4\sum_{k=1}^{n-1} f(x_{2k-1}) + f(b) \right) + R = S_n(h) + h^4 C}$ $\boxed{|R| \le \frac{b-a}{180} h^4 \max_{a \le x \le b} |f^{(4)}(x)|}$

Spezialfall: Quadraturformel: Formel Exakt bis Polynome vom Grad 3

$\boxed{Q_s = \frac{b-a}{6} \cdot (f(a) + 4 \cdot f(\frac{a+b}{2}) + f(b))}$ $\boxed{|R| \le \frac{(b-a)^5}{2880} \cdot \max_{a \le x \le b} |f^{(4)}(x)|}$

Richardson-Extrapolation: Allgemein gilt: Φ(x,h) ist durch Schrittweite h erhaltene Lösung. Mit zwei verschiedenen Schrittweiten h_1 und h_2 erhält man: $\Phi(x,h) = \frac{h_2^2 \Phi(x,h_1) - h_1^2 \Phi(x,h_2)}{h_2^2 - h_1^2}$ → für Sehnen-T-R: $\boxed{\int_a^b f(x)dx = \frac{4T_{2n}(\frac{h}{2}) - T_n(h)}{3} = S_{2n}(\frac{h}{2})}$

S_{2n} bezeichnet die Simpsonformel für 2n Teilinterwalle h/2.

Sehnen-Tangenten-Fehler kann verbessert werden durch Rich-Extrap. Mit größerer Stützweite sinkt der Rundungsfehler, und steigt der Sehnen-Tangenten-Fehler ⇒ optimale Stützweite.

Gauß´sche Integrationsformeln:

Der Genauigkeitsgrad der Gaussschen Quadraturformeln bei n+1 Stützstellen ist höchstens 2n+1

Numerische Differentation

f(x) ist an Stützstellen bekannt, gesucht sind die Ableitungen an diesen Stellen. Num Int gleicht Fehler einer Näherung aus, Num Diff bewirkt eine Aufrauung der Fehler.

$\boxed{Q_n = \int_a^b f(x)dy = \frac{b-a}{2} \sum_{k=0}^{n} {}_n \beta_k \cdot f\left(\frac{z_n \cdot (b-a(+b+a}{2} \right)}$ $\boxed{R_n == \frac{(b-a)^{2n+3}\left[(n+1)^4\right]}{(2n+3[2n+2)!]^3}_n \cdot f^{2n+1}(\xi) \quad \text{mit} \quad a < \xi < b}$

Ableitung durch Taylor-Reihe: (x_k,y_k sind Messwerte)

$$y'_k = \frac{y_{k+1} - y_{k-1}}{2h} - \frac{h^2}{6} f'''(\xi) = T_n(x_k) - h^2 C \qquad y''_k = \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} - \frac{h^2}{12} f^{(4)}(\xi)$$

Ableitung durchInterpolationspolynom pn(x): p_n Polynom n-ten Grades mit $p_n(x_{k+i})=y_{ki}$ (i=0..n)

$$n=1 \Rightarrow y_k' = \tfrac{1}{h}(y_{k+1} - y_k) - \tfrac{h}{2} f''(\xi) \qquad n=2 \Rightarrow y_k' = \tfrac{1}{2h}(-3y_k - 4y_{k+1} - y_{k+2}) - \tfrac{h^2}{3} f'''(\xi)$$

$$n=3 \Rightarrow y_k' = \tfrac{1}{6h}(-11y_k + 18y_{k+1} - 9y_{k+2} + 2y_{k+3}) - \tfrac{h^3}{4} f^{(4)}(\xi) \qquad y_k'' = \tfrac{1}{6h}(12y_k - 30y_{k+1} + 24y_{k+2} - 6y_{k+3}) - \tfrac{11}{12} h^2 f^{(4)}(\xi)$$

die Ableitungen an den mittleren Stützstellen (y_k) sind genauer als an den Rändern (y_{k+i}) – Fehler für y'_{k+i} nicht angegeben.

Ableitung durch Ausgleichspolynom: Ausgleichspolynom p(x) 3-ter Ordnung durch fünf benachbarte Messpunkte (x_{k+i},y_{k+i}) i=-2,-1..+2

$p(x) = a + b(x - x_k) + c(x - x_k)^2 + d(x - x_k)^3$ Messwerte y_{k+i} werden durch $p(x_{k+i})$ ersetzt. Polynome höher als 3-ten Grades sind ungeeignet, da diese eine größere Welligkeit im Interpolations- oder Ausgleichsbereich aufweisen und somit hohe Abweichungen aufweisen.

$$\Rightarrow y_k' = \tfrac{1}{12h}(y_{k-2} - 8y_{k-1} + 8y_{k+1} - y_{k+2}) - \tfrac{h^4}{30} f^{(5)}(\xi) = A_h(x_k) - h^4 C \qquad y_k'' = \tfrac{1}{12h^2}(-y_{k-2} + 16y_{k-1} - 30y_k + 16y_{k+1} - y_{k+2}) - \tfrac{h^4}{90} f^{(4)}(\xi)$$

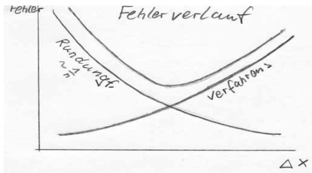
Richardson-Extrapolation: für Taylor folg $y'(x_k) = \frac{4T_h(x_k) - T_{2h}(x_k)}{3} = A_h(x_k) \Rightarrow y'(x_k) = \frac{1}{12h} \cdot (y_{k-2} - 8 \cdot y_{k-1} + 8 \cdot y_{k+1} - y_{k+2})$

Fehleranalyse: Bestimmung der optimalen Schrittweite h. Bei kleinerer Schrittweite spielt der Rundungsfehler eine größere Rolle.

$$y'_k = \frac{f(x_k + h) - f(x_k - h)}{2h} \qquad f(x_k + h) = y_{k+1} + e_1 \qquad f(x_k - h) = y_{k+1} + e_{-1}$$

$|E| \approx \frac{e}{h} + \frac{Mh^2}{6} \Rightarrow |E|' = \frac{e}{h^2} + \frac{Mh}{3}$ $\quad \Rightarrow h = \sqrt[3]{\frac{3e}{M}}$ e= Fehlertungenauigkeit der Maschine

(Ea Verfahrensfehler, E_R Rundungsfehler, E Gesamtfehler)



Numerische Behandlung von DGL AWP: $y' = f(x,y)$ $y(x_o)=y_o$

Falls $\frac{\partial f}{\partial y}$ in einer Umgebung von (x_o,y_o) existiert und stetig ist, ist das Anfangswertproblem in]x_o-ε,x_o+ε [eindeutig lösbar. LIPSCHITZ-Bedingung: f(x,y) stetig um (x_o,y_o) und es gilt L>0 mit $L \cdot |y_1-y_2| \geq | f(x_1,y_1)-f(x_2,y_2)|$

Prinzip der **Einschrittverfahren**: $x_k=x_o+k \cdot h=x_{k-1}+h$ $\quad y' = f(x, y(x)) \Rightarrow y_{k+1} = y_k + \int_{x_k}^{x_{k+1}} f(x, y(x))dx$

Eulersche Polygonzugverfahren (Rechteckregel) =Fehlerformer=$h^2/2 \cdot f'(\xi_0)$ $\xi_0 \in [x_k,x_{k+1}]$ Ordnung Verfahrensfehler≡ $O(h^2)$

$$x_{k+1} = x_k + h \qquad y_{k+1} = y_k + h \cdot f(x_k, y_k)$$

Heunsche-Prädiktor-Korrektor-Verfahren (Sehnentrapezregel) Nach der 2. Iteration abbrechen $y_{k+1}^{(0)} = y_k + h \cdot f(x_k, y_k)$ **Prädiktor**

$y_{k+1}^{(v+1)} = y_k + \frac{h}{2} \cdot f(x_k, y_k) + \frac{h}{2} \cdot f(x_{k+1}, y_{k+1}^{(v)}) = \varphi(y_{k+1}^{(v)})$ **Korrektor**

lokaler Verfahrensfehler**:** $O(h^3)$**.** Konvergenzuntersuchung: $|\varphi'| = \left|\frac{h}{2}\frac{\partial f}{\partial y}(xk+1, yk+1)\right| \leq \frac{h}{2} \cdot \max\left|\frac{\partial f}{\partial y}\right| = \frac{h}{2} \cdot L = M \leq 1$

Es muss h*L<2 sein, damit M<1 ist (Bedingung für Konvergenz) h*L heißt Schrittkennzahl, typischerweise im Intervall [0.05,0.2]

Runge-Kutta-Verfahren (Modifikation der Simpson-Regel) $y_{k+1} = y_k + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ lokaler Verfahrensfehler = $O(h^5)$

$$k_1 = h \cdot f(x_k, y_k) \Rightarrow k_2 = h \cdot f(x_k + \tfrac{h}{2}, y_k + \tfrac{k_1}{2}) \Rightarrow k_3 = h \cdot f(x_k + \tfrac{h}{2}, y_k + \tfrac{k_2}{2}) \Rightarrow k_4 = h \cdot f(x_k + h, y_k + k_3)$$

Im jeden Schritt müssen 4 Hilfsgrößen k_n berechnet werden. Es entstehen große Fehlerschwankungen bei Schrittweitenänderung wegen Verfahrensfehler-Ordnung h^5. Schrittkennzahl K=h*L (**siehe Heun)**

Mehrschrittverfahren Beim Schritt auf y_{k+1} wird nicht nur y_k sondern noch s davor liegende Wertepaare benutzt - Extrapolation.

(x_i,$f(x_i,y_i)$=(x_i;f_i) i= k-s,k-s+1,...,k

Expliziete **Adams-Bashford** für drei(s=2) & vierstufiges(s=3) Verfahren:

$$s = 2 \Rightarrow y_1 = y_0 + \tfrac{h}{12}(23f_o - 16f_{-1} + 5f_{-2})$$

$$s = 3 \Rightarrow y_1 = y_0 + \tfrac{h}{24}(55f_o - 59f_{-1} + 37f_{-2} - 9f_{-3})$$

lokaler Fehler des vierstufigen Verfahren $O(h^5)$ wie Runge-Kutta

Vorteil: geringerer Rechenaufwand, für nächsten Schritt nur einen neuen Wert berechnen ↔

Nachteil: bei Extrapolation wächst der Fehler grundsätzlich schnell, deshalb Modifikation zum Adams-Moulton-Verfahren, indem der unbekannte Punkt (x_1,f_1) mit in das Interpolationsintervall mit einbezogen wird.

$$\text{prädiktor}: y_1^P = y_0 + \tfrac{h}{12}(23f_0 - 16f_{-1} + 5f_{-2}) \text{ dreistufig}$$
$$\text{korrektor}: y_1 = y_0 + \tfrac{h}{24}(9f(x_1, y_1^P) + 19f_0 - 5f_{-1} + f_{-2})$$

$$\text{prädiktor}: y_1^P = y_0 + \tfrac{h}{24}(55f_0 - 59f_{-1} + 37f_{-2} - 9f_{-3}) \text{ vierstufig}$$
$$\text{korrektor}: y_1 = y_0 + \tfrac{h}{720}(251f(x_1, y_1^P) + 646f_0 - 264f_{-1} + 106f_{-2} - 19f_{-3})$$

DGL-System 1.Ordnung

Jede DGL n-ten Grades lässt sich auf ein DGL-System 1.Ordnung zurückführen mit n-Gleichungen

$y'' + 6 \cdot y' + 5 \cdot y = 2 - x^2 \Rightarrow y'' = 2 - x^2 - 6 \cdot y' - 5 \cdot y$ Anfangsbedingung: $u(0) = 0, \; u'(0) = 0$

$u_1 = y \qquad u_1' = u_2 \qquad \vec{u}' = \begin{pmatrix} u \\ 2 - x^2 - 5u_1 - 6u_2 \end{pmatrix} = \begin{pmatrix} f_1(x,u_1,u_2) \\ f_2(x,u_1,u_2) \end{pmatrix} \quad \vec{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \vec{u}_0 \qquad \vec{u}_1^{<0>} = \vec{u}_0 + h \cdot f(x_0, \vec{u}_0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + 0{,}1 \cdot \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0{,}2 \end{pmatrix}$

$u_2 = y' \qquad u_2' = 2 - x^2 - 5 \cdot u_1 - 6 \cdot u_2 \qquad \vec{u}_1^{<0>} = \vec{u}_0 + \frac{h}{2} \cdot f(x_0, \vec{u}_0) + \frac{h}{2} \cdot f(x_1, \vec{u}^{<0>}{}_1) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \frac{0{,}1}{2} \cdot \begin{pmatrix} 0 \\ 2 \end{pmatrix} + \frac{0{,}1}{2} \cdot \begin{pmatrix} 0{,}2 \\ 0{,}79 \end{pmatrix} = \begin{pmatrix} 0{,}01 \\ 0{,}0395 \end{pmatrix} \begin{matrix} \leftarrow y(0{,}1) \\ \leftarrow y'(0{,}1) \end{matrix}$