Finding Maxima and Minima of DiffEq Solutions

Chris Rackauckas

August 7, 2021

0.0.1 Setup

In this tutorial we will show how to use Optim.jl to find the maxima and minima of solutions. Let's take a look at the double pendulum:

```
#Constants and setup
using OrdinaryDiffEq
initial = [0.01, 0.01, 0.01, 0.01]
tspan = (0.,100.)

#Define the problem
function double_pendulum_hamiltonian(udot,u,p,t)
    α  = u[1]
    lα = u[2]
    β  = u[3]
    lβ = u[4]
    udot .=
    [2(lα-(1+cos(β))lβ)/(3-cos(2β)),
    -2sin(α) - sin(α+β),
    2(-(1+cos(β))lα + (3+2cos(β))lβ)/(3-cos(2β)),
    -sin(α+β) - 2sin(β)*(((lα-lβ)lβ)/(3-cos(2β))) + 2sin(2β)*((lα^2 - 2(1+cos(β))lα*lβ
+ (3+2cos(β))lβ^2)/(3-cos(2β))^2)]
end

#Pass to solvers
poincare = ODEProblem(double_pendulum_hamiltonian, initial, tspan)
```

```
ODEProblem with uType Vector{Float64} and tType Float64. In-place: true
timespan: (0.0, 100.0)
u0: 4-element Vector{Float64}:
 0.01
 0.01
 0.01
 0.01
```

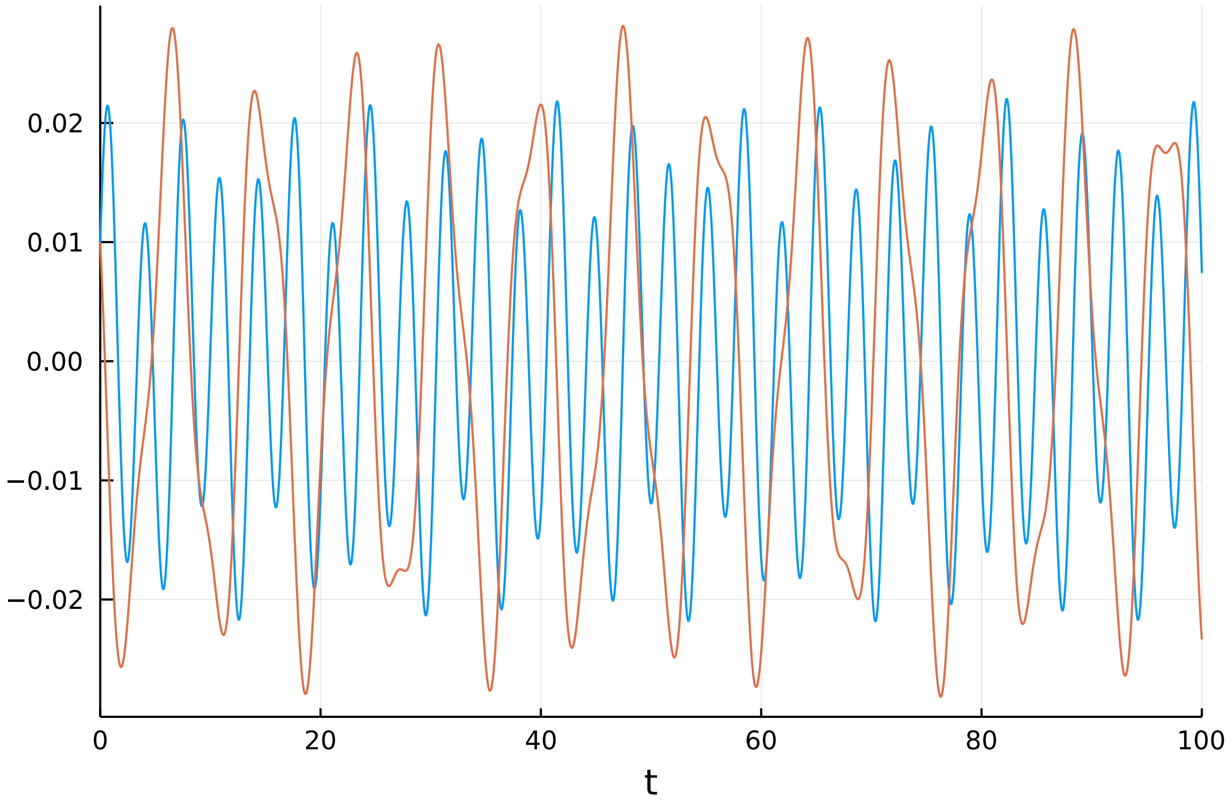
```
sol = solve(poincare, Tsit5())
```

```
retcode: Success
Interpolation: specialized 4th order "free" interpolation
t: 193-element Vector{Float64}:
   0.0
   0.08332584852065579
   0.24175300587841853
   0.4389533535703127
   0.6797301355043014
```

```
  0.9647629621490508
  1.3179425637594349
  1.7031226016307728
  2.0678503967116617
  2.4717899847517866
  ⋮
 95.8457309586563
 96.3577910122243
 96.92913461915474
 97.44679415429573
 97.96248479179103
 98.51183391850897
 99.0608253308051
 99.58284388126884
100.0
u: 193-element Vector{Vector{Float64}}:
 [0.01, 0.01, 0.01, 0.01]
 [0.009170687380405334, 0.006669000455384281, 0.012420525490765841, 0.00826
6408515192909]
 [0.007673275265972504, 0.00037461737897660443, 0.016442590227730397, 0.004
636827483318277]
 [0.006125974419239289, -0.007305450189721187, 0.019967371084231897, -0.000
3364979830896869]
 [0.004966110662711131, -0.01630851653373806, 0.021440659476204722, -0.0067
05037098400474]
 [0.0047955683310194714, -0.026238103489235838, 0.01882432520883759, -0.013
913364556753736]
 [0.0060546798253553686, -0.03712455187908053, 0.010055702788069564, -0.021
038127478647375]
 [0.007900784412908646, -0.04667606960847394, -0.002673581831574513, -0.025
18303627203377]
 [0.008276510489473166, -0.05278433365633976, -0.012731546444725367, -0.025
25804037623962]
 [0.00552349681674124, -0.05525250414492613, -0.016843881882621835, -0.0218
98963191274153]
 ⋮
 [-0.014886751154788403, 0.04233275827248491, 0.0136282832580092, 0.0180290
82291419467]
 [-0.008190258536393156, 0.054422679804409874, 0.009448013826704854, 0.0177
4006800908217]
 [0.004124711787695587, 0.05674878820505975, -0.00515418739191979, 0.017596
983103942972]
 [0.013079718118471138, 0.048077043077395416, -0.01377066122508919, 0.01828
6648610391296]
 [0.015316040241448815, 0.03163095955755212, -0.008956991644884404, 0.01711
8404049844594]
 [0.011115490017375213, 0.00992901822063005, 0.007297481421219374, 0.010353
371812537674]
 [0.005713878919291721, -0.011787427051187821, 0.02050806401368854, -0.0023
10458905852316]
 [0.004211439726126673, -0.029911199361470703, 0.018750446422905413, -0.015
650712294907165]
 [0.005741239607321043, -0.04165385985159563, 0.007413270184094278, -0.0233
4897852528026]
```

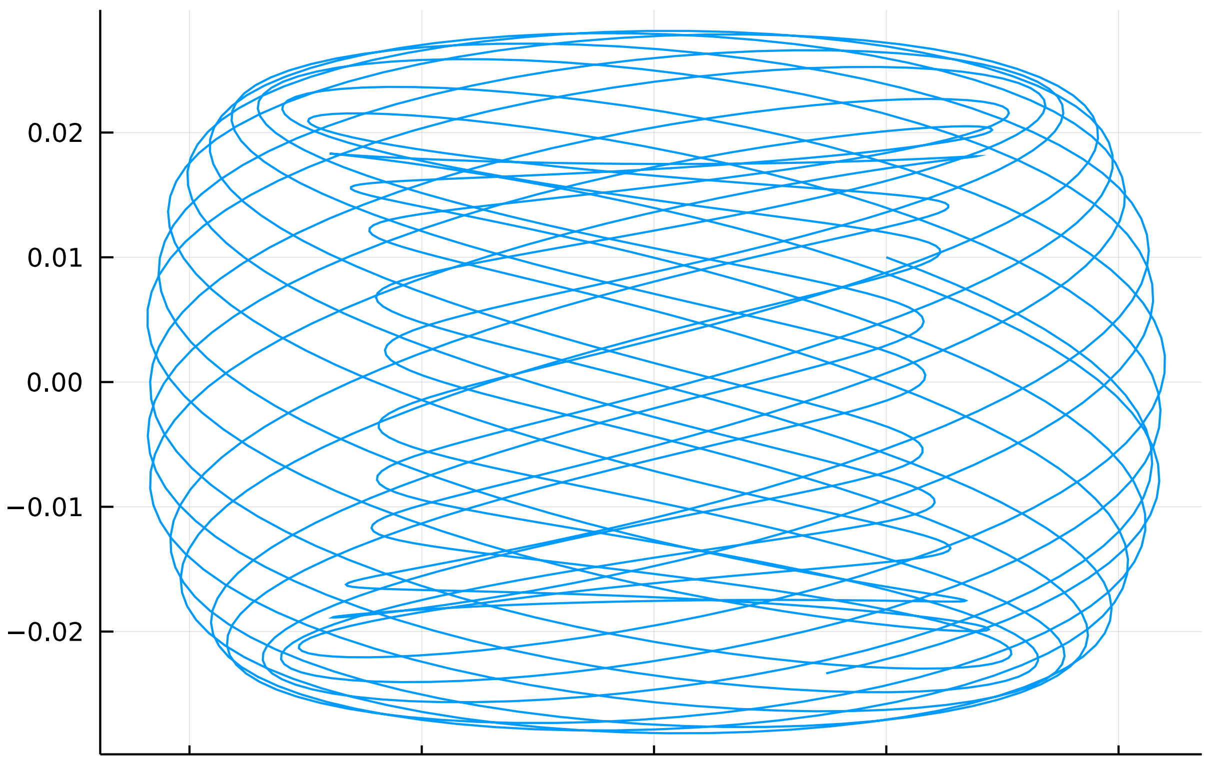
In time, the solution looks like:

```
using Plots; gr()
plot(sol, vars=[(0,3),(0,4)], leg=false, plotdensity=10000)
```



while it has the well-known phase-space plot:

```
plot(sol, vars=(3,4), leg=false)
```



0.0.2 Local Optimization

Let's fine out what some of the local maxima and minima are. Optim.jl can be used to minimize functions, and the solution type has a continuous interpolation which can be used. Let's look for the local optima for the 4th variable around `t=20`. Thus our optimization function is:

```
f = (t) -> sol(t,idxs=4)
```

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

`first(t)` is the same as `t[1]` which transforms the array of size 1 into a number. `idxs=4` is the same as `sol(first(t))[4]` but does the calculation without a temporary array and thus is faster. To find a local minima, we can simply call Optim on this function. Let's find a local minimum:

```
using Optim
opt = optimize(f,18.0,22.0)
```

```
Results of Optimization Algorithm
 * Algorithm: Brent's Method
 * Search Interval: [18.000000, 22.000000]
 * Minimizer: 1.863213e+01
 * Minimum: -2.793164e-02
 * Iterations: 11
 * Convergence: max(|x - x_upper|, |x - x_lower|) <= 2*(1.5e-08*|x|+2.2e-16
): true
 * Objective Function Calls: 12
```

From this printout we see that the minimum is at `t=18.63` and the value is `-2.79e-2`. We can get these in code-form via:

```
println(opt.minimizer)
println(opt.minimum)
```

```
18.632127451866573
-0.027931635651544888
```

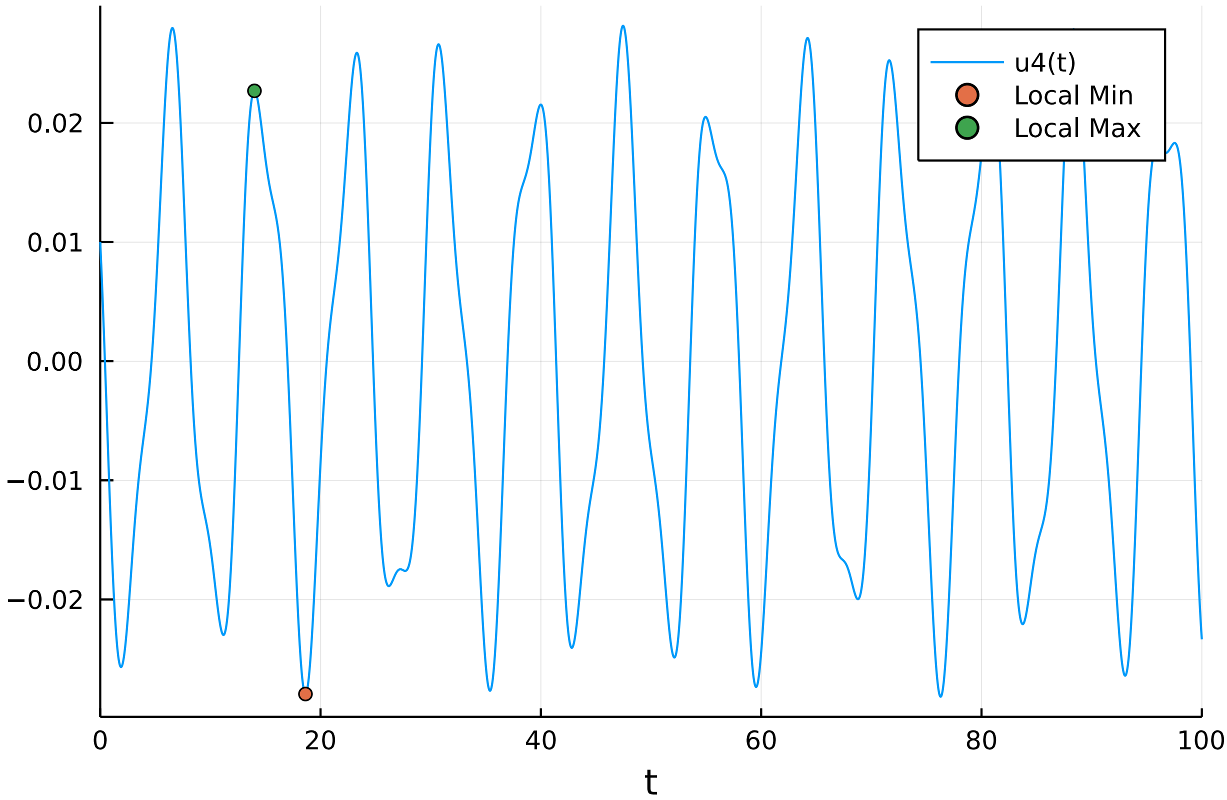
To get the maximum, we just minimize the negative of the function:

```
f = (t) -> -sol(first(t),idxs=4)
opt2 = optimize(f,0.0,22.0)
```

```
Results of Optimization Algorithm
 * Algorithm: Brent's Method
 * Search Interval: [0.000000, 22.000000]
 * Minimizer: 1.399975e+01
 * Minimum: -2.269411e-02
 * Iterations: 13
 * Convergence: max(|x - x_upper|, |x - x_lower|) <= 2*(1.5e-08*|x|+2.2e-16
): true
 * Objective Function Calls: 14
```

Let's add the maxima and minima to the plots:

```
plot(sol, vars=(0,4), plotdensity=10000)
scatter!([opt.minimizer],[opt.minimum],label="Local Min")
scatter!([opt2.minimizer],[-opt2.minimum],label="Local Max")
```



Brent's method will locally minimize over the full interval. If we instead want a local maxima nearest to a point, we can use `BFGS()`. In this case, we need to optimize a vector `[t]`, and thus dereference it to a number using `first(t)`.

```
f = (t) -> -sol(first(t),idxs=4)
opt = optimize(f,[20.0],BFGS())
```

```
* Status: success

 * Candidate solution
    Final objective value:     -2.588588e-02

 * Found with
    Algorithm:     BFGS

 * Convergence measures
    |x - x'|               = 1.11e-04 ≰ 0.0e+00
    |x - x'|/|x'|          = 4.78e-06 ≰ 0.0e+00
    |f(x) - f(x')|         = 1.68e-10 ≰ 0.0e+00
    |f(x) - f(x')|/|f(x')| = 6.49e-09 ≰ 0.0e+00
    |g(x)|                 = 8.44e-12 ≤ 1.0e-08

 * Work counters
    Seconds run:   0  (vs limit Inf)
    Iterations:    4
    f(x) calls:    16
    ∇f(x) calls:   16
```

0.0.3 Global Optimization

If we instead want to find global maxima and minima, we need to look somewhere else. For this there are many choices. A pure Julia option is BlackBoxOptim.jl, but I will use NLopt.jl. Following the NLopt.jl tutorial but replacing their function with out own:

```
import NLopt, ForwardDiff

count = 0 # keep track of # function evaluations

function g(t::Vector, grad::Vector)
  if length(grad) > 0
    #use ForwardDiff for the gradients
    grad[1] = ForwardDiff.derivative((t)->sol(first(t),idxs=4),t)
  end
  sol(first(t),idxs=4)
end
opt = NLopt.Opt(:GN_ORIG_DIRECT_L, 1)
NLopt.lower_bounds!(opt, [0.0])
NLopt.upper_bounds!(opt, [40.0])
NLopt.xtol_rel!(opt,1e-8)
NLopt.min_objective!(opt, g)
(minf,minx,ret) = NLopt.optimize(opt,[20.0])
println(minf," ",minx," ",ret)
NLopt.max_objective!(opt, g)
(maxf,maxx,ret) = NLopt.optimize(opt,[20.0])
println(maxf," ",maxx," ",ret)
```

```
Error: ArgumentError: Package ForwardDiff not found in current path:
- Run `import Pkg; Pkg.add("ForwardDiff")` to install the ForwardDiff packa
ge.
```

```
plot(sol, vars=(0,4), plotdensity=10000)
scatter!([minx],[minf],label="Global Min")
scatter!([maxx],[maxf],label="Global Max")
```

```
Error: UndefVarError: minx not defined
```

0.1 Appendix

These tutorials are a part of the SciMLTutorials.jl repository, found at: https://github.com/SciML/SciML For more information on high-performance scientific machine learning, check out the SciML Open Source Software Organization https://sciml.ai.

To locally run this tutorial, do the following commands:

```
using SciMLTutorials
SciMLTutorials.weave_file("tutorials/ode_extras","03-ode_minmax.jmd")
```

Computer Information:

```
Julia Version 1.6.2
Commit 1b93d53fc4 (2021-07-14 15:36 UTC)
Platform Info:
```

```
  OS: Linux (x86_64-pc-linux-gnu)
  CPU: AMD EPYC 7502 32-Core Processor
  WORD_SIZE: 64
  LIBM: libopenlibm
  LLVM: libLLVM-11.0.1 (ORCJIT, znver2)
Environment:
  JULIA_DEPOT_PATH = /root/.cache/julia-buildkite-plugin/depots/a6029d3a-f78b-41ea-bc9
  JULIA_NUM_THREADS = 16
```

Package Information:

```
      Status `/var/lib/buildkite-agent/builds/5-amdci4-julia-csail-mit-edu/julialang/sc
  [f3b72e0c] DiffEqDevTools v2.27.2
  [0c46a032] DifferentialEquations v6.17.1
  [961ee093] ModelingToolkit v5.17.3
  [76087f3c] NLopt v0.6.2
  [2774e3e8] NLsolve v4.5.1
  [429524aa] Optim v1.3.0
  [1dea7af3] OrdinaryDiffEq v5.56.0
  [91a5bcdd] Plots v1.15.2
  [30cb0354] SciMLTutorials v0.9.0
  [37e2e46d] LinearAlgebra
  [2f01184e] SparseArrays
```

And the full manifest:

```
      Status `/var/lib/buildkite-agent/builds/5-amdci4-julia-csail-mit-edu/julialang/sc
  [c3fe647b] AbstractAlgebra v0.16.0
  [1520ce14] AbstractTrees v0.3.4
  [79e6a3ab] Adapt v3.3.0
  [ec485272] ArnoldiMethod v0.1.0
  [4fba245c] ArrayInterface v3.1.15
  [4c555306] ArrayLayouts v0.7.0
  [aae01518] BandedMatrices v0.16.9
  [6e4b80f9] BenchmarkTools v1.0.0
  [764a87c0] BoundaryValueDiffEq v2.7.1
  [fa961155] CEnum v0.4.1
  [00ebfdb7] CSTParser v2.5.0
  [d360d2e6] ChainRulesCore v0.9.44
  [b630d9fa] CheapThreads v0.2.5
  [523fee87] CodecBzip2 v0.7.2
  [944b1d66] CodecZlib v0.7.0
  [35d6a980] ColorSchemes v3.12.1
  [3da002f7] ColorTypes v0.11.0
  [5ae59095] Colors v0.12.8
  [861a8166] Combinatorics v1.0.2
  [a80b9123] CommonMark v0.8.1
```

```
[38540f10] CommonSolve v0.2.0
[bbf7d656] CommonSubexpressions v0.3.0
[34da2185] Compat v3.30.0
[8f4d0f93] Conda v1.5.2
[187b0558] ConstructionBase v1.2.1
[d38c429a] Contour v0.5.7
[a8cc5b0e] Crayons v4.0.4
[9a962f9c] DataAPI v1.6.0
[864edb3b] DataStructures v0.18.9
[e2d170a0] DataValueInterfaces v1.0.0
[bcd4f6db] DelayDiffEq v5.31.0
[2b5f629d] DiffEqBase v6.62.2
[459566f4] DiffEqCallbacks v2.16.1
[f3b72e0c] DiffEqDevTools v2.27.2
[5a0ffddc] DiffEqFinancial v2.4.0
[c894b116] DiffEqJump v6.14.2
[77a26b50] DiffEqNoiseProcess v5.7.3
[055956cb] DiffEqPhysics v3.9.0
[163ba53b] DiffResults v1.0.3
[b552c78f] DiffRules v1.0.2
[0c46a032] DifferentialEquations v6.17.1
[c619ae07] DimensionalPlotRecipes v1.2.0
[b4f34e82] Distances v0.10.3
[31c24e10] Distributions v0.24.18
[ffbed154] DocStringExtensions v0.8.4
[e30172f5] Documenter v0.26.3
[d4d017d3] ExponentialUtilities v1.8.4
[e2ba6199] ExprTools v0.1.3
[c87230d0] FFMPEG v0.4.0
[7034ab61] FastBroadcast v0.1.8
[9aa1b823] FastClosures v0.3.2
[1a297f60] FillArrays v0.11.7
[6a86dc24] FiniteDiff v2.8.0
[53c48c17] FixedPointNumbers v0.8.4
[59287772] Formatting v0.4.2
[f6369f11] ForwardDiff v0.10.18
[069b7b12] FunctionWrappers v1.1.2
[28b8d3ca] GR v0.57.4
[5c1252a2] GeometryBasics v0.3.12
[42e2da0e] Grisu v1.0.2
[cd3eb016] HTTP v0.9.9
[eafb193a] Highlights v0.4.5
[0e44f5e4] Hwloc v2.0.0
[7073ff75] IJulia v1.23.2
[b5f81e59] IOCapture v0.1.1
[615f187c] IfElse v0.1.0
[d25df0c9] Inflate v0.1.2
[83e8ac13] IniFile v0.5.0
[c8e1da08] IterTools v1.3.0
```

```
[42fd0dbc] IterativeSolvers v0.9.1
[82899510] IteratorInterfaceExtensions v1.0.0
[692b3bcd] JLLWrappers v1.3.0
[682c06a0] JSON v0.21.1
[7d188eb4] JSONSchema v0.3.3
[98e50ef6] JuliaFormatter v0.13.7
[b964fa9f] LaTeXStrings v1.2.1
[2ee39098] LabelledArrays v1.6.1
[23fbe1c1] Latexify v0.15.5
[093fc24a] LightGraphs v1.3.5
[d3d80556] LineSearches v7.1.1
[2ab3a3ac] LogExpFunctions v0.2.4
[bdcacae8] LoopVectorization v0.12.23
[1914dd2f] MacroTools v0.5.6
[b8f27783] MathOptInterface v0.9.22
[fdba3010] MathProgBase v0.7.8
[739be429] MbedTLS v1.0.3
[442fdcdd] Measures v0.3.1
[e1d29d7a] Missings v1.0.0
[961ee093] ModelingToolkit v5.17.3
[46d2c3a1] MuladdMacro v0.2.2
[f9640e96] MultiScaleArrays v1.8.1
[ffc61752] Mustache v1.0.10
[d8a4904e] MutableArithmetics v0.2.19
[d41bc354] NLSolversBase v7.8.0
[76087f3c] NLopt v0.6.2
[2774e3e8] NLsolve v4.5.1
[77ba4419] NaNMath v0.3.5
[8913a72c] NonlinearSolve v0.3.8
[6fe1bfb0] OffsetArrays v1.9.0
[429524aa] Optim v1.3.0
[bac558e1] OrderedCollections v1.4.1
[1dea7af3] OrdinaryDiffEq v5.56.0
[90014a1f] PDMats v0.11.0
[65888b18] ParameterizedFunctions v5.10.0
[d96e819e] Parameters v0.12.2
[69de0a69] Parsers v1.1.0
[ccf2f8ad] PlotThemes v2.0.1
[995b91a9] PlotUtils v1.0.10
[91a5bcdd] Plots v1.15.2
[e409e4f3] PoissonRandom v0.4.0
[f517fe37] Polyester v0.3.1
[85a6dd25] PositiveFactorizations v0.2.4
[21216c6a] Preferences v1.2.2
[1fd47b50] QuadGK v2.4.1
[74087812] Random123 v1.3.1
[fb686558] RandomExtensions v0.4.3
[e6cf234a] RandomNumbers v1.4.0
[3cdcf5f2] RecipesBase v1.1.1
```

```
[01d81517] RecipesPipeline v0.3.2
[731186ca] RecursiveArrayTools v2.11.4
[f2c3362d] RecursiveFactorization v0.1.12
[189a3867] Reexport v1.0.0
[ae029012] Requires v1.1.3
[ae5879a3] ResettableStacks v1.1.0
[79098fc4] Rmath v0.7.0
[47965b36] RootedTrees v1.0.0
[7e49a35a] RuntimeGeneratedFunctions v0.5.2
[476501e8] SLEEFPirates v0.6.20
[1bc83da4] SafeTestsets v0.0.1
[0bca4576] SciMLBase v1.13.4
[30cb0354] SciMLTutorials v0.9.0
[6c6a2e73] Scratch v1.0.3
[efcf1570] Setfield v0.7.0
[992d4aef] Showoff v1.0.3
[699a6c99] SimpleTraits v0.9.3
[b85f4697] SoftGlobalScope v1.1.0
[a2af1166] SortingAlgorithms v1.0.0
[47a9eef4] SparseDiffTools v1.13.2
[276daf66] SpecialFunctions v1.4.1
[aedffcd0] Static v0.2.4
[90137ffa] StaticArrays v1.2.0
[82ae8749] StatsAPI v1.0.0
[2913bbd2] StatsBase v0.33.8
[4c63d2b9] StatsFuns v0.9.8
[9672c7b4] SteadyStateDiffEq v1.6.2
[789caeaf] StochasticDiffEq v6.34.1
[7792a7ef] StrideArraysCore v0.1.11
[09ab397b] StructArrays v0.5.1
[c3572dad] Sundials v4.4.3
[d1185830] SymbolicUtils v0.11.2
[0c5d862f] Symbolics v0.1.25
[3783bdb8] TableTraits v1.0.1
[bd369af6] Tables v1.4.2
[8290d209] ThreadingUtilities v0.4.4
[a759f4b9] TimerOutputs v0.5.9
[0796e94c] Tokenize v0.5.16
[3bb67fe8] TranscodingStreams v0.9.5
[a2a6695c] TreeViews v0.3.0
[5c2747f8] URIs v1.3.0
[3a884ed6] UnPack v1.0.2
[1986cc42] Unitful v1.7.0
[3d5dd08c] VectorizationBase v0.20.11
[81def892] VersionParsing v1.2.0
[19fa3120] VertexSafeGraphs v0.1.2
[44d3d7a6] Weave v0.10.8
[ddb6d928] YAML v0.4.6
[c2297ded] ZMQ v1.2.1
```

```
[a5390f91] ZipFile v0.9.3
[700de1a5] ZygoteRules v0.2.1
[6e34b625] Bzip2_jll v1.0.6+5
[83423d85] Cairo_jll v1.16.0+6
[5ae413db] EarCut_jll v2.1.5+1
[2e619515] Expat_jll v2.2.10+0
[b22a6f82] FFMPEG_jll v4.3.1+4
[a3f928ae] Fontconfig_jll v2.13.1+14
[d7e528f0] FreeType2_jll v2.10.1+5
[559328eb] FriBidi_jll v1.0.5+6
[0656b61e] GLFW_jll v3.3.4+0
[d2c73de3] GR_jll v0.57.2+0
[78b55507] Gettext_jll v0.21.0+0
[7746bdde] Glib_jll v2.68.1+0
[e33a78d0] Hwloc_jll v2.4.1+0
[aacddb02] JpegTurbo_jll v2.0.1+3
[c1c5ebd0] LAME_jll v3.100.0+3
[dd4b983a] LZO_jll v2.10.1+0
[dd192d2f] LibVPX_jll v1.9.0+1
[e9f186c6] Libffi_jll v3.2.2+0
[d4300ac3] Libgcrypt_jll v1.8.7+0
[7e76a0d4] Libglvnd_jll v1.3.0+3
[7add5ba3] Libgpg_error_jll v1.42.0+0
[94ce4f54] Libiconv_jll v1.16.1+0
[4b2f31a3] Libmount_jll v2.35.0+0
[89763e89] Libtiff_jll v4.1.0+2
[38a345b3] Libuuid_jll v2.36.0+0
[079eb43e] NLopt_jll v2.7.0+0
[e7412a2a] Ogg_jll v1.3.4+2
[458c3c95] OpenSSL_jll v1.1.1+6
[efe28fd5] OpenSpecFun_jll v0.5.4+0
[91d4177d] Opus_jll v1.3.1+3
[2f80f16e] PCRE_jll v8.44.0+0
[30392449] Pixman_jll v0.40.1+0
[ea2cea3b] Qt5Base_jll v5.15.2+0
[f50d1b31] Rmath_jll v0.3.0+0
[fb77eaff] Sundials_jll v5.2.0+1
[a2964d1f] Wayland_jll v1.17.0+4
[2381bf8a] Wayland_protocols_jll v1.18.0+4
[02c8fc9c] XML2_jll v2.9.12+0
[aed1982a] XSLT_jll v1.1.34+0
[4f6342f7] Xorg_libX11_jll v1.6.9+4
[0c0b7dd1] Xorg_libXau_jll v1.0.9+4
[935fb764] Xorg_libXcursor_jll v1.2.0+4
[a3789734] Xorg_libXdmcp_jll v1.1.3+4
[1082639a] Xorg_libXext_jll v1.3.4+4
[d091e8ba] Xorg_libXfixes_jll v5.0.3+4
[a51aa0fd] Xorg_libXi_jll v1.7.10+4
[d1454406] Xorg_libXinerama_jll v1.1.4+4
```

```
[ec84b674] Xorg_libXrandr_jll v1.5.2+4
[ea2f1a96] Xorg_libXrender_jll v0.9.10+4
[14d82f49] Xorg_libpthread_stubs_jll v0.1.0+3
[c7cfdc94] Xorg_libxcb_jll v1.13.0+3
[cc61e674] Xorg_libxkbfile_jll v1.1.0+4
[12413925] Xorg_xcb_util_image_jll v0.4.0+1
[2def613f] Xorg_xcb_util_jll v0.4.0+1
[975044d2] Xorg_xcb_util_keysyms_jll v0.4.0+1
[0d47668e] Xorg_xcb_util_renderutil_jll v0.3.9+1
[c22f9ab0] Xorg_xcb_util_wm_jll v0.4.1+1
[35661453] Xorg_xkbcomp_jll v1.4.2+4
[33bec58e] Xorg_xkeyboard_config_jll v2.27.0+4
[c5fb5394] Xorg_xtrans_jll v1.4.0+3
[8f1865be] ZeroMQ_jll v4.3.2+6
[3161d3a3] Zstd_jll v1.5.0+0
[0ac62f75] libass_jll v0.14.0+4
[f638f0a6] libfdk_aac_jll v0.1.6+4
[b53b4c65] libpng_jll v1.6.38+0
[a9144af2] libsodium_jll v1.0.20+0
[f27f6e37] libvorbis_jll v1.3.6+6
[1270edf5] x264_jll v2020.7.14+2
[dfaa095f] x265_jll v3.0.0+3
[d8fb68d0] xkbcommon_jll v0.9.1+5
[0dad84c5] ArgTools
[56f22d72] Artifacts
[2a0f44e3] Base64
[ade2ca70] Dates
[8bb1440f] DelimitedFiles
[8ba89e20] Distributed
[f43a241f] Downloads
[7b1f6079] FileWatching
[9fa8497b] Future
[b77e0a4c] InteractiveUtils
[b27032c2] LibCURL
[76f85450] LibGit2
[8f399da3] Libdl
[37e2e46d] LinearAlgebra
[56ddb016] Logging
[d6f4376e] Markdown
[a63ad114] Mmap
[ca575930] NetworkOptions
[44cfe95a] Pkg
[de0858da] Printf
[3fa0cd96] REPL
[9a3f8284] Random
[ea8e919c] SHA
[9e88b42a] Serialization
[1a1011a3] SharedArrays
[6462fe0b] Sockets
```

```
[2f01184e] SparseArrays
[10745b16] Statistics
[4607b0f0] SuiteSparse
[fa267f1f] TOML
[a4e569a6] Tar
[8dfed614] Test
[cf7118a7] UUIDs
[4ec0a83e] Unicode
[e66e0078] CompilerSupportLibraries_jll
[deac9b47] LibCURL_jll
[29816b5a] LibSSH2_jll
[c8ffd9c3] MbedTLS_jll
[14a3606d] MozillaCACerts_jll
[4536629a] OpenBLAS_jll
[bea87d4a] SuiteSparse_jll
[83775a58] Zlib_jll
[8e850ede] nghttp2_jll
[3f19e933] p7zip_jll
```