

# notebook

October 12, 2022

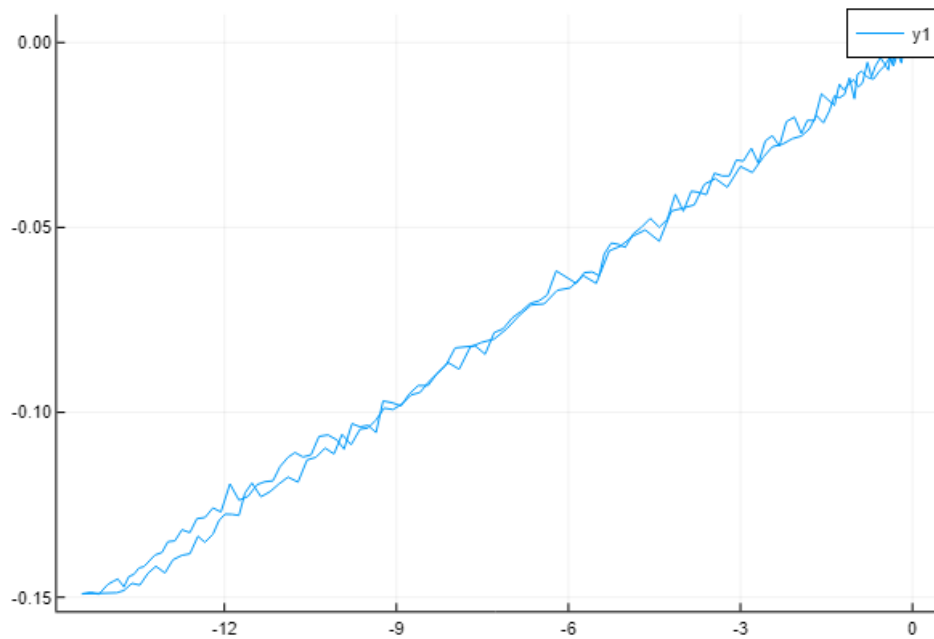
```
[ ]: using DataFrames, Statistics, CSV
      using Measurements, Unitful
      using Plots, PlotThemes
      using EasyFit, Peaks, Symbolics, Latexify
      plotlyjs()
```

```
Plots.PlotlyJSBackend()
```

```
[ ]: data = CSV.read("data.csv", DataFrame)

      data = data[data[:,1] .> -0.15,:]

      plot(data[:,2],data[:,1])
```



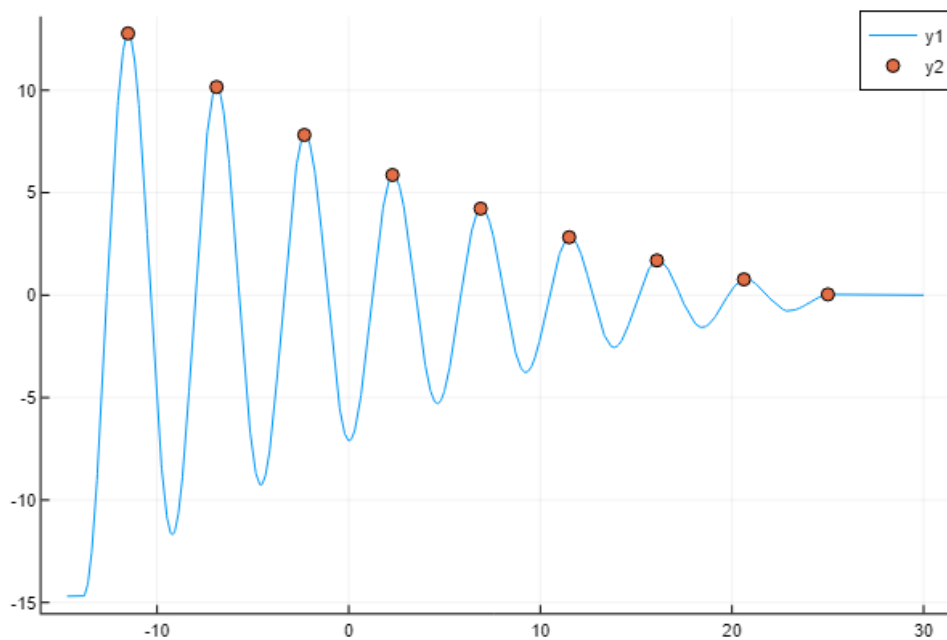
```
[ ]: data2 = CSV.read("data2.csv", DataFrame)

data2 = data2 .- data2[end,2]

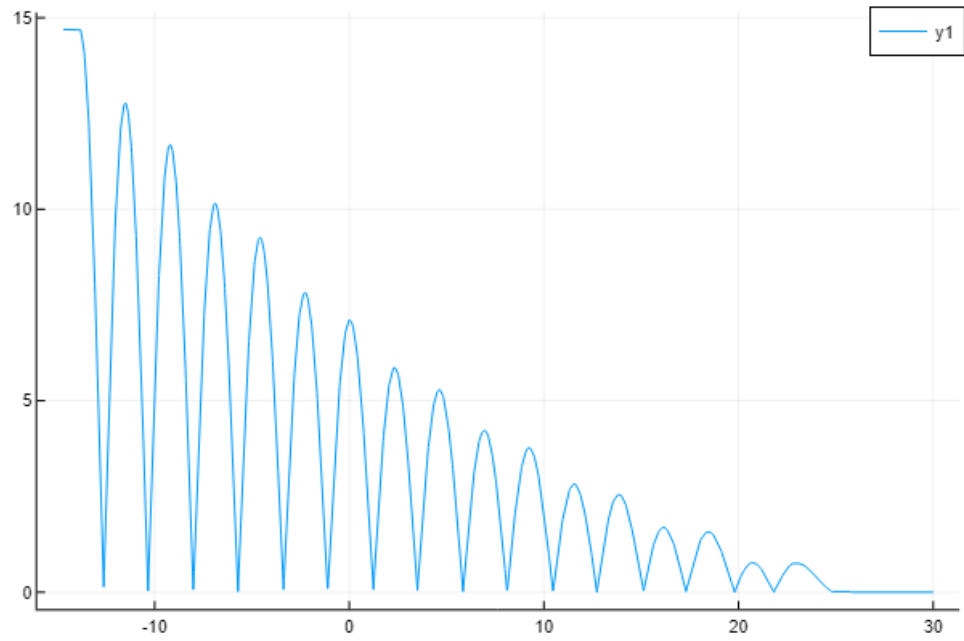
plot(data2[:,1], data2[:,2])

peaks = findmaxima(data2[:,2])
peaks = peaks[1][peaks[2] .> -10]

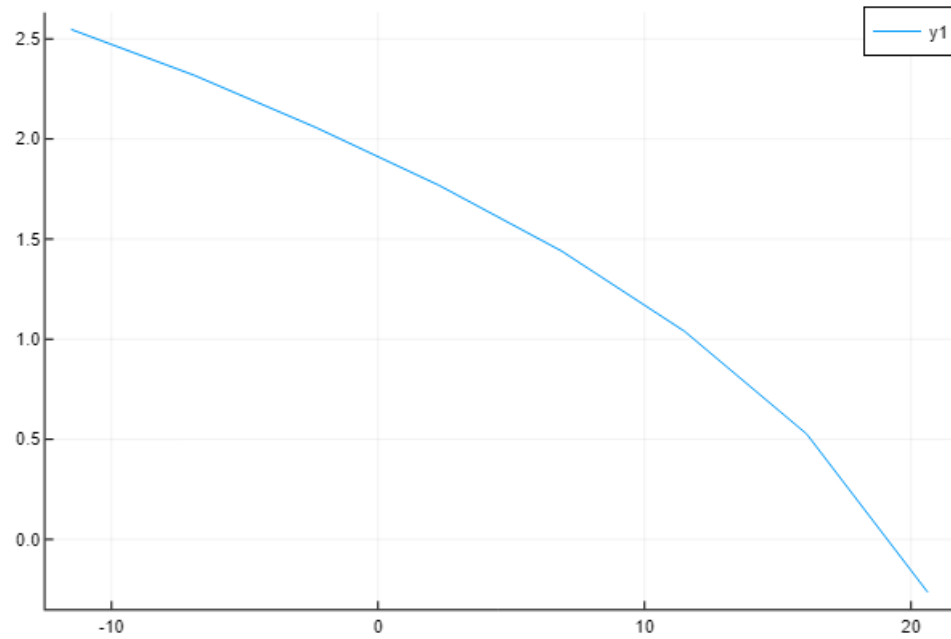
scatter!(data2[peaks,1], data2[peaks,2])
```



```
[ ]: plot(data2[:,1], abs.(data2[:,2]))
```

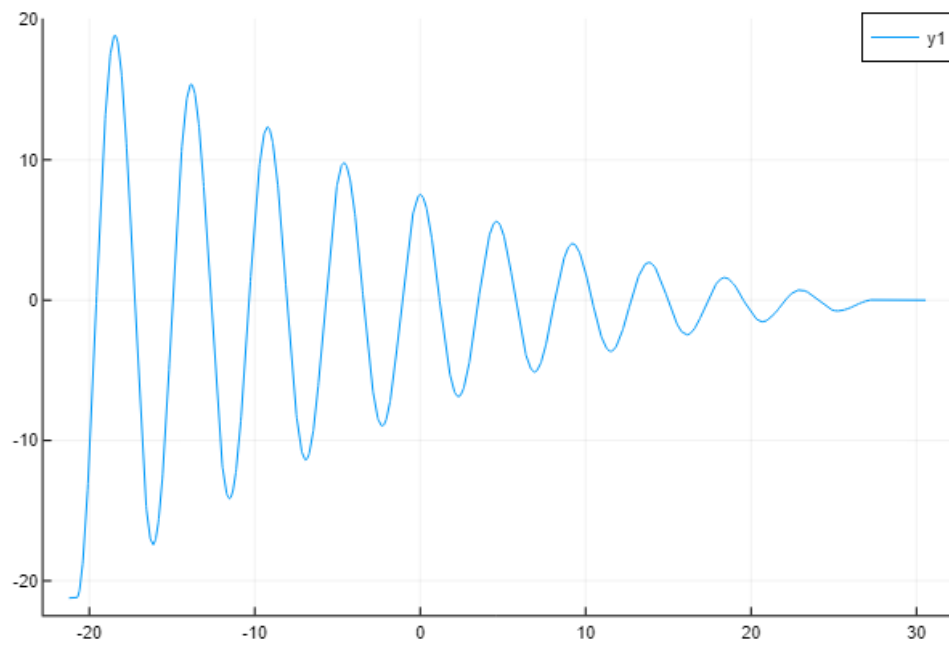


```
[ ]: plot(data2[peaks,1][1:end-1], log.(data2[peaks,2][1:end-1]))
```



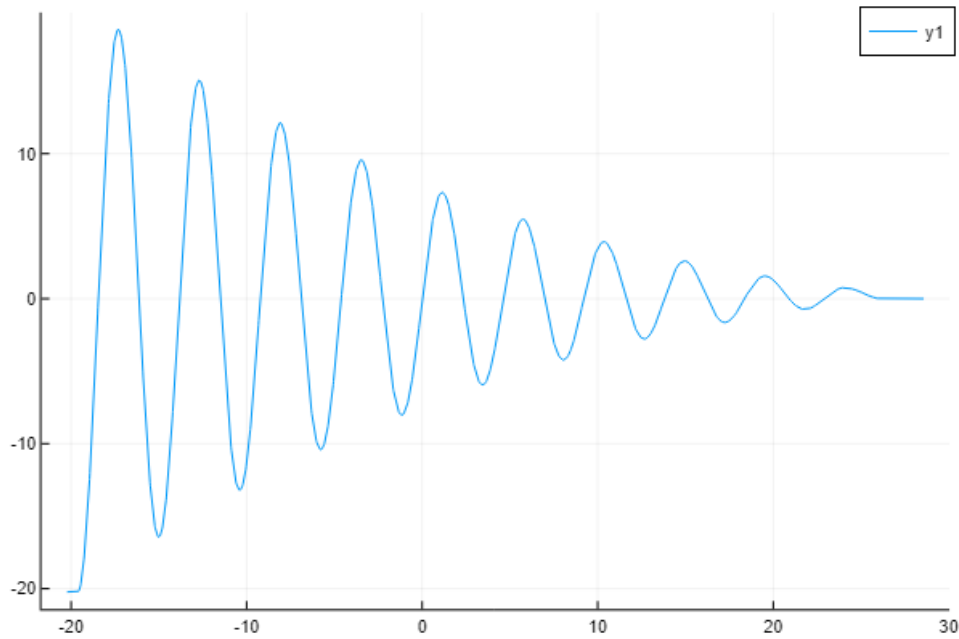
```
[ ]: data3 = CSV.read("data3.csv", DataFrame)
      data3 = data3 .- data3[end,2]

      plot(data3[:,1], data3[:,2])
```



```
[ ]: data4 = CSV.read("data4.csv", DataFrame)
      data4 = data4 .- data4[end,2]

      plot(data4[:,1], data4[:,2])
```



```
[ ]: function findErrorFromSym(symExpr; errorPrefix = "Err")
    vars = Symbolics.get_variables(symExpr)
    varErrs = []

    for i in vars
        push!(varErrs, Symbolics.variable(string(errorPrefix,i)))
    end

    Dvars = [expand_derivatives(Differential(i)(symExpr)) for i in vars]

    symErr = sqrt(sum((Dvars[i]*varErrs[i])^2 for i in eachindex(vars)))

    return symErr
end
```

```
findErrorFromSym (generic function with 1 method)
```

```
[ ]: k = fitlinear(data[:,2],data[:,1]).a*1u"N*m"
```

```
0.010578748137015617 m N
```

```
[ ]: rskvfull = (5.16±0.01)u"cm"
      rskvinn = (0.27±0.01)u"cm"
      r = rskvfull-rskvinn
```

4.89 ± 0.014 cm

```
[ ]: r = (9.5)
      Δr = 0.1
      m = (122)
      Δm = 1
      k = fitlinear(data2[peaks,1],log.(data2[peaks,2]))
```

----- Linear Fit -----

Equation:  $y = ax + b$

With:  $a = -0.12751728726254244$   
 $b = 1.7714598834466944$

Pearson correlation coefficient,  $R = 0.874955589289366$   
Average square residue = 0.695718033797647

Predicted Y:  $y_{pred} = [3.2399489635621332, 2.650819096409187...$   
residues =  $[0.6924588274723651, 0.33265597345182263...$

-----

```
[ ]: latexify(: (Δk=sqrt(n/(n*sum(x.^2)-sum(x.^2))))),env = :eq)
```

$$\Delta k = \sqrt{\frac{n}{n \cdot \sum x^2 - \sum x^2}} \quad (1)$$

```
[ ]: x = data2[peaks,1]
      n = length(x)
      Δk = sqrt(n/(n*sum(x.^2)-sum(x.^2)))
```

0.025879541985573656

```
[ ]: @variables R M Δ ΔR ΔM
      b = -1/2 *M*R^2*
```

$$-0.5R^2M\kappa \quad (2)$$

```
[ ]: Δb = findErrorFromSym(b,errorPrefix = "Δ")
```

$$\sqrt{0.25\Delta\kappa^2R^4M^2 + 0.25\kappa^2\Delta M^2R^4 + \kappa^2\Delta R^2R^2M^2} \quad (3)$$

```
[ ]: bfun = eval(build_function(b, M, ,R))
Δbfun = eval(build_function(Δb, M,R,Δ ,ΔM, ,ΔR))
b = substitute(b, Dict([ => k.a, R => r, M => m]))
b = bfun(m, k.a, r);
@show b;
```

```
b = 702.0145457021117
```

```
[ ]: Δb = Δbfun(m,r, Δk, Δm,k.a, Δr);
b = substitute(Δb, Dict([ => k.a, R => r, M => m, ΔM => Δm, Δ => Δk, ΔR =>
↳Δr]))
@show Δb;
```

```
Δb = 143.35338291864204
```