Praktikum 1 : DGL

André Harms, Oliver Steenbuck

02.11.2011

Inhaltsverzeichnis

1	Steife Differentialgleichungen								
	1.1	Betrachtete Gleichung	2						
	1.2	Simulink/Analogrechner	2						
	1.3	Iterationsgleichungen	3						
		1.3.1 Euler (expl)	3						
		1.3.2 Euler (impl)	3						
		1.3.3 Runge-Kutta 2	3						
	1.4	Matlab Programme	3						
		1.4.1 Euler (expl)	3						
		1.4.2 Euler (impl)	3						
		1.4.3 RungeKutta	4						
		1.4.4 stiff	5						
	1.5	Ergebnisausdrucke	5						
	1.6	Interpretation der Ergebnisse	5						
2	Van-der-Pol-DGL								
	2.1	Betrachtete Gleichung	5						
	2.2	Simulink/Analogrechner	5						
	2.3	Zu DGL 1 Ordnung transformierte DGL	5						
	2.4	ŭ	6						
		2.4.1 Euler (expl)	6						
		· - /	6						
	2.5	Ergebnisausdrucke	6						
	2.6	Interpretation der Ergebnisse	6						
3	Lore	enz-Attraktor mit RK 2	6						
	3.1	Simulink/Analogrechner	_						
	3.2	Iterationsgleichungen							

MT, Pareigis		Abbildungsverzeichnis				Praktikum 1				
3.3	3.3.1 Ergebr	b Programme					6 6 6			
3.5 Abbi l		gsverzeichnis		٠	•	•	6			
1 2		ink-Schaltbild: Steife DGL					2 5			
Listir	ngs									
1 2 3	Impliz	zites Euler-Verfahren					3 4			
1 St	eife C	Differentialgleichungen								
1.1 B	etrach	ntete Gleichung								

$$y(0) = 1 \tag{1}$$

$$y' = 10 - 500 \cdot y + 5000 \cdot x \tag{2}$$

1.2 Simulink/Analogrechner

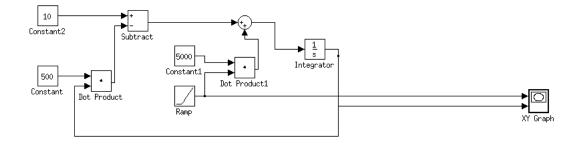


Abbildung 1: Simulink-Schaltbild: Steife DGL

Generiert am: 13. Oktober 2011

Oliver Steenbuck Andre Harms

2/6

1.3 Iterationsgleichungen

1.3.1 **Euler** (expl)

$$u_{i+1} = u_i + h \cdot (10 - 500 \cdot u_i + 5000 \cdot x_i) \tag{3}$$

1.3.2 **Euler (impl)**

1.3.3 Runge-Kutta 2

$$k_{1_j} = f(x_j, u_i) \tag{4}$$

$$k_{2_j} = f(x_j + h, u_j + h * k_{1_j})$$
(5)

$$u_{j+1} = u_j + \frac{h}{2} \cdot (k_{1_j} + k_{2_j}) \tag{6}$$

1.4 Matlab Programme

1.4.1 **Euler (expl)**

```
Listing 1: Explizites Euler-Verfahren
```

```
_{1} function [x, y] = euler_{-}expl(h, xend, f)
      ska_y_i = 1;
      vec_ytmp = [];
      vec_xtmp = [];
      vec_range = [0:h:xend];
      for ska_cur_x=vec_range
           ska_y_i = ska_y_i + h * f(ska_cur_x, ska_y_i);
          vec_xtmp = [vec_xtmp, ska_cur_x];
9
          vec_ytmp = [vec_ytmp, ska_y_i];
      end
11
12
      y = vec_ytmp;
14
      x = vec_xtmp;
_{15} end
```

1.4.2 **Euler (impl)**

Listing 2: Implizites Euler-Verfahren

```
_{1} function [x, y] = euler_{-}expl(h, xend)
```

Generiert am: 13. Oktober 2011

Oliver Steenbuck Andre Harms

3/6

```
ska_y_i = 1;
2
      vec_ytmp = [];
3
      vec_xtmp = [];
4
5
      vec\_range = [0:h:xend];
6
      for ska_cur_x=vec_range
           ska_y_i = ska_y_i + h * mtp0101(ska_cur_x, ska_y_i);
           vec_xtmp = [vec_xtmp, ska_cur_x];
9
           vec_ytmp = [vec_ytmp, ska_y_i];
10
      end
11
13
      y = vec_ytmp;
      x = vec_xtmp;
14
_{15} end
```

1.4.3 RungeKutta

Listing 3: Runge Kutta

```
_{1} function [x, y] = rk2(h, xend, f)
        ska_y_i = 1;
       vec_ytmp = [];
3
       vec_xtmp = [];
4
5
        vec_range = [0:h:xend];
       for ska_cur_x=vec_range
             k1 = f(ska\_cur\_x, ska\_y\_i);
             k2 = f(ska_cur_x + h, ska_y_i + h * k1);
             s\,k\,a\,_-y\,_-i \ = \ s\,k\,a\,_-y\,_-i \ + \ h/2 \ * \ (\,k1 \ + \ k2\,) \ ;
10
             vec\_xtmp \; = \; [\; vec\_xtmp \; , \quad ska\_cur\_x \; ] \; ;
11
             vec_ytmp = [vec_ytmp, ska_y_i];
12
       end
       y = vec_ytmp;
15
       x = vec_xtmp;
16
17 end
```

1.4.4 stiff

1.5 Ergebnisausdrucke

1.6 Interpretation der Ergebnisse

2 Van-der-Pol-DGL

2.1 Betrachtete Gleichung

$$y(0) = 0 (7)$$

$$\dot{y}(0) = 1 \tag{8}$$

$$\ddot{y} = 6 \cdot (1 - y^2) \cdot \dot{y} - y \tag{9}$$

2.2 Simulink/Analogrechner

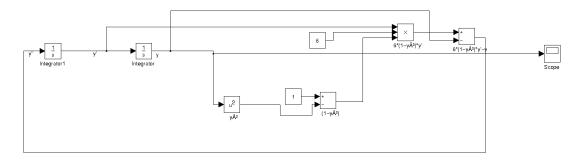


Abbildung 2: Simulink-Schaltbild: Van-der-Pol-DGL

2.3 Zu DGL 1 Ordnung transformierte DGL

$$y = z_2 \tag{10}$$

$$y' = z_1 \tag{11}$$

$$y'' = z'_1$$

$$z'_1 = 6 \cdot (1 - z_2^2) \cdot z_1 - z_2$$
(12)
(13)

$$z_2' = z_1 \tag{14}$$

(15)

$$\begin{pmatrix} z_1' \\ z_2' \end{pmatrix} \begin{pmatrix} 6 \cdot (1 - z_2^2) \cdot z_1 - z_2 \\ z_1 \end{pmatrix}$$

- 2.4 Iterationsgleichungen
- 2.4.1 Euler (expl)
- 2.4.2 Runge-Kutta 2
- 2.5 Ergebnisausdrucke
- 2.6 Interpretation der Ergebnisse
- 3 Lorenz-Attraktor mit RK 2
- 3.1 Simulink/Analogrechner
- 3.2 Iterationsgleichungen
- 3.2.1 Runge-Kutta 2
- 3.3 Matlab Programme
- 3.3.1 Lorenz
- 3.4 Ergebnisausdrucke
- 3.5 Interpretation der Ergebnisse