

Praktikum 1 : DGL

André Harms, Oliver Steenbuck

02.11.2011

Inhaltsverzeichnis

1	Steife Differentialgleichungen	2
1.1	Simulink/Analogrechner	2
1.2	Iterationsgleichungen	2
1.2.1	Euler (expl)	2
1.2.2	Euler (impl)	2
1.2.3	Runge-Kutta 2	2
1.3	Matlab Programme	2
1.3.1	Euler (expl)	2
1.3.2	Euler (impl)	3
1.3.3	RungeKutta	3
1.3.4	stiff	4
1.4	Ergebnisdrucke	4
1.5	Interpretation der Ergebnisse	4
2	Van-der-Pol-DGL	4
2.1	Simulink/Analogrechner	4
2.2	Zu DGL 1 Ordnung transformierte DGL	5
2.3	Iterationsgleichungen	5
2.3.1	Euler (expl)	5
2.3.2	Runge-Kutta 2	5
2.4	Ergebnisdrucke	5
2.5	Interpretation der Ergebnisse	5
3	Lorenz-Attraktor mit RK 2	5
3.1	Simulink/Analogrechner	5
3.2	Iterationsgleichungen	5
3.2.1	Runge-Kutta 2	5
3.3	Matlab Programme	5
3.3.1	Lorenz	5

3.4	Ergebnisdrucke	5
3.5	Interpretation der Ergebnisse	5

1 Steife Differentialgleichungen

1.1 Simulink/Analogrechner

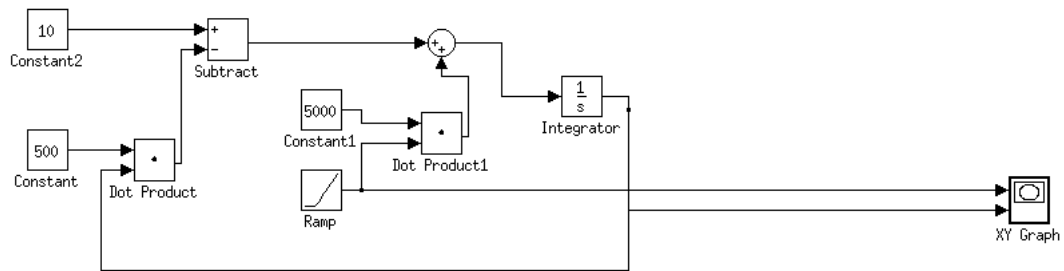


Abbildung 1: abbildung 1

1.2 Iterationsgleichungen

1.2.1 Euler (expl)

1.2.2 Euler (impl)

1.2.3 Runge-Kutta 2

1.3 Matlab Programme

1.3.1 Euler (expl)

Listing 1: Explizites Euler-Verfahren

```

1 function [x, y] = euler_expl(h, xend, f)
2     ska_y_i = 1;
3     vec_ytmp = [];
4     vec_xtmp = [];
5
6     vec_range = [0:h:xend];
7     for ska_cur_x=vec_range

```

```

8         ska_y_i = ska_y_i + h * f(ska_cur_x, ska_y_i);
9         vec_xtmp = [vec_xtmp, ska_cur_x];
10        vec_ytmp = [vec_ytmp, ska_y_i];
11    end
12
13    y = vec_ytmp;
14    x = vec_xtmp;
15 end

```

1.3.2 Euler (impl)

Listing 2: Implizites Euler-Verfahren

```

1 function [x, y] = euler_expl(h, xend)
2     ska_y_i = 1;
3     vec_ytmp = [];
4     vec_xtmp = [];
5
6     vec_range = [0:h:xend];
7     for ska_cur_x=vec_range
8         ska_y_i = ska_y_i + h * mtp0101(ska_cur_x, ska_y_i);
9         vec_xtmp = [vec_xtmp, ska_cur_x];
10        vec_ytmp = [vec_ytmp, ska_y_i];
11    end
12
13    y = vec_ytmp;
14    x = vec_xtmp;
15 end

```

1.3.3 RungeKutta

Listing 3: Runge Kutta

```

1 function [x, y] = rk2(h, xend, f)
2     ska_y_i = 1;
3     vec_ytmp = [];
4     vec_xtmp = [];
5
6     vec_range = [0:h:xend];
7     for ska_cur_x=vec_range
8         ska_y_i = ska_y_i + h * f(ska_cur_x + h / 2, ska_y_i
9             + h/2 * f(ska_cur_x, ska_y_i));
10        vec_xtmp = [vec_xtmp, ska_cur_x];
11        vec_ytmp = [vec_ytmp, ska_y_i];
12    end

```

```

13     y = vec_ytmp;
14     x = vec_xtmp;
15 end

```

1.3.4 stiff

1.4 Ergebnisausdrucke

1.5 Interpretation der Ergebnisse

2 Van-der-Pol-DGL

2.1 Simulink/Analogrechner

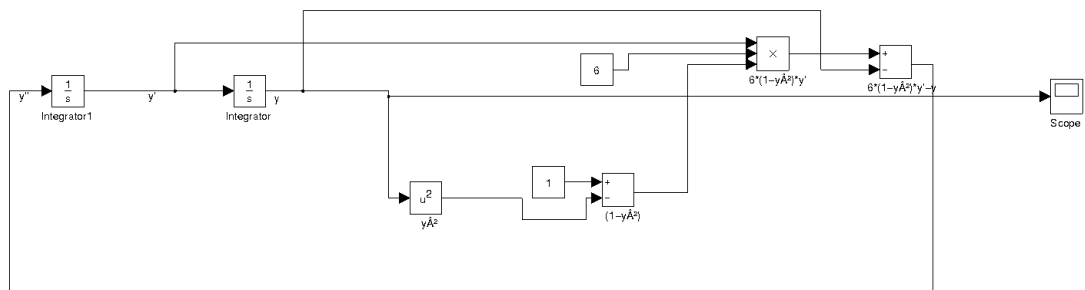


Abbildung 2: Simulink-Schaltbild Van-der-Pol-DGL

2.2 Zu DGL 1 Ordnung transformierte DGL

2.3 Iterationsgleichungen

2.3.1 Euler (expl)

2.3.2 Runge-Kutta 2

2.4 Ergebnisausdrucke

2.5 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