

# DAS EULER-VERFAHREN

*Praktische Aufgabe P2 im Modul Math-Ba-NUM*

**Lars Ortscheidt & Eric Kunze**

## 1. Analytische Betrachtung der Anfangswertprobleme

Im Folgenden betrachten wir die beiden Anfangswertprobleme

$$u' = \beta(1 - u)u \quad \text{mit } x \in (0, 10] \text{ und } u(0) = \alpha > 0 \quad (1)$$

und

$$u' = 10(\sin(x) - \cos(5x)) \quad \text{mit } x \in (0, 10] \text{ und } u(0) = 10 \quad (2)$$

Die Differentialgleichung in (1) hat entsprechend der Vorlesung GDIM als Bernoulli-Gleichung mit Parameter 2 die allgemeine Lösung

$$u(x) = \frac{e^{\beta x}}{e^{\beta x} + c}$$

Damit hat (1) mit dem Anfangswert  $u(0) = \alpha$  die Lösung

$$u(x) = \frac{\alpha \cdot e^{\beta x}}{1 - \alpha \cdot (1 - e^{\beta x})}$$

Für den Wert an der Stelle  $x = 10$  ergibt sich mit  $\alpha = 10$  und  $\beta = 2$

$$u(10) = \frac{10e^{20}}{10e^{20} - 9} \approx 1,000000001855038263635868998339029786$$

und als Grenzwert mithilfe der Regel von L'Hôpital

$$\lim_{x \rightarrow \infty} u(x) = 1$$

Wir lösen nun (2) mittels elementarer Integration, denn die rechte Seite ist von  $u$  un-

abhängig. Somit ist

$$\begin{aligned} u(x) &= \int u(x) \, dx = \int 10(\sin(x) - \cos(5x)) \, dx \\ &= 10 \left( \int \sin(x) \, dx - \int \cos(5x) \, dx \right) \\ &= -10 \cos(x) - 2 \sin(5x) + c \end{aligned}$$

Zur Bestimmung von  $c$  setzen wir die Anfangswertbedingung ein, d.h.

$$u(0) = -10 \cos(0) - 2 \sin(5 \cdot 0) + c = -10 + c \stackrel{!}{=} 10 \quad \implies \quad c = 20$$

Somit ergibt sich die analytische Lösung von (2) zu  $u(x) = -10 \cos(x) - 2 \sin(5x) + 20$ . Für den Wert an der Stelle  $x = 10$  ergibt sich damit

$$u(10) = 28.915$$

## 2. Die implementierten Verfahren

Im File `ex_euler.m` wird das explizite Eulerverfahren implementiert. Ausgehend von der Verfahrensfunktion  $\Phi(x, y, z, h) := f(x, y)$  erhalten wir die Verfahrensvorschrift

$$y^{k+1} = y^k + h_k f(x_k, y^k)$$

Der Term  $x_k$  berechnet sich in unserem Fall eines äquidistanten Gitters auch durch  $x_k = x_0 + k \cdot h = a + k \cdot h$ . In Octave sieht dies wie folgt aus:

```
1 % Lars Ortscheidt Eric Kunze P2 Octave
2 function y = ex_euler(fun,h,a,b,y0)
3     n = (b-a)/h; % Anzahl der Schritte
4     y(1) = y0; % Anfangswert
5     for i = 1:n
6         y(i+1) = y(i) + h * fun( a + i*h , y(i) );
7     endfor
8 endfunction
```

Das verbesserte Polygonzugverfahren ist gegeben durch die Iterationsvorschrift

$$\begin{aligned} y^{k+1} &= y^k + \frac{h_k}{2} \left[ f(x_k, y^k) + f(x_k + h_k, y^k + h_k f(x_k, y^k)) \right] \\ &= y^k + \frac{h}{2} \left[ f(x_k, y^k) + f(x_k + h, y^k + h f(x_k, y^k)) \right] \end{aligned}$$

Der  $x_k$  berechnet sich wieder wie oben mit äquidistantem Gitter als  $x_k = a + k \cdot h$ . Analog ergibt sich dann natürlich auch  $x_k + h = a + k \cdot h + h = a + (k + 1) \cdot h$ . Praktisch umgesetzt sieht dies wie folgt aus:

```

1  % Lars Ortscheidt Eric Kunze P2 Octave
2  function y = poly(fun,h,a,b,y0)
3
4      n      = (b-a)/h; % Anzahl der Schritte
5      y(1) = y0;        % Anfangswert
6
7      for i = 1:n
8          y(i+1) = y(i) + h/2 * (fun(a+(i)*h , y(i)) +
9                                fun(a+(i+1)*h , y(i)+h*fun(a+(i)*h , y(i))
10                               ));
11      endfor
12  endfunction

```

Das implizite Eulerverfahren für (1) lässt sich mir der Vorschrift

$$y^{k+1} = y^k + h_k \cdot \beta(1 - y^{k+1})y^{k+1} \quad (3)$$

ausdrücken. Dieses können wir in eine Nullstellengleichung umformen, d.h. (3) ist mittels einfacher Umformungen äquivalent zu

$$0 = (y^{k+1})^2 + \frac{1 - h_k \beta}{h_k \beta} y^{k+1} - \frac{1}{h_k \beta} y^k$$

Daraus ergibt sich mit der  $p$ - $q$ -Formel für quadratische Gleichungen

$$y^{k+1} = -\frac{1 - h_k \beta}{2h_k \beta} \pm \sqrt{\left(\frac{1 - h_k \beta}{2h_k \beta}\right)^2 + \frac{1}{h_k \beta} y^k}$$

Davon wählen wir nur die positive Lösung aus, da in der Population lediglich nichtnegative Werte von Interesse sein sollten und die Bedeutung einer negativen Population nicht erklärbar ist. Also erhalten wir eine Iterationsvorschrift mit

$$y^{k+1} = -\frac{1 - h_k \beta}{2h_k \beta} + \sqrt{\left(\frac{1 - h_k \beta}{2h_k \beta}\right)^2 + \frac{1}{h_k \beta} y^k}$$

Dies ist schließlich auch in der praktischen Realisierung wiederzufinden:

```

1  % Lars Ortscheidt Eric Kunze P2 Octave
2  function y = im_euler(fun,h,a,b,y0, beta)
3      n      = (b-a)/h;  % Anzahl der Schritte
4      y(1)   = y0;       % Anfangswert
5      phalbe = (1-h*beta)/(2*h*beta); % p/2 in p-q-
        Formel
6      for i = 1:n
7          % Herleitung fuer AWP (1) mit pq Formel
8          y(i+1) = -phalbe + sqrt((phalbe^2 + y(i)/(h*
                beta)));
9      endfor
10 endfunction

```

### 3. Näherungsweise Lösung der Anfangswertprobleme

Wir setzen zunächst die Parameter  $\alpha := 10$  und  $\beta := 2$  und berechnen mit den obigen Verfahren eine Näherung  $u_h(10)$  für  $u(10)$ . Beim expliziten Euler-Verfahren und dem verbesserten Polygonzugverfahren starten wir mit  $h = h_1 = \frac{1}{100}$  und halbieren in sieben Schritten jeweils den Gitterabstand. Das implizite Euler-Verfahren soll mit  $h = h_1 = \frac{1}{5}$  starten.

Dabei ergibt sich eine Lösungskurve, die in Abbildung Abb. 1 ersichtlich ist.

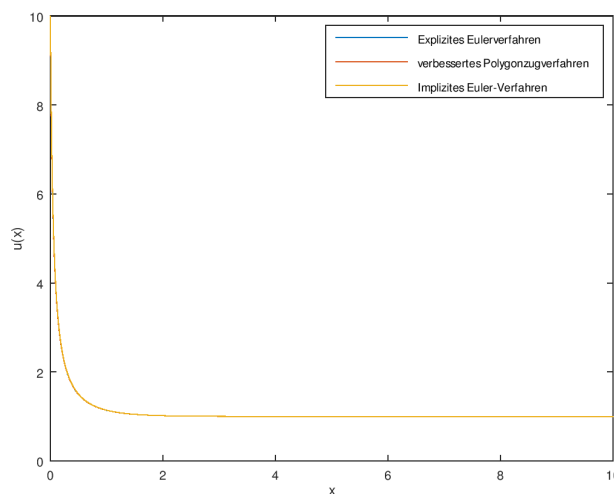


Abbildung 1: Numerische Lösung des Anfangswertproblems (1)

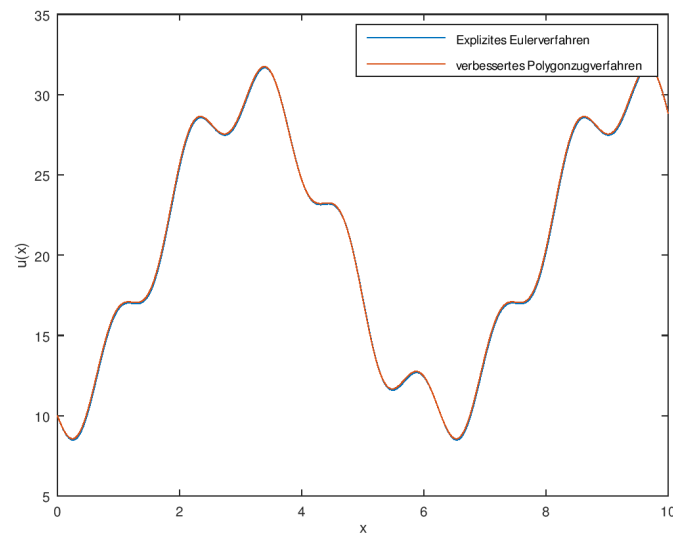
Betrachten wir hier schon einmal die Näherung in der Stelle  $x = 10$ . Dabei wird jeder Schrittweite  $h_i := \frac{1}{100} \cdot 0.5^{i-1}$  die entsprechende Näherung  $u_{h_i}(10)$  zugeordnet.

$i$	$u_{h_i}(10)$ <b>Euler explizit</b>	$u_{h_i}(10)$ <b>Polygonzug</b>	$u_{h_i}(10)$ <b>Euler implizit</b>
0	1.0000000014427488	1.0000000018612263	1.0000001083003349
1	1.000000001638212	1.0000000018565394	1.0000000182649336
2	1.0000000017438231	1.0000000018554076	1.0000000063144361
3	1.0000000017987125	1.0000000018551316	1.0000000035060665
4	1.0000000018266939	1.0000000018550623	1.0000000025675782
5	1.0000000018408197	1.0000000018550432	1.0000000021864111
6	1.0000000018479125	1.0000000018550437	1.0000000020148434
7	1.0000000018514767	1.000000001855037	1.00000000193549

**Tabelle 1:** Näherungslösung für (1)

Zum Vergleich noch einmal die exakte Lösungen:  $u_1(10) \approx 1,00000000185503826363587$ .

Für das Anfangswertproblem (2) verwenden wir ebenso das explizite Eulerverfahren und das verbesserte Polygonzugverfahren mit einer Anfangsschrittweite von  $h = 2$ , die dann in 7 Schritten jeweils halbiert wird. Damit ergibt sich das Bild der numerischen Lösung in Abb. 2.



**Abbildung 2:** Numerische Lösung des Anfangswertproblems (2)

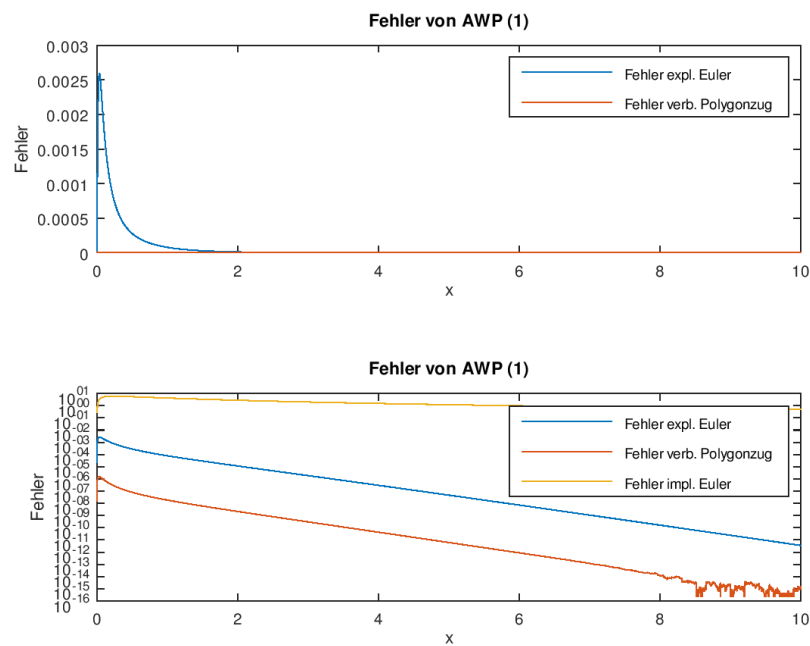
Auch hier seien die Näherung für den Wert  $u(10)$  bereits erwähnt. Erneut ordnen wir jeder Schrittweite  $h_i := \frac{1}{100} \cdot 0.5^{i-1}$  die entsprechende Näherung  $u_{h_i}(10)$  z.

$i$	$u_{h_i}(10)$ <b>Euler explizit</b>	$u_{h_i}(10)$ <b>Polygonzug</b>
0	15.942532391769284	2.6172434568951584
1	22.530915535595341	14.631286724808701
2	26.951452035021134	23.090920578941571
3	28.113171157763215	26.588114788175627
4	28.556201604193728	27.997347990269034
5	28.746142175550531	28.525781863185799
6	28.833371052366854	28.738458644584565
7	28.875059287667515	28.831446857490462

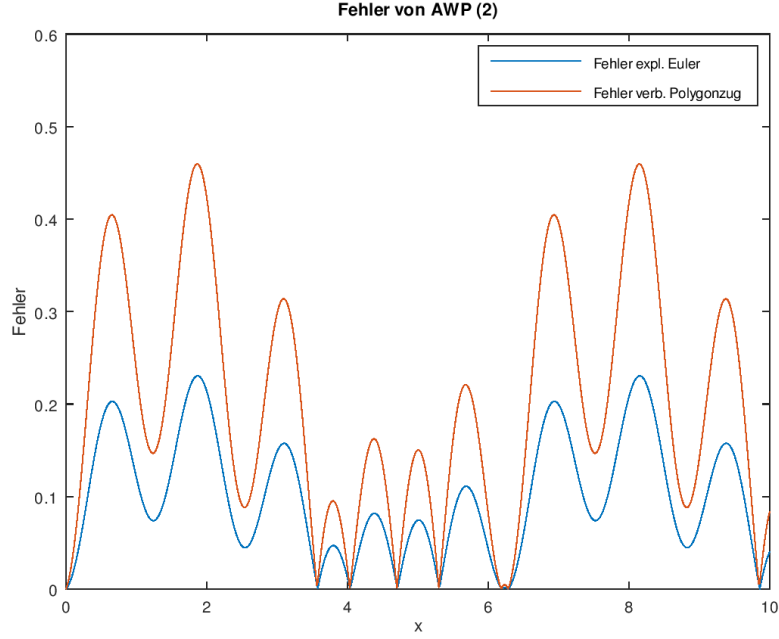
**Tabelle 2:** Näherungslösung für (2)

Zum Vergleich beträgt die exakte Lösung  $u(10) = 28.915$

## 4. Fehlerbetrachtung



**Abbildung 3:** Fehler in der Näherungslösung des AWP (1)



**Abbildung 4:** Fehler in der Näherungslösung des AWP (2)

Wir berechnen die experimentelle Konvergenzordnung  $EOC(h_{N_1}, h_{N_2})$ , wobei  $h_i := \frac{1}{100} \cdot 0.5^{i-1}$ .

$i$	Euler explizit	Polygonzug	Euler implizit
0	0.92711785521996615	2.0435143167844134	2.4979817240564346
1	0.96318524424129315	2.0232350778822767	1.7824201383743983
2	0.98148609898967798	1.9853269356211445	1.3884848412381854
3	0.99073108198050697	1.9593580155026575	1.1912373511171221
4	0.99528357306620763	2.2954558835261714	1.0943838138638462
5	0.99664696093629146	2.1255308820838581	1.047256320826385
6	1.0005395605775844	1.9999999999999951	0.98473008390473871

**Tabelle 3:** Experimentelle Konvergenzordnung für AWP (1)

$i$	Euler explizit	Polygonzug
0	1.0228479237946984	0.8805471936163225
1	1.700780365142474	1.2942009292463943
2	1.2916018258023403	1.3234568924745176
3	1.1590887771215856	1.3419383854865135
4	1.0852655001393778	1.2363765324725053
5	1.0444287073921383	1.1385003484406608
6	1.0227166290239404	1.0750283818189208

**Tabelle 4:** Experimentelle Konvergenzordnung für AWP (2)

Für das Anfangswertproblem (1) ist anhand von Abb. 3 erkennbar, dass das verbesserte Polygonzugverfahren einen geringeren Fehler aufweist als das explizite Eulerverfahren. Außerdem ist eine ungefähr doppelt so große experimentelle Konvergenzordnung zu erkennen. Damit konvergiert also das verbesserte Polygonzugverfahren schneller gegen die exakte Lösung. Das implizite Eulerverfahren dagegen weist einen deutlich größeren Fehler auf als die beiden anderen Verfahren (man beachte allerdings die gröbere Schrittweite).

Betrachten wir das Anfangswertproblem (2). Es ist aus Abb. 4 ersichtlich, dass hier das explizite Eulerverfahren einen geringeren Fehler als das verbesserte Polygonzugverfahren aufweist. Somit lässt sich also schließen, dass die Aufgabe (2) schlechter für das verbesserte Polygonzugverfahren geeignet ist als (1). Dies spiegelt sich auch in der experimentellen Konvergenzordnung wieder, die beim expliziten Eulerverfahren im Vergleich zu (1) ungefähr gleich geblieben ist (bzw. leicht gestiegen). Das Polygonzugverfahren für (2) konvergiert hingegen nur ungefähr halb so schnell wie für (1) und damit ungefähr genauso schnell wie das explizite Eulerverfahren.