1 Aufgabenstellung

Ziel der Aufgabenstellung ist es folgendes Randwertproblem für Differentialgleichungen zweiter Ordnung mithilfe des sogenannten **Differenzenverfahren** zu approximieren.

Gegeben sind stetige Funktionen $b : I := [0, 1] \to \mathbb{R}$ und $c : I \times \mathbb{R} \to \mathbb{R}$ sowie $\alpha, \beta \in \mathbb{R}$. Die Funktion c erfülle die Bedingung

$$c(x, v) - c(x, w) \geq 0 \tag{1}$$

für alle $x \in I$ und $v, w \in \mathbb{R}$ mit $v \geq w$. Gesucht ist eine zweimal stetig

differenzierbare Funktion $u : [0, 1] \to \mathbb{R}$, so dass

(1a) $-u''(x) + b(x)u' + c(c, u(x)) = 0$ für alle $x \in I$

(1b) $u(0) = \alpha,\ u(1) = \beta$

Dabei ist das Differenzenverfahren wie folgt definiert: Ersetzt man in $(1a)$ die Differentialoperatoren mit den aus dem Modul 61511 bekannten Differenzenquotienten

$$g' \approx (\delta_h g) := \frac{g(x + h) - g(x - h)}{2h} \tag{2}$$

$$g'' \approx \left(\delta_h^2 g\right) := \frac{g(x + h) - 2g(x) + g(x - h)}{h^2} \tag{3}$$

für die erste bzw. die zweite Ableitung und überzieht das Intervall I mit einem Gitter $\Delta : 0 = x_0 < x_1 < \cdots < x_N = 1$ - wobei die Abstände zwischen den einzelnen Stützpunkten überall gleich sein soll, also: $x_i := i/N, i = 0, 1, \ldots, N, h := 1/N$ - erhält man folgende Aufgabenstellung die wir zu lösen haben.

Gesucht ist eine Funktion $u^N : \Delta \to \mathbb{R}$, die die Randbedingungen

$(2a)$ $u^N(x_0) = \alpha$ und $u^N(x_N) = \delta$

und die Differenzengleichung

$(2b)$ $-\left(\delta_h^2 u^N\right)(x_i) + b(x_i)\left(\delta_h u^N\right)(x_i) + c\left(x_i, u^N(x_i)\right) = 0$

für alle $i = 1, \ldots, N - 1$ erfüllt.

2 Lösungsansatz

Wir beschränken uns zuerst auf den linearen Fall, also $c(x,\nu) = d(x)\nu - f(x)$ und widmen uns anschließend dem allgemeinen Fall. Im folgenden benennen wir die $u^N(x_i)$ der übersichthalber um in u_i. Außerdem kürzen wir bei Referenz auf Modul 61511 ab durch *NUM*.

2.1 Linearer Fall

Mit (2) und (3) erhalten wir zunächst

$(\delta_h u)(x_i) = \frac{u_{i+1}-u_{i-1}}{2h}$ und

$\left(\delta_h^2 u\right)(x_i) = \frac{u_{i+1}-2u_i+u_{i-1}}{h^2}$

Einsetzen in $(1a)$ ergibt:

$$-\left[\frac{u_{i+1}-2u_i+u_{i-1}}{h^2}\right] + b(x_i)\left[\frac{u_{i+1}-u_{i-1}}{2h}\right] + d(x_i)u_i - f(x_i) = 0, \quad \text{bzw.}$$
$$-\left[\frac{u_{i+1}-2u_i+u_{i-1}}{h^2}\right] + b(x_i)\left[\frac{u_{i+1}-u_{i-1}}{2h}\right] + d(x_i)u_i = f(x_i)$$

Wir fassen nun die Terme u_{i-1}, u_i, u_{i+1} zusammen. Anschließend müssen wir noch beachten das wegen $(2a)$ $u_0 = \alpha$ und $u_N = \delta$ gilt.

Es gilt allgemein für x_i, $i = 1, \ldots, N-1$:

$$u_{i-1}\left[-\frac{1}{h^2} - \frac{b(x_i)}{2h}\right] + u_i\left[\frac{2}{h^2} + d(x_i)\right] + u_{i+1}\left[\frac{b(x_i)}{2h} - \frac{1}{h^2}\right] = f(x_i)$$

Dies führt auf ein lineares Gleichungssystem mit $N-1$ Unbekannten u_i. Definieren wir nun noch die Koeffizienten der u_i's in der obigen Gleichung durch:

$d_i := \frac{2}{h^2} + d(x_i)$

$f_i := \frac{b(x_i)}{2h} - \frac{1}{h^2}$ und

$e_i := -\frac{1}{h^2} - \frac{b(x_i)}{2h}$,

so können wir das lineare Gleichungssystem in Matrix-form darstellen als:

$$\begin{pmatrix} d_1 & f_1 & & & 0 \\ e_2 & d_2 & f_3 & & 0 \\ & \ddots & \ddots & \ddots & \\ & & e_{N-2} & d_{N-2} & f_{N-2} \\ 0 & & & e_{N-1} & d_{N-1} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{pmatrix} = \begin{pmatrix} f(x_1) + \alpha\left[\frac{1}{h^2} + \frac{b(x_1)}{2h}\right] \\ f(x_2) \\ \vdots \\ f(x_{N-2}) \\ f(x_{N-1}) + \beta\left[\frac{1}{h^2} + \frac{b(x_{N-1})}{2h}\right] \end{pmatrix}$$

Dies ist eine **Tridiagonale Matrix** (vgl. *NUM*, Def. 6.3.4) welches man mit dem **Thomas-Algorithmus** (*NUM*, 6.3.5) lösen kann.

2.2 Allgemeiner Fall

Für den allgemeinen Fall setzen wir wieder mit (2) und (3) in (1a) ein und erhalten:

$$-\frac{u_{i+1}-2u_i+u_{i-1}}{h^2}+b(x_i)\frac{u_{i+1}-u_{i-1}}{2h}+c\left(x_i,u_i\right)=0 \tag{4}$$

Sei

$$\mathbf{u}:=\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{pmatrix}$$

Für jedes $i=1,\ldots,N-1$ definieren wir die Funktion $F_i\left(u_{i-1},u_i,u_{i+1}\right)$ durch (4). Dies resultiert in $N-1$ nichtlinearen Funktionen die wir zusammenfassen:

$$F(u):=\begin{pmatrix} F_1\left(u_0,u_1,u_2\right) \\ F_2\left(u_1,u_2,u_3\right) \\ \vdots \\ F_{N-1}\left(u_{N-2},u_{N-1},u_N\right) \end{pmatrix}=0$$

Um die Unbekannten u_i zu finden, bemühen wir das Newton-Verfahren (vgl. Abschnitt 2.3, *NUM*). Die Iterationsvorschrift für die k-te Iterierte erhält man durch:

$u^{k+1}=u^k-J^{-1}F\left(u^k\right)$

wobei J die **Jacobi-Matrix** ist. Wie genau sieht diese aus?

Wir definieren $J_{ij}:=\frac{\partial F_i}{\partial u_j}$, mit $i,j=1,\ldots,N-1$. Jedes F_i hängt nur von den Werten u_{i-1},u_i und u_{i+1} ab. Also ist die Jacobi-Matrix eine Tridiagonalmatrix in der auf jeder Zeile die partiellen Ableitungen der einzelnen F_i's im Bezug auf die drei Variablen stehen.

Auf der Hauptdiagonalen stehen

$J_{ii}=\frac{2}{h^2}+\frac{\partial c(x_i,u_i)}{\partial u_i}$

und auf den Nebendiagonalen:

$J_{i,i-1}=-\frac{1}{h^2}-\frac{b(x_i)}{2h}$

$J_{i,i+1}=-\frac{1}{h^2}+\frac{b(x_i)}{2h}$

Daraus folgt für die Jacobi-Matrix J:

$$J = \begin{pmatrix} J_{11} & J_{12} & & & & 0 \\ J_{21} & J_{22} & J_{23} & & & 0 \\ & \ddots & \ddots & \ddots & & \\ & & J_{N-2,N-3} & J_{N-2,N-2} & J_{N-2,N-1} & \\ 0 & & & J_{N-1,N-2} & J_{N-1,N-1} & \end{pmatrix}$$

Bei einer Implementierung auf einem Computer wird man die Inverse von J nicht berechnen. Vielmehr löst man das lineare Gleichungssystem $J\left(x^k\right) s^k = -F\left(x^k\right)$ für s^k. Da die Jaxobi Matrix ebenfalls von tridiagonaler Gestalt ist, nehmen wir dazu wieder den Thomas Algorithmus zur Hilfe. Anschließend setzt man $x^{k+1} := x^k + s^k$.

3 Richardson-Extrapolation

Nachdem man mit den Mitteln aus dem vorherigen Abschnitt Lösungen für verschiedene N ermittelt hat, kann man mit der sogenannten **Richardson-Extrapolation** (vgl. Abschnitt 5.6, *NUM*) durch Kombination zweier Lösungen U^N und U^{2N} eine bessere Approximation der Lösung u erreichen.
Die Richardson Extrapolation ist eine Methode zur Verbesserung der Genauigkeit numerischer Approximationen, indem Fehler höherer Ordnung systematisch reduziert werden. Sie basiert auf der Idee, dass der Fehler einer numerischen Lösung als eine asymptotische Entwicklung des Gitterabstandes h dargestellt werden kann. Gegeben zwei Approximationen U^N und U^{2N}, die mit Gitterabständen $N/2$ und N berechnet wurden, gilt unter der Annahme einer Fehlerordnung $\mathcal{O}\left(h^p\right)$:

$$U^N = u + c_1 h^p + \mathcal{O}\left(h^{p+1}\right),$$

$$U^{2N} = u + c_1 \left(h/2\right)^p + \mathcal{O}\left(h^{p+1}\right),$$

mit einer gewissen Konstanten $c_1 \in \mathbb{R}$.

Durch Linearkombination dieser beiden Lösungen eliminiert man den führenden Fehlerterm:

$$U_{extrap} = U^{2N} + \frac{U^{2N} - U^N}{\left(\frac{N}{N/2}\right)^k - 1} \tag{5}$$

Das Ergebnis U_{extrap} ist eine genauere Approximation von u, da der Fehlerterm $\mathcal{O}\left(h^p\right)$ entfernt wurde. k nennt man die **Konvergenzordnung** des Verfahrens und es wird sich herausstellen das in dem uns vorliegenden Problem $k = 2$ ist. Das bedeutet dass der Fehler proportonial zu h^2 ist: Wenn h halbiert wird, verringert sich der Fehler etwa um den Faktor 4.

4 Numerische Resultate

Für den linearen Fall betrachten wir das Problem

$$-u''(x) + u'(x) + 2u(x) = -6, \qquad u(0) = 0, \quad u(1) = 0 \tag{6}$$

Die exakte Lösung von 6) ist

$$u(x) = \frac{3e^{-x}e(e+1)}{e^2+e+1} + \frac{3e^{2x}}{e^2+e+1} - 3$$

Tabelle 1 zeigt den maximalen Fehler $|u - U^N|$ für verschiedene N. Tabelle 2 zeigt die Verbesserung der Approximation nach Anwendung von Richardson-Extrapolation. Man beobachtet außerdem das man in der Extrapolation nur die gemeinsamen Gitterpunkte für zwei verschiedene N, die des gröberen Gitters, benutzen kann und sich deswegen nach zweifacher Anwendung der Richardson-Extrapolation die Tabelle um zwei Zeilen verringert.

N	Max Error
8	0.00068
16	0.00017
32	4.27776e-05
64	1.06989e-05
128	2.67529e-06
256	6.68834e-07
512	1.67210e-07
1024	4.18026e-08
2048	1.04505e-08

Tabelle 1

N	Max Error	Extrapolation (1. Durchlauf)	Extrapolation (2. Durchlauf)
8	0.00068	9.32280e-07	1.87039e-10
16	0.00017	5.89323e-08	2.93842e-12
32	4.27776e-05	3.68599e-09	4.89608e-14
64	1.06989e-05	2.30549e-10	2.56461e-14
128	2.67529e-06	1.43864e-11	1.89737e-13
256	6.68834e-07	7.24753e-13	1.23789e-13
512	1.67210e-07	7.44959e-14	5.77038e-13

Tabelle 2

Es macht sich hier bemerkbar, dass unser Rechner mit Gleitkommazahlen (eps=e-16) arbeitet und nicht mit reellen Zahlen, weshalb die Konvergenz in Spalte 4 im zweiten Durchlauf bei $N = 64$ abreißt.

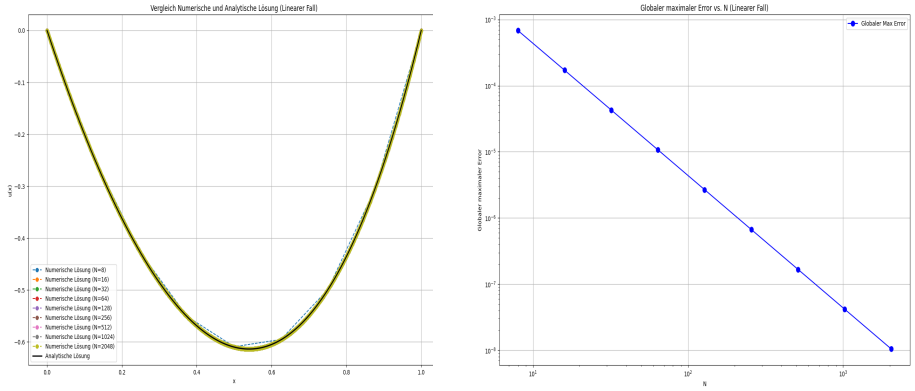


Abbildung 1

In **Abbildung 1** links sieht man die analytische Lösung u des Problems 5) und die verschiedenen Approximationen von u. Optisch ist für die Approximationen für N größer 8 kaum noch ein Unterschied zu erkennen. Der doppelt logarithmische Plot rechts bestätigt diese Beobachtung und zeigt die Konvergenz des Verfahrens.

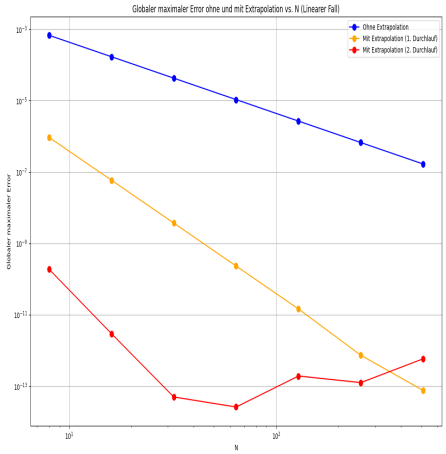


Abbildung 2

In **Abbildung 2** sieht man den globalen maximalen Error $|u - U^N_{extrap}|$. Man erkennt das mit vergleichsweise kleinem Rechenaufwand eine deutlich bessere Approximation erreicht wird. Für $N = 8$ beträgt der maximale Error nach zweifacher Anwendung der Richardson-Extrapolation ungefähr $e - 10$ im Vergleich zum Ursprünglichen $e - 3$.

Für den allgemeinen Fall betrachten wir das Problem

$$-u''(x) + (e^x + 1)\,u'(x) + u^2(x) = 0, \qquad u(0) = -1, \quad u(1) = -e \tag{7}$$

Die exakte Lösung von 7) ist

$$u(x) = -e^x$$

Wie im linearen Fall, zeigt **Tabelle 3** den maximalen Fehler $|u - U^N|$ für verschiedene N und **Tabelle 4** zeigt den maximalen Fehler $|u - U^N_{extrap}|$. Man sieht das für $N = 128$ im zweiten Durchlauf der Richardson-Extrapolation bereits eine Genauigkeit im Bereich der Maschinengenauigkeit erreicht wird.

N	Max Error
8	0.00174
16	0.00043
32	0.00010
64	2.71750e-05
128	6.79232e-06
256	1.69799e-06
512	4.24492e-07
1024	1.06122e-07
2048	2.65306e-08

Tabelle 3

N	Max Error	Extrapolation (1. Durchlauf)	Extrapolation (2. Durchlauf)
8	0.00174	8.40313e-06	9.48759e-09
16	0.00043	5.16301e-07	1.51042e-10
32	0.00010	3.21314e-08	2.35234e-12
64	2.71750e-05	2.00607e-09	3.68594e-14
128	6.79232e-06	1.25346e-10	8.88178e-16
256	1.69799e-06	7.83417e-12	8.88178e-16
512	4.24492e-07	4.89830e-13	8.88178e-16

Tabelle 4

Es ist außerdem erstaunlich dass das Newton-Verfahren für alle Werte von N nach bereits 4 Iterationen konvergiert. Dabei wird für den Wert s^k in der k-ten Iteration untersucht ob $\|s^k\|_\infty < \mathsf{tol}$ ist, wobei tol eine Konstante ist die in unserem Fall auf $e - 6$ gesetzt wurde.

Außerdem stellt sich die Frage welche Werte als Startwerte $\tilde{U}$ im Newton-Verfahren in Frage kommen. Am besten funktioniert hier die lineare Interpolation von α und β in der Form

$$\tilde{U}_i = \alpha + (\beta - \alpha)\,x_i^N, \qquad i = 1, \ldots, N$$

wobei $x_i^N \in \Delta$, die Stützpunkte des Gitters für N sind.

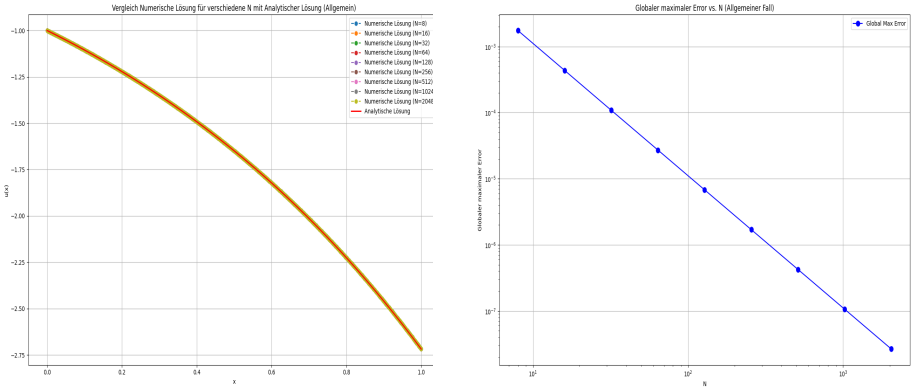


Abbildung 3

In **Abbildung 3** links werden wieder Lösungen des Verfahrens für verschiedene N geplottet. Es ist Optisch kein Unterschied der Approximationen mit der analytischen Lösung zu erkennen. Rechts kann man die Konvergenz des Verfahrens sehen.

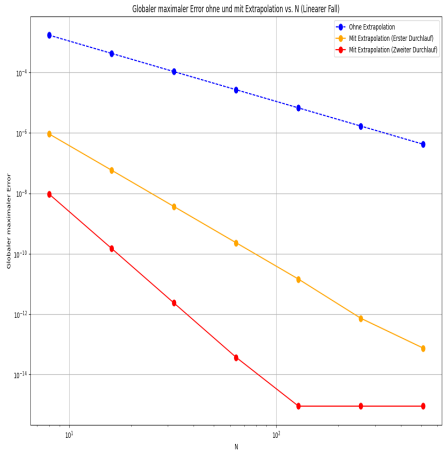


Abbildung 4

Abbildung 4 zeigt den Plot der maximalen Fehler für verschiedene N ohne und mit Anwendung der Richardson-Extrapolation. Es ist an dieser Stelle zu bemerken dass aufgrund der in *NUM* gezeigten $\mathcal{Q}$-quadratischen Konvergenz des Newton-Verfahrens, die Konvergenzordnung k in 5) im ersten Durchlauf der Richardson-Extrapolation gleich 2 ist. Im zweiten Durchlauf hingegen muss man $k = 4$ setzen. Das kommt daher dass der Fehler nach einmaliger Anwendung der Richardson-Extrapolation von $\mathcal{O}\left(h^2\right)$ auf $\mathcal{O}\left(h^4\right)$ sinkt. Gleiches beobachtet man im linearen Fall.

Zum Schluss untersuchen wir noch das Problem

$$-u''(x) + 2u'(x) + u(x) = \frac{1}{4x^{3/2}} + \frac{1}{x^{1/2}} + x^{1/2} \tag{8}$$

mit

$$u(0) = 0, u(1) = 1$$

Die Lösung von (8) ist

$$u(x) = \sqrt{x}$$

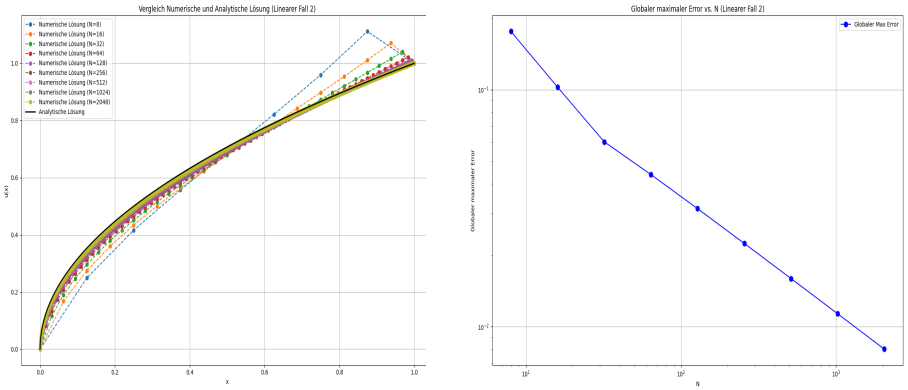
Für die Ableitungen von $u(x)$ gelten

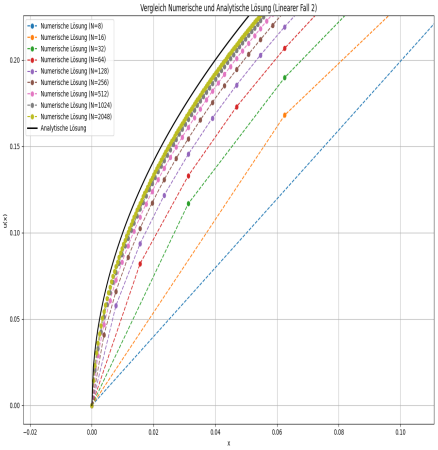
$$u'(x) = \tfrac{1}{2\sqrt{x}} \qquad \text{und} \qquad u''(x) = -\tfrac{1}{4x^{3/2}}.$$

Man erkennt das die Ableitungen Singularitäten bei $x = 0$ besitzen, also für $x \to 0$ unbeschränkt sind. Das Differenzenverfahren approximiert die erste und zweite Ableitung von $u(x)$ auf einem Gitter mit Gitterabstand N, weshalb man bei $x = 0$ einen größeren Fehler erwarten wird.

Betrachtet man Abbildung 5 auf der nächsten Seite, besonders den Plot in der Mitte, so erkennt man das auch für $N = 2048$ die Approximation nie ganz die analytische Lösung erreicht. Auch der Vergleich des maximalen Errors mit dem des Problems (6) auf den Seiten 5-6 zeigt das die Konvergenz des Verfahrens sehr langsam ist.

Eine Möglichkeit mit Problemen dieser Art umzugehen ist die Verwendung eines **nicht-äquidistanten Gitters**, welches in Bereichen mit starker Variation (wie nahe bei $x = 0$) kleinere Gitterabstände benutzt. Ebenso kann man ein **adaptives Gitter** benutzen, dass basierend auf der Lösung die Gitterpunkte anpasst.


Vergleich Numerische und Analytische Lösung (Linearer Fall 2)
Numerische Lösung (N=8)
Numerische Lösung (N=16)
Numerische Lösung (N=32)
Numerische Lösung (N=64)
Numerische Lösung (N=128)
Numerische Lösung (N=256)
Numerische Lösung (N=512)
Numerische Lösung (N=1024)
Numerische Lösung (N=2048)
Analytische Lösung
u(x)
x
Globaler maximaler Error vs. N (Linearer Fall 2)
Globaler Max Error
Globaler maximaler Error
N


Vergleich Numerische und Analytische Lösung (Linearer Fall 2)
Numerische Lösung (N=8)
Numerische Lösung (N=16)
Numerische Lösung (N=32)
Numerische Lösung (N=64)
Numerische Lösung (N=128)
Numerische Lösung (N=256)
Numerische Lösung (N=512)
Numerische Lösung (N=1024)
Numerische Lösung (N=2048)
Analytische Lösung
u(x)
x

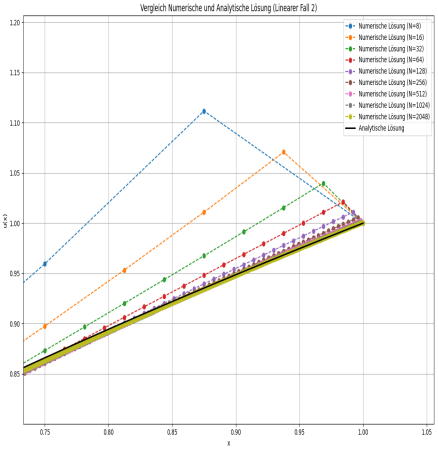

Vergleich Numerische und Analytische Lösung (Linearer Fall 2)
Numerische Lösung (N=8)
Numerische Lösung (N=16)
Numerische Lösung (N=32)
Numerische Lösung (N=64)
Numerische Lösung (N=128)
Numerische Lösung (N=256)
Numerische Lösung (N=512)
Numerische Lösung (N=1024)
Numerische Lösung (N=2048)
Analytische Lösung
u(x)
x

Abbildung 5