```
using DataFrames, JLD2, FileIO, Plots, DataFramesMeta
```

18

```
begin
    # Plots default fonts and font sizes, could add colors as well
    leg_size = 12
    tick_size = 14
    label_size = 18
end
```

```
@recipe function f(::Type{Val{:samplemarkers}}, x, y, z; step = 10)
    n = length(y)
    sx, sy = x[1:step:n], y[1:step:n]
    # add an empty series with the correct type for legend markers
    @series begin
        seriestype := :path
        markershape --> :auto
        x := [Inf]
        y := [Inf]
    end
    # add a series for the line
    @series begin
        primary := false # no legend entry
        markershape := :none # ensure no markers
        seriestype := :path
        seriescolor := get(plotattributes, :seriescolor, :auto)
        x := x
        y := y
    end
    # return  a series for the sampled markers
    primary := false
    seriestype := :scatter
    markershape --> :auto
    x := sx
    y := sy
end
```

"E:\\JuliaStuff\\Notebooks"

```
pwd()
```

Main.workspace2.plot_lastframe

```
"""
    plot_lastframe()

Evaluates the height field at the last time step and returns the height, image and
data.
"""
function plot_lastframe(; λ=8, T=20000)
    df =
load("..\\Swalbe\\data\\Moving_wettability\\height_direc_diagonal_sp_0_sine_$(λ)_tmax_
$(T)_v2.jld2") |> DataFrame
    LL, t = size(df)
    l = Int(sqrt(LL))
    h = reshape(df[!, Symbol("h_$T")], l, l)

    p = heatmap(h, aspect_ratio=1, c=:viridis)

    return p,h,df
end
```

Main.workspace2.find_tr

```
"""
    find_tr()

Marks the first occurence of a rupture of the film.
"""
function find_tr(; λ=8, T=20000, δ=100, vel=0)
    df =
load("..\\Swalbe\\data\\Moving_wettability\\height_direc_diagonal_sp_$(vel)_sine_$(λ)_
tmax_$(T)_v2.jld2") |> DataFrame
    Tr = 0
    for i in δ:δ:T
        h = df[!, Symbol("h_$i")]
        hm = minimum(h)
        if hm ≤ 0.05
            Tr = i
            break
        end
    end
    return Tr
end
```

t_0 (generic function with 1 method)

```
function t_0(;hᵦ=0.07, γ=0.01, μ=1/6, θ=1/6)
```

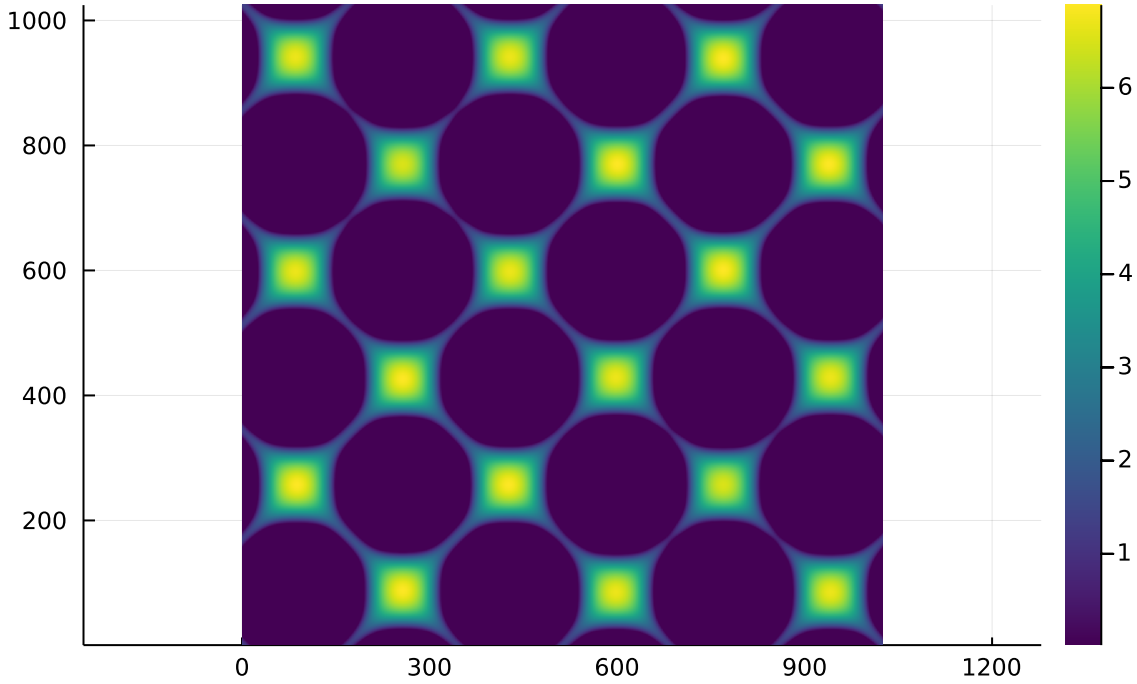
```
    qsq = hᵦ * (1 - cospi(θ)) * (2 - 3 * hᵦ)
    charT = 3 * μ / (γ * qsq^2)

    return charT, qsq
end
```

df =

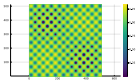
	h_1000	h_10000	h_100000	h_101000	h_102000	h_103000	h_104000	h_10
1	1.0	1.00001	0.996098	0.995998	0.995896	0.995793	0.995689	0.99
2	1.0	1.00001	0.996112	0.996012	0.995911	0.995808	0.995704	0.99
3	1.0	1.00001	0.996155	0.996056	0.995956	0.995853	0.99575	0.99
4	1.0	1.00002	0.996227	0.996129	0.996029	0.995928	0.995826	0.99
5	1.0	1.00002	0.996326	0.99623	0.996132	0.996033	0.995932	0.99
6	1.00001	1.00003	0.996455	0.996361	0.996265	0.996168	0.996069	0.99
7	1.00001	1.00004	0.996611	0.996519	0.996426	0.996332	0.996235	0.99
8	1.00001	1.00005	0.996796	0.996707	0.996616	0.996525	0.996431	0.99
9	1.00001	1.00006	0.997007	0.996922	0.996835	0.996747	0.996657	0.99
10	1.00001	1.00008	0.997247	0.997165	0.997082	0.996997	0.996912	0.99

```
df = load("..\\Swalbe\\data\\Moving_wettability\\height_direc_diagonal_sp_0_sine_3_tmax_500000_v2.jld2") |> DataFrame
```




```
heatmap(reshape(df[!, :h_500000], 1024, 1024), aspect_ratio=1, c=:viridis)
```

```
hm =
```



```
512×512 Matrix{Float64}:
 1.07585  1.07541  1.07408  1.07194   …  1.06906   1.07194   1.07408   1.0754
 1.07541  1.07113  1.06597  1.06002      1.08341   1.08271   1.08119   1.0787
 1.07408  1.06597  1.05693  1.04705      1.09639   1.09231   1.08726   1.0811
 1.07194  1.06002  1.04705  1.03317      1.10799   1.10073   1.09231   1.0827
 1.06906  1.05339  1.03651  1.01855      1.11823   1.10799   1.09639   1.0834
 1.06558  1.04625  1.02548  1.00343   …  1.12716   1.11415   1.09956   1.0833
 1.06165  1.0388   1.01423  0.988111     1.13482   1.11929   1.10192   1.0827
 ⋮                                   ⋱                     ⋮
 1.06165  1.08271  1.10192  1.11929      0.960659  0.988111  1.01423   1.0388
 1.06558  1.08337  1.09956  1.11415      0.980309  1.00343   1.02548   1.0462
 1.06906  1.08341  1.09639  1.10799      0.999721  1.01855   1.03651   1.0533
 1.07194  1.08271  1.09231  1.10073      1.01855   1.03317   1.04705   1.0600
 1.07408  1.08119  1.08726  1.09231   …  1.03651   1.04705   1.05693   1.0659
 1.07541  1.07876  1.08119  1.08271      1.05339   1.06002   1.06597   1.0711
```

```
hm = plot_lastframe(λ=9, T=15000)
```

```
begin
    ET = [500000, 80000, 30000, 20000, 20000, 20000, 15000]
    lam = [3, 4, 5, 6, 7, 8, 9]
    delta_t = [1000, 100, 100, 100, 100, 100, 100]
    data = zeros(7, 3)
    data[:, 1] = ET
    data[:, 2] = lam
    for i in 1:7
        fir = find_tr(λ=lam[i], T=ET[i], δ=delta_t[i])
        data[i, 3] = fir
    end
end
```

```
7×3 Matrix{Float64}:
 500000.0  3.0  105000.0
  80000.0  4.0   25000.0
  30000.0  5.0   19500.0
```

```
20000.0  6.0   15800.0
20000.0  7.0   13800.0
20000.0  8.0   12300.0
15000.0  9.0   11700.0
```

```
data
```

▸ [0.591788, 0.140902, 0.109904, 0.08905, 0.0777779, 0.0693238, 0.0659421]

```
begin
    time_by_t0 = zeros(7)
    time_by_t0 = data[:, 3] ./ t_0()[1]
end
```

▸ [1.69082, 0.591788, 0.394526, 0.225443, 0.140902, 0.109904, 0.08905, 0.0777779, 0.0(

```
begin
    all_time = zeros(10,2)
    t0 = t_0()[1]
    all_time[:, 1] = [512, 342, 256, 171, 128, 102, 85, 73, 64, 56]
    all_time[:, 2] = [1.69082, data[1, 3]/t0, 0.394526, 0.225443, data[2,
3]/t0,data[3, 3]/t0,data[4, 3]/t0,data[5, 3]/t0,data[6, 3]/t0, data[7, 3]/t0]
end
```

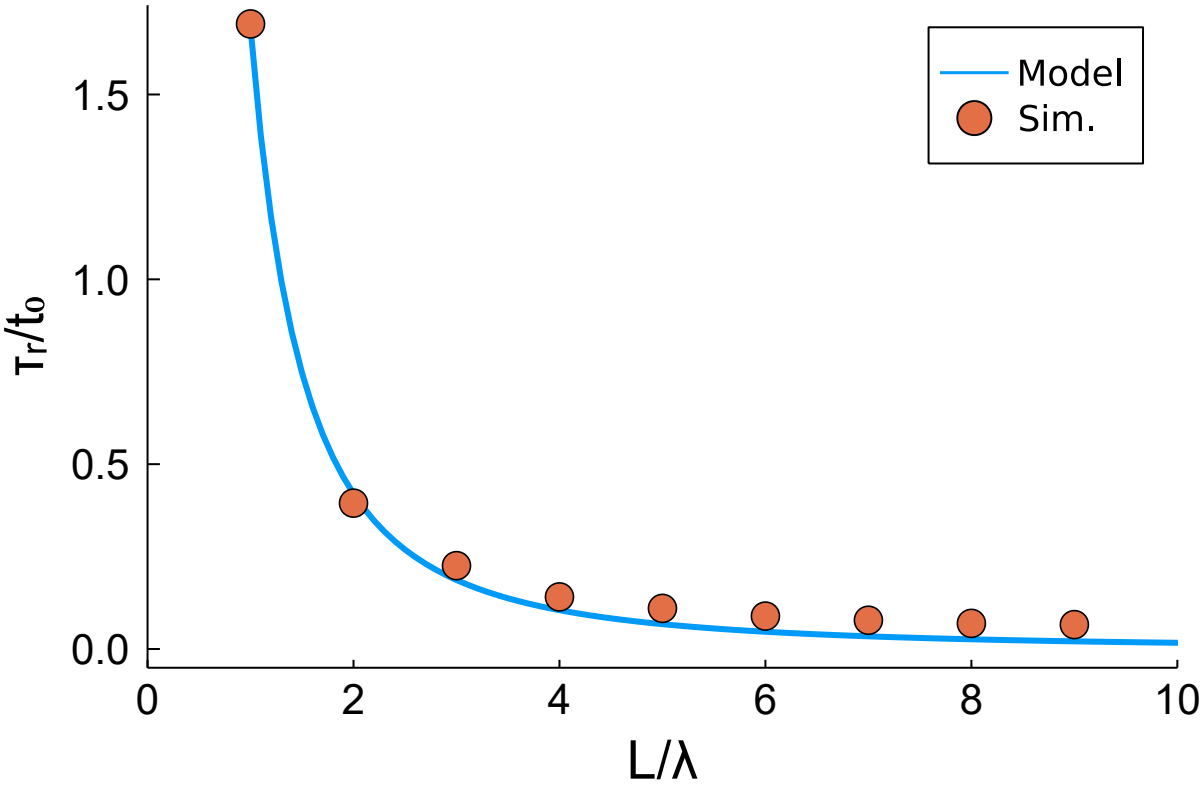
model_tr (generic function with 1 method)

```
function model_tr(;λ=512, α=-1.8)
    model = []
    q0 = sqrt(t_0()[2])
    l0 = 2π/q0
    res = 0.0
    for i in 1:0.1:512
        res = λ / (l0 * 2π * i^α)
        push!(model, res)
    end
    return model
end
```

model_tr2 (generic function with 1 method)

```
function model_tr2(;λ=512, α=-1.8, v=0.1)
    model = []
    q0 = sqrt(t_0()[2])
    l0 = 2π/q0
    res = 0.0
```

```
    for i in 1:0.1:512
        res = λ / (l0 * 2π * i^α)
        push!(model, res)
    end
    return model
end
```




```
begin
    my_mod = model_tr(λ=512, α=2)
    p = plot(collect(1:0.1:512),
         my_mod,                          # y-axis
         xlim=(-0.0, 10),
         label="Model",                       # labels
         xlabel="L/λ",                        # x-axis label
         ylabel="τᵣ/t₀",                      # y-axis label
         w = 3,                               # line width
         legendfontsize = leg_size,           # legend font size
         tickfont = (tick_size, "Arial"),     # tick font and size
         guidefont = (label_size, "Arial"), # label font and size
```

```
        grid = :none,                    # grid variable
        legend=:topright)                # legend position

    scatter!([1,1000,2,3,4,5,6,7,8,9], all_time[:, 2], ms=8, label="Sim.")
end
```

```
savefig(p, "..\\Figures\\Model_rt.pdf")
```

```
3×6 Matrix{Float64}:
 19603.2    24504.1    29404.9    34305.7    39206.5    44107.3
  1960.32    2450.41    2940.49    3430.57    3920.65    4410.73
   196.032    245.041    294.049    343.057    392.065    441.073
```

```
begin
    results = zeros(3,6)
    for i in enumerate([0.1, 1.0, 10.0])
        for j in enumerate(4:9)
            results[i[1],j[1]] = sqrt(2)t_0()[1]/((512/j[2])*i[2])
        end
    end
    results
end
```

```
▸ [0.1, 1.0, 10.0, 0.1, 1.0, 10.0, 0.1, 1.0, 10.0, 0.1, 1.0, 10.0, 0.1, 1.0, 10.0
```

```
begin
    vel_norm = [0.1, 1, 10]
    v4 = [19603, 1960, 196]
    v5 = [24504, 2450, 245]
    v6 = [29404, 2940, 294]
    v7 = [34305, 3430, 343]
    v8 = [39206, 3920, 392]
    v9 = [44107, 4410, 441]

    all_vs = zeros(3, 7)
    all_vs[:, 1] = vel_norm
    all_vs[:, 2] = v4
    all_vs[:, 3] = v5
    all_vs[:, 4] = v6
    all_vs[:, 5] = v7
    all_vs[:, 6] = v8
    all_vs[:, 7] = v9

    rt_data = DataFrame()
    rts = []
```

```
    lams_w = []
    vels = []

    for i in enumerate(4:9)
        for v in enumerate(all_vs[:,i[1]+1])
            fir = find_tr(λ=i[2], T=75000, δ=500, vel=Int(v[2]))
            push!(rts, fir/t_0()[1])
            push!(lams_w, i[2])
            push!(vels, all_vs[v[1],1])
        end
    end
    rt_data[!, "Rupture_time"] = rts
    rt_data[!, "Wavelengths"] = lams_w
    rt_data[!, "Velocities"] = vels
end
```

	Rupture_time	Wavelengths
1	0.28744	4
2	0.225443	5
3	0.1719	6
4	0.160628	7
5	0.152174	8
6	0.154992	9

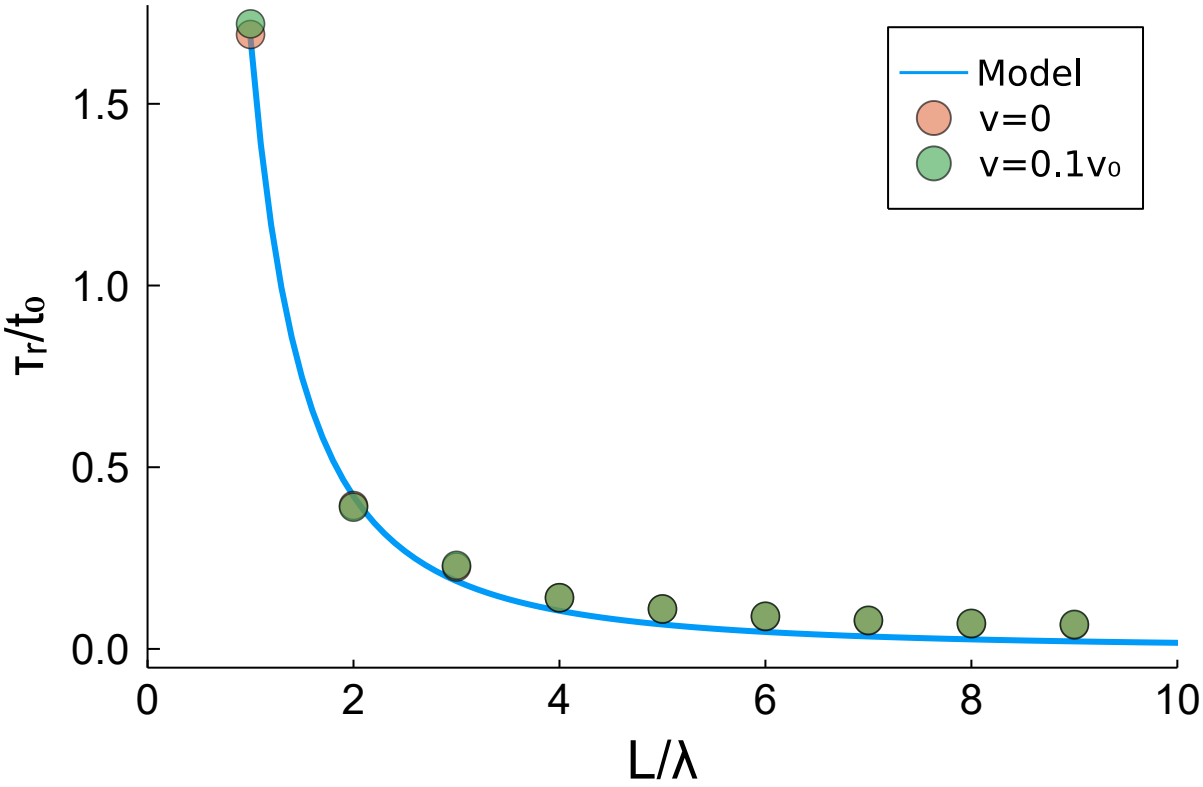
```
begin
    rupture_v01 = @linq rt_data |>
    where(:Velocities .== 0.1) |>
    select(:Rupture_time, :Wavelengths)

    rupture_v1 = @linq rt_data |>
    where(:Velocities .== 1) |>
    select(:Rupture_time, :Wavelengths)

    rupture_v10 = @linq rt_data |>
    where(:Velocities .== 10) |>
    select(:Rupture_time, :Wavelengths)

```

```
end
```




```
begin
    all_v01 = zeros(9)
    all_v01[1:3] = [1.72, 0.39, 0.23]
    all_v01[4:end] = rupture_v01[!, :Rupture_time]

    all_v1 = zeros(9)
    all_v1[1:3] = [2.34, 0.59, 0.25]
    all_v1[4:end] = rupture_v1[!, :Rupture_time]

    all_v10 = zeros(9)
    all_v10[1:3] = [2.25, 0.68, 0.45]
    all_v10[4:end] = rupture_v10[!, :Rupture_time]

    p_now = plot(collect(1:0.1:512),
         my_mod,                          # y-axis
```

```
        xlim=(-0.0, 10),
        label="Model",                    # labels
        xlabel="L/λ",                     # x-axis label
        ylabel="τᵣ/t₀",                   # y-axis label
        w = 3,                            # line width
        legendfontsize = leg_size,        # legend font size
        tickfont = (tick_size, "Arial"),  # tick font and size
        guidefont = (label_size, "Arial"), # label font and size
        grid = :none,                     # grid variable
        legend=:topright)                 # legend position
    scatter!([1,1000,2,3,4,5,6,7,8,9], all_time[:, 2], ms=8, alpha=0.6, label="v=0")
    scatter!([1,2,3,4,5,6,7,8,9], all_v01, ms=8, alpha=0.6, label="v=0.1v₀")
    # scatter!([1,2,3,4,5,6,7,8,9], all_v1, ms=8, alpha=0.6, label="v=1v₀")
    # scatter!([1,2,3,4,5,6,7,8,9], all_v1, ms=8, alpha=0.6, xlim=(0, 10),
label="v=10v₀")

end
```