

Differentialgleichungen

numerische Lösung

```
In [1]: using DifferentialEquations, Plots;
```

Erstes Beispiel

Exponentieller Zerfall

```
In [2]: f(y,p,t)=p*y;
```

```
In [3]: deqprbl=ODEProblem(f,3.0,(0.0,12.0),-0.33)
```

```
Out[3]: ODEProblem with uType Float64 and tType Float64. In-place: false
Non-trivial mass matrix: false
timespan: (0.0, 12.0)
u0: 3.0
```

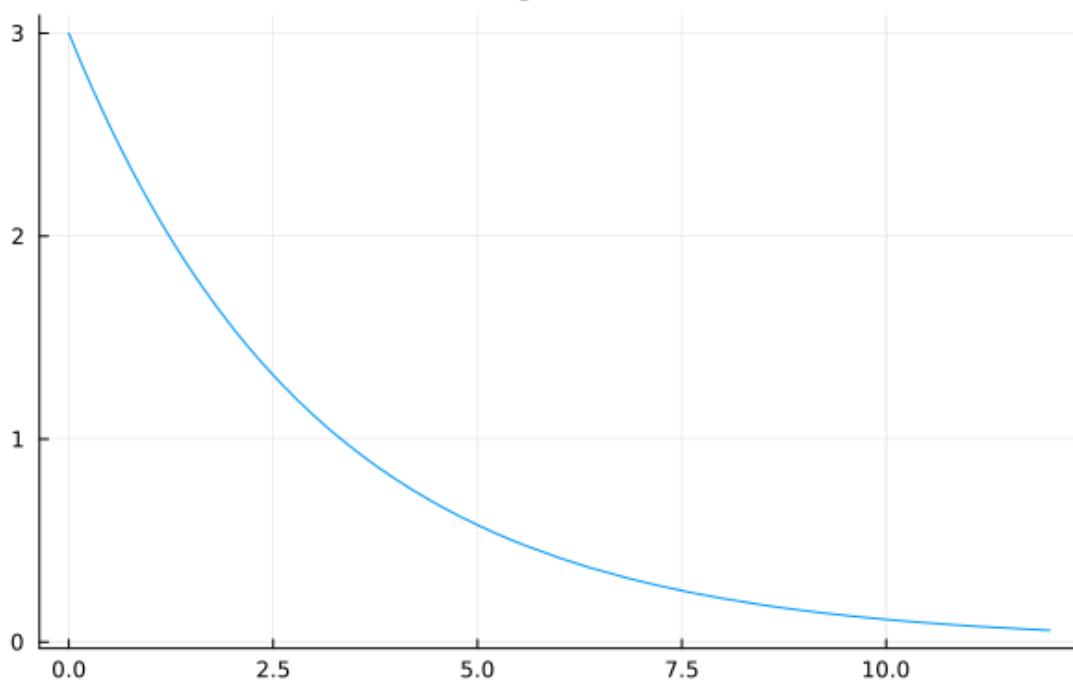
```
In [4]: solu=solve(deqprbl)
```

```
Out[4]: retcode: Success
Interpolation: 3rd order Hermite
t: 11-element Vector{Float64}:
 0.0
 0.12483206655122647
 0.6886030741856269
 1.5683314195545082
 2.5697752584258504
 3.814990854882284
 5.211084672862995
 6.787294336984937
 8.495975157356488
10.33295133653541
12.0
u: 11-element Vector{Float64}:
 3.0
 2.878927148046797
 2.390190997811257
 1.7879380049976432
 1.2847790129192596
 0.8518596122098523
 0.5373839046768774
 0.3194396051680836
 0.1817645018746746
 0.09913909588840228
 0.05719147224311148
```

```
In [5]: plot(t->solu(t),0,12,legend=false,title="Lösung des AWP")
```

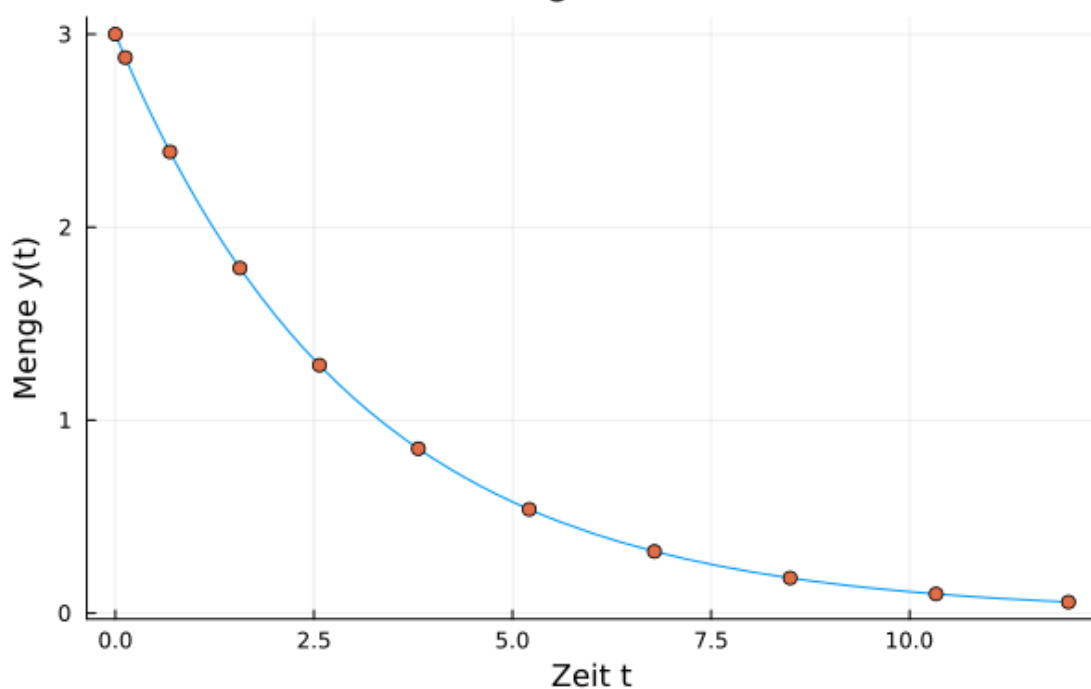
Out[5]:

Lösung des AWP

In [6]: `scatter!(solu.t,solu.u,xlabel="Zeit t",ylabel="Menge y(t)")`

Out[6]:

Lösung des AWP



Zweites Beispiel

System von 2 ODEs (Pendel)

```
In [7]: function f!(yp,y,p,t)
          g=p[1]; l=p[2];
          yp[1]=y[2];
          yp[2]=-g*sin(y[1])/l;
          return;
        end;
```

```
y0=[0.0,6.26]; tspan=(0.0,12.0); p=(9.81,1.0);  
pendprob=ODEProblem(f!,y0,tspan,p)
```

Out[7]: `ODEProblem` with uType `Vector{Float64}` and tType `Float64`. In-place: `true`
Non-trivial mass matrix: `false`
timespan: (0.0, 12.0)
u0: 2-element Vector{Float64}:
 0.0
 6.26

In [8]: `pendsol=solve(pendprob,reltol=1.0e-4)`

```

Out[8]: retcode: Success
Interpolation: 3rd order Hermite
t: 77-element Vector{Float64}:
 0.0
 0.0015948963317384368
 0.013261602775004096
 0.039563636350133734
 0.07848460760072484
 0.12885739197020463
 0.19378324959991539
 0.27140427112112103
 0.3656724096299395
 0.48660050305089914
 0.6133868267665499
 0.7455284551837222
 0.8857316437369804
  :
10.089214893133336
10.262092120572163
10.437589645376253
10.605917510186504
10.759790038210822
10.938660108296784
11.119812229409316
11.30889283554939
11.496590610764633
11.679028259801468
11.855367568005446
12.0
u: 77-element Vector{Vector{Float64}}:
 [0.0, 6.26]
 [0.009984009513870205, 6.259921895914342]
 [0.08299377212857774, 6.254603730701196]
 [0.24703694195328355, 6.212242841521462]
 [0.4864388506606143, 6.075503550012873]
 [0.7855992239020742, 5.782581208969148]
 [1.1447980337661436, 5.260719868236802]
 [1.5249775777062335, 4.523963760043152]
 [1.908212614037087, 3.6155766546696206]
 [2.281822321400845, 2.600647274515747]
 [2.5573939436173387, 1.7892730215689738]
 [2.7518637404071216, 1.1911623804865483]
 [2.886759129333568, 0.7623826770073654]
  :
 [-2.9904964516139154, -0.41690103299178743]
 [-3.043503705216938, -0.21113916763746918]
 [-3.067274804300176, -0.06652721531394024]
 [-3.0684379397922297, 0.05238769543703016]
 [-3.0514088524981724, 0.1732058901599721]
 [-3.0042845460704464, 0.36732916643025837]
 [-2.9115387784321967, 0.6835012302989832]
 [-2.7337762346979493, 1.2487272762977224]
 [-2.416951293256817, 2.2090788423968113]
 [-1.8891317324576662, 3.6646369049045977]
 [-1.093326536600427, 5.346665880770893]
 [-0.2462563390906526, 6.212793835761673]

```

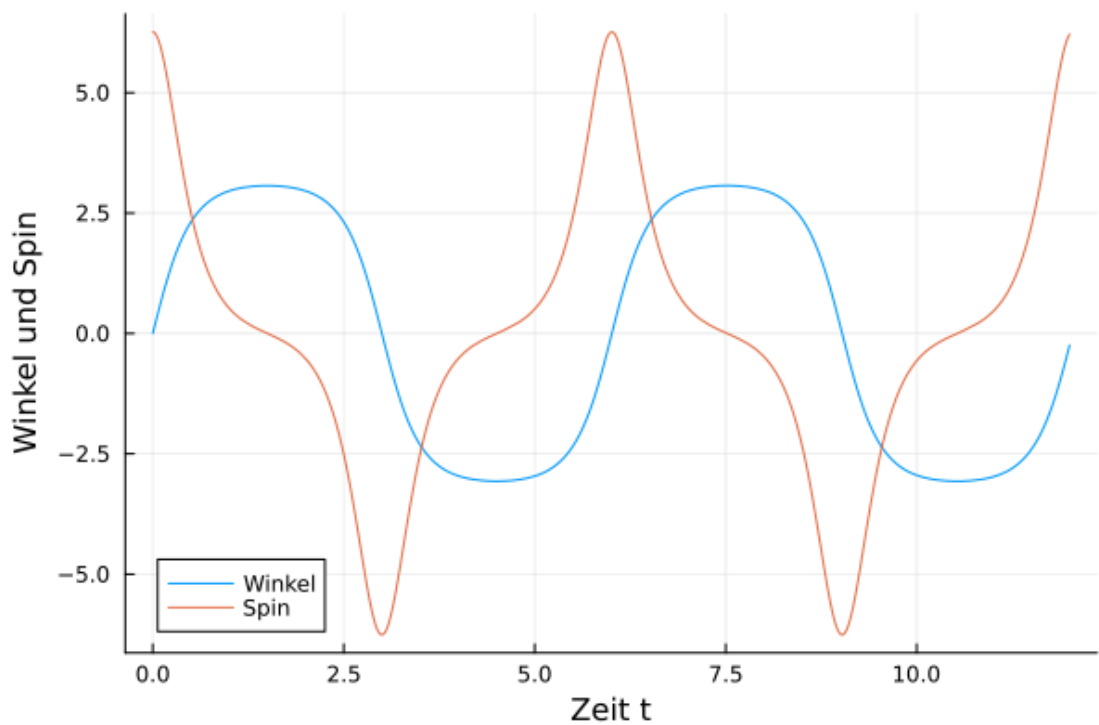
```

In [9]: plot(t->pendsol(t)[1],0,12,label="Winkel",
           xlabel="Zeit t")

```

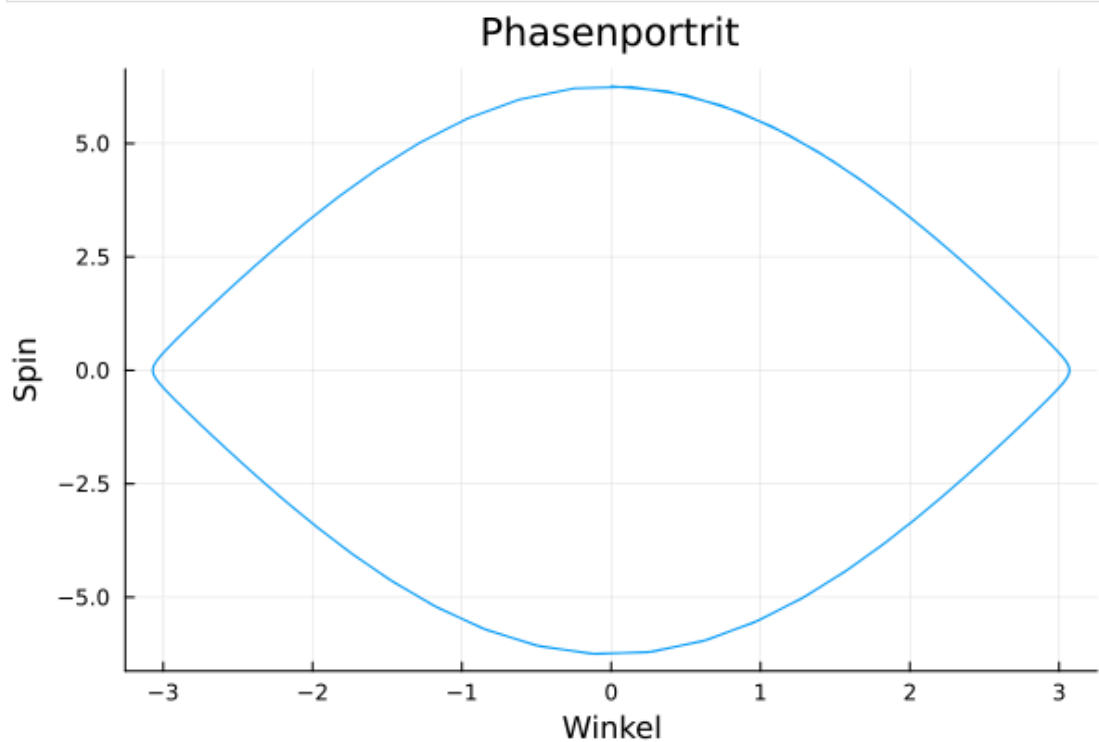
```
plot!(t->pendsol(t)[2],0,12,label="Spin",
      ylabel="Winkel und Spin")
```

Out[9]:



```
In [10]: phpl=plot(t->pendsol(t)[1],t->pendsol(t)[2],0,12,leg=false,
              title="Phasenportrit",xlabel="Winkel",ylabel="Spin")
```

Out[10]:



In []: