FitzHugh-Nagumo Parameter Estimation Benchmarks

Vaibhav Dixit, Chris Rackauckas

July 5, 2020

1 Parameter estimation of FitzHugh-Nagumo model using optimisation methods

```
using ParameterizedFunctions, OrdinaryDiffEq, DiffEqParamEstim
using BlackBoxOptim, NLopt, Plots,QuadDIRECT
```

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

```
gr(fmt=:png)
```

```
Plots.GRBackend()
```

```
loc_bounds = Tuple{Float64,Float64}[(0, 1), (0, 1), (0, 1), (0, 1)]
glo_bounds = Tuple{Float64,Float64}[(0, 5), (0, 5), (0, 5), (0, 5)]
loc_init = [0.5,0.5,0.5,0.5]
glo_init = [2.5,2.5,2.5,2.5]
```

```
4-element Array{Float64,1}:
 2.5
 2.5
 2.5
 2.5
```

```
fitz = @ode_def FitzhughNagumo begin
  dv = v - v^3/3 -w + l
  dw = τinv*(v +  a - b*w)
end a b τinv l
```

```
(::Main.##WeaveSandBox#317.FitzhughNagumo{Main.##WeaveSandBox#317.var"###Pa
rameterizedDiffEqFunction#337",Main.##WeaveSandBox#317.var"###Parameterized
TGradFunction#338",Main.##WeaveSandBox#317.var"###ParameterizedJacobianFunc
tion#339",Nothing,Nothing,ModelingToolkit.ODESystem}) (generic function wit
h 1 method)
```

```
p = [0.7,0.8,0.08,0.5]              # Parameters used to construct the dataset
r0 = [1.0; 1.0]                     # initial value
tspan = (0.0, 30.0)                 # sample of 3000 observations over the (0,30)
timespan
prob = ODEProblem(fitz, r0, tspan,p)
tspan2 = (0.0, 3.0)                 # sample of 300 observations with a timestep of 0.01
prob_short = ODEProblem(fitz, r0, tspan2,p)
```

```
ODEProblem with uType Array{Float64,1} and tType Float64. In-place: true
timespan: (0.0, 3.0)
u0: [1.0, 1.0]
```

```
dt = 30.0/3000
tf = 30.0
tinterval = 0:dt:tf
t  = collect(tinterval)
```

```
3001-element Array{Float64,1}:
  0.0
  0.01
  0.02
  0.03
  0.04
  0.05
  0.06
  0.07
  0.08
  0.09
  ⋮
 29.92
 29.93
 29.94
 29.95
 29.96
 29.97
 29.98
 29.99
 30.0
```

```
h = 0.01
M = 300
tstart = 0.0
tstop = tstart + M * h
tinterval_short = 0:h:tstop
t_short = collect(tinterval_short)
```

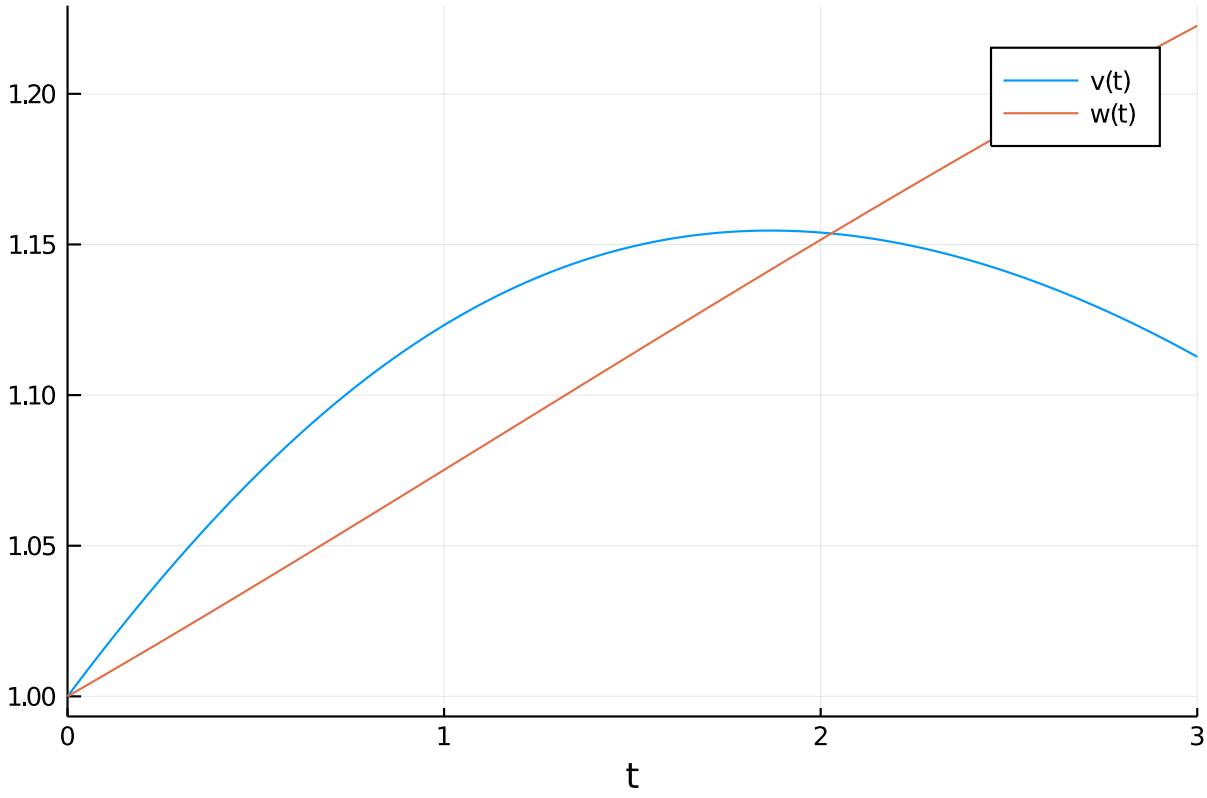
```
301-element Array{Float64,1}:
 0.0
 0.01
 0.02
 0.03
 0.04
 0.05
 0.06
 0.07
 0.08
 0.09
 ⋮
 2.92
 2.93
 2.94
 2.95
 2.96
 2.97
 2.98
 2.99
 3.0
```

```
#Generate Data
data_sol_short = solve(prob_short,Vern9(),saveat=t_short,reltol=1e-9,abstol=1e-9)
data_short = convert(Array, data_sol_short) # This operation produces column major
dataset obs as columns, equations as rows
data_sol = solve(prob,Vern9(),saveat=t,reltol=1e-9,abstol=1e-9)
data = convert(Array, data_sol)
```

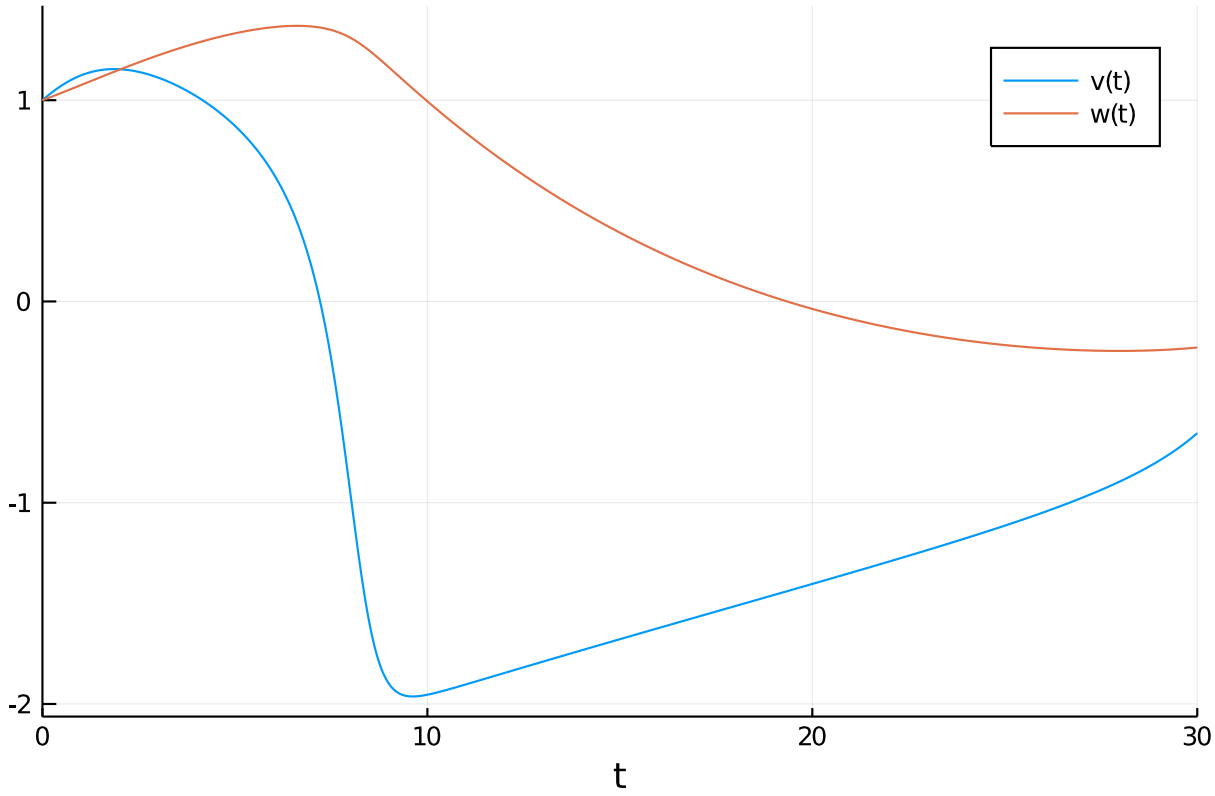
```
2×3001 Array{Float64,2}:
 1.0  1.00166  1.00332  1.00497  1.00661  ...  -0.65759   -0.655923  -0.65424
8
 1.0  1.00072  1.00144  1.00216  1.00289      -0.229157  -0.228976  -0.22879
3
```

Plot of the solution

```
plot(data_sol_short)
```




```
plot(data_sol)
```



1.1 Local Solution from the short data set

```
obj_short =
build_loss_objective(prob_short,Tsit5(),L2Loss(t_short,data_short),tstops=t_short)
res1 = bboptimize(obj_short;SearchRange = glo_bounds, MaxSteps = 7e3)
```

```
Starting optimization with optimizer BlackBoxOptim.DiffEvoOpt{BlackBoxOptim
.FitPopulation{Float64},BlackBoxOptim.RadiusLimitedSelector,BlackBoxOptim.A
daptiveDiffEvoRandBin{3},BlackBoxOptim.RandomBound{BlackBoxOptim.Continuous
RectSearchSpace}}
0.00 secs, 0 evals, 0 steps
0.50 secs, 3385 evals, 3269 steps, improv/step: 0.172 (last = 0.1719), fitn
ess=0.014515096
1.00 secs, 6878 evals, 6762 steps, improv/step: 0.139 (last = 0.1076), fitn
ess=0.000024436

Optimization stopped after 7001 steps and 1.03 seconds
Termination reason: Max number of steps (7000) reached
Steps per second = 6780.74
Function evals per second = 6893.09
Improvements/step = 0.13800
Total function evaluations = 7117


Best candidate found: [0.492203, 0.786429, 0.100594, 0.499825]

Fitness: 0.000024436

# Lower tolerance could lead to smaller fitness (more accuracy)
```

```
obj_short =
build_loss_objective(prob_short,Tsit5(),L2Loss(t_short,data_short),tstops=t_short,reltol=1e-9)
res1 = bboptimize(obj_short;SearchRange = glo_bounds, MaxSteps = 7e3)
```

```
Starting optimization with optimizer BlackBoxOptim.DiffEvoOpt{BlackBoxOptim
.FitPopulation{Float64},BlackBoxOptim.RadiusLimitedSelector,BlackBoxOptim.A
daptiveDiffEvoRandBin{3},BlackBoxOptim.RandomBound{BlackBoxOptim.Continuous
RectSearchSpace}}
0.00 secs, 0 evals, 0 steps
0.50 secs, 3471 evals, 3370 steps, improv/step: 0.189 (last = 0.1893), fitn
ess=0.008784685
1.00 secs, 7095 evals, 6996 steps, improv/step: 0.154 (last = 0.1211), fitn
ess=0.000515705

Optimization stopped after 7001 steps and 1.00 seconds
Termination reason: Max number of steps (7000) reached
Steps per second = 6994.19
Function evals per second = 7093.09
Improvements/step = 0.15400
Total function evaluations = 7100


Best candidate found: [0.268355, 0.83353, 0.158298, 0.500039]

Fitness: 0.000515705
```

```
# Change in tolerance makes it worse

obj_short =
build_loss_objective(prob_short,Vern9(),L2Loss(t_short,data_short),tstops=t_short,reltol=1e-9,abstol=1
res1 = bboptimize(obj_short;SearchRange = glo_bounds, MaxSteps = 7e3)
```

```
Starting optimization with optimizer BlackBoxOptim.DiffEvoOpt{BlackBoxOptim
.FitPopulation{Float64},BlackBoxOptim.RadiusLimitedSelector,BlackBoxOptim.A
daptiveDiffEvoRandBin{3},BlackBoxOptim.RandomBound{BlackBoxOptim.Continuous
RectSearchSpace}}
0.00 secs, 0 evals, 0 steps
0.50 secs, 2390 evals, 2287 steps, improv/step: 0.230 (last = 0.2296), fitn
ess=0.028657339
1.00 secs, 4796 evals, 4694 steps, improv/step: 0.176 (last = 0.1251), fitn
ess=0.000507583

Optimization stopped after 7001 steps and 1.49 seconds
Termination reason: Max number of steps (7000) reached
Steps per second = 4711.55
Function evals per second = 4778.84
Improvements/step = 0.15943
Total function evaluations = 7101


Best candidate found: [0.131948, 0.739195, 0.171571, 0.499862]

Fitness: 0.000304892
```

```
# using the moe accurate Vern9() reduces the fitness marginally and leads to some
increase in time taken
```

1.2 Using NLopt

```
obj_short =
build_loss_objective(prob_short,Vern9(),L2Loss(t_short,data_short),tstops=t_short,reltol=1e-9,abstol=1

(::DiffEqParamEstim.DiffEqObjective{DiffEqParamEstim.var"#43#48"{Nothing,Bo
ol,Int64,typeof(DiffEqParamEstim.STANDARD_PROB_GENERATOR),Base.Iterators.Pa
irs{Symbol,Any,Tuple{Symbol,Symbol,Symbol},NamedTuple{(:tstops, :reltol, :a
bstol),Tuple{Array{Float64,1},Float64,Float64}}},DiffEqBase.ODEProblem{Arra
y{Float64,1},Tuple{Float64,Float64},true,Array{Float64,1},Main.##WeaveSandB
ox#317.FitzhughNagumo{Main.##WeaveSandBox#317.var"###ParameterizedDiffEqFun
ction#337",Main.##WeaveSandBox#317.var"###ParameterizedTGradFunction#338",M
ain.##WeaveSandBox#317.var"###ParameterizedJacobianFunction#339",Nothing,No
thing,ModelingToolkit.ODESystem},Base.Iterators.Pairs{Union{},Union{},Tuple
{},NamedTuple{(),Tuple{}}},DiffEqBase.StandardODEProblem},OrdinaryDiffEq.Ve
rn9,DiffEqParamEstim.L2Loss{Array{Float64,1},Array{Float64,2},Nothing,Nothi
ng,Nothing},Nothing},DiffEqParamEstim.var"#47#53"{DiffEqParamEstim.var"#43#
48"{Nothing,Bool,Int64,typeof(DiffEqParamEstim.STANDARD_PROB_GENERATOR),Bas
e.Iterators.Pairs{Symbol,Any,Tuple{Symbol,Symbol,Symbol},NamedTuple{(:tstop
s, :reltol, :abstol),Tuple{Array{Float64,1},Float64,Float64}}},DiffEqBase.O
DEProblem{Array{Float64,1},Tuple{Float64,Float64},true,Array{Float64,1},Mai
n.##WeaveSandBox#317.FitzhughNagumo{Main.##WeaveSandBox#317.var"###Parameter
izedDiffEqFunction#337",Main.##WeaveSandBox#317.var"###ParameterizedTGradF
unction#338",Main.##WeaveSandBox#317.var"###ParameterizedJacobianFunction#3
39",Nothing,Nothing,ModelingToolkit.ODESystem},Base.Iterators.Pairs{Union{}
,Union{},Tuple{},NamedTuple{(),Tuple{}}},DiffEqBase.StandardODEProblem},Ord
inaryDiffEq.Vern9,DiffEqParamEstim.L2Loss{Array{Float64,1},Array{Float64,2}
,Nothing,Nothing,Nothing},Nothing}}}) (generic function with 2 methods)

opt = Opt(:GN_ORIG_DIRECT_L, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[5.0,5.0,5.0,5.0])
min_objective!(opt, obj_short.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,glo_init)

1.968021 seconds (3.72 M allocations: 683.429 MiB, 2.32% gc time)
(0.11016600768053846, [0.19204389575055014, 1.1316872427993379, 1.111111111
1140621, 0.5095776833484579], :XTOL_REACHED)

opt = Opt(:GN_CRS2_LM, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[5.0,5.0,5.0,5.0])
min_objective!(opt, obj_short.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,glo_init)

2.056717 seconds (3.86 M allocations: 708.619 MiB, 3.25% gc time)
(2.1477098266942874e-16, [0.6999999673585416, 0.8000000427647648, 0.0800000
0680024155, 0.4999999994999742], :MAXEVAL_REACHED)

opt = Opt(:GN_ISRES, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[5.0,5.0,5.0,5.0])
min_objective!(opt, obj_short.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,glo_init)
```

```
2.047030 seconds (3.86 M allocations: 708.619 MiB, 2.69% gc time)
(0.028196273387650954, [4.918072660016354, 4.521118181275437, 0.06819113496
512773, 0.5043603583048624], :MAXEVAL_REACHED)
```

```
opt = Opt(:GN_ESCH, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[5.0,5.0,5.0,5.0])
min_objective!(opt, obj_short.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,glo_init)
```

```
2.039821 seconds (3.86 M allocations: 708.619 MiB, 2.64% gc time)
(0.013104733684654141, [2.5578277801130014, 2.6138727320865005, 0.090863303
2734097, 0.5025836167877843], :MAXEVAL_REACHED)
```

Now local optimization algorithms are used to check the global ones, these use the local constraints, different intial values and time step

```
opt = Opt(:LN_BOBYQA, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[1.0,1.0,1.0,1.0])
min_objective!(opt, obj_short.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,loc_init)
```

```
0.233924 seconds (451.24 k allocations: 82.838 MiB)
(5.347017544723801e-25, [0.700000000027877, 0.8000000000071278, 0.079999999
998271, 0.5000000000000343], :SUCCESS)
```

```
opt = Opt(:LN_NELDERMEAD, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[1.0,1.0,1.0,1.0])
min_objective!(opt, obj_short.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,loc_init)
```

```
0.195830 seconds (357.44 k allocations: 65.618 MiB, 5.52% gc time)
(8.965505337548727e-5, [1.0, 1.0, 0.07355092574875884, 0.500404702213785],
:XTOL_REACHED)
```

```
opt = Opt(:LD_SLSQP, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[1.0,1.0,1.0,1.0])
min_objective!(opt, obj_short.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,loc_init)
```

```
0.383380 seconds (763.19 k allocations: 127.139 MiB, 3.19% gc time)
(3.046792228542053e-14, [0.6999883967325203, 0.799998925267369, 0.080000888
85110575, 0.5000000008669039], :XTOL_REACHED)
```

```
opt = Opt(:LN_COBYLA, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[1.0,1.0,1.0,1.0])
min_objective!(opt, obj_short.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,loc_init)
```

```
2.050752 seconds (3.86 M allocations: 708.619 MiB, 2.72% gc time)
(0.0007192410534949541, [0.18529723007611437, 0.8330107259428428, 0.1928450
2038537973, 0.5003517885479659], :MAXEVAL_REACHED)

opt = Opt(:LN_NEWUOA_BOUND, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[1.0,1.0,1.0,1.0])
min_objective!(opt, obj_short.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,loc_init)

0.081854 seconds (81.84 k allocations: 15.023 MiB)
(0.0003953791439122125, [0.3206565344523288, 0.4365415194546205, 0.07868204
657659038, 0.4991677260203611], :SUCCESS)

opt = Opt(:LN_PRAXIS, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[1.0,1.0,1.0,1.0])
min_objective!(opt, obj_short.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,loc_init)

0.173075 seconds (337.37 k allocations: 61.934 MiB)
(5.165360134583704e-25, [0.7000000000179996, 0.8000000000039914, 0.07999999
999882157, 0.5000000000000274], :XTOL_REACHED)

opt = Opt(:LN_SBPLX, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[1.0,1.0,1.0,1.0])
min_objective!(opt, obj_short.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,loc_init)

2.061385 seconds (3.86 M allocations: 708.619 MiB, 3.15% gc time)
(8.350548444517348e-15, [0.7000068766310836, 0.800001581229246, 0.079999561
72591924, 0.5000000032838425], :MAXEVAL_REACHED)

opt = Opt(:LD_MMA, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[1.0,1.0,1.0,1.0])
min_objective!(opt, obj_short.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,loc_init)

18.318945 seconds (34.72 M allocations: 6.224 GiB, 2.65% gc time)
(0.00010583308079105241, [0.22244491859627777, 0.703544637051332, 0.1324536
414749745, 0.49970791611000387], :MAXEVAL_REACHED)
```

1.2.1 Now the longer problem is solved for a global solution

Vern9 solver with reltol=1e-9 and abstol=1e-9 is used and the dataset is increased to 3000 observations per variable with the same integration time step of 0.01.

```
obj = build_loss_objective(prob,Vern9(),L2Loss(t,data),tstops=t,reltol=1e-9,abstol=1e-9)
res1 = bboptimize(obj;SearchRange = glo_bounds, MaxSteps = 4e3)
```

```
Starting optimization with optimizer BlackBoxOptim.DiffEvoOpt{BlackBoxOptim
.FitPopulation{Float64},BlackBoxOptim.RadiusLimitedSelector,BlackBoxOptim.A
daptiveDiffEvoRandBin{3},BlackBoxOptim.RandomBound{BlackBoxOptim.Continuous
RectSearchSpace}}
0.00 secs, 0 evals, 0 steps
0.50 secs, 254 evals, 180 steps, improv/step: 0.467 (last = 0.4667), fitnes
s=1637.297918224
1.00 secs, 506 evals, 417 steps, improv/step: 0.379 (last = 0.3122), fitnes
s=1354.882597069
1.50 secs, 759 evals, 664 steps, improv/step: 0.321 (last = 0.2227), fitnes
s=1091.748427972
2.00 secs, 1012 evals, 917 steps, improv/step: 0.272 (last = 0.1423), fitne
ss=424.269373633
2.51 secs, 1263 evals, 1168 steps, improv/step: 0.241 (last = 0.1275), fitn
ess=424.269373633
3.01 secs, 1516 evals, 1421 steps, improv/step: 0.223 (last = 0.1423), fitn
ess=356.606806320
3.51 secs, 1769 evals, 1674 steps, improv/step: 0.207 (last = 0.1146), fitn
ess=356.606806320
4.01 secs, 2022 evals, 1927 steps, improv/step: 0.197 (last = 0.1344), fitn
ess=356.606806320
4.51 secs, 2275 evals, 2180 steps, improv/step: 0.188 (last = 0.1186), fitn
ess=257.581268079
5.01 secs, 2528 evals, 2433 steps, improv/step: 0.176 (last = 0.0751), fitn
ess=229.098643272
5.51 secs, 2781 evals, 2686 steps, improv/step: 0.173 (last = 0.1462), fitn
ess=51.749835849
6.01 secs, 3034 evals, 2939 steps, improv/step: 0.169 (last = 0.1186), fitn
ess=16.502862847
6.51 secs, 3287 evals, 3192 steps, improv/step: 0.161 (last = 0.0711), fitn
ess=2.501012920
7.02 secs, 3540 evals, 3445 steps, improv/step: 0.160 (last = 0.1462), fitn
ess=2.501012920
7.52 secs, 3793 evals, 3699 steps, improv/step: 0.157 (last = 0.1142), fitn
ess=1.817512039
8.02 secs, 4046 evals, 3953 steps, improv/step: 0.154 (last = 0.1102), fitn
ess=0.491128592

Optimization stopped after 4001 steps and 8.11 seconds
Termination reason: Max number of steps (4000) reached
Steps per second = 493.30
Function evals per second = 504.76
Improvements/step = 0.15350
Total function evaluations = 4094


Best candidate found: [0.714618, 0.793707, 0.0806293, 0.509453]

Fitness: 0.491128592

BlackBoxOptim.OptimizationResults("adaptive_de_rand_1_bin_radiuslimited", "
Max number of steps (4000) reached", 4001, 1.593921861913241e9, 8.110732078
552246, BlackBoxOptim.DictChain{Symbol,Any}[BlackBoxOptim.DictChain{Symbol,
Any}[Dict{Symbol,Any}(:RngSeed => 245446,:SearchRange => [(0.0, 5.0), (0.0,
 5.0), (0.0, 5.0), (0.0, 5.0)],:MaxSteps => 4000),Dict{Symbol,Any}()],Dict{
Symbol,Any}(:FitnessScheme => BlackBoxOptim.ScalarFitnessScheme{true}(),:Nu
mDimensions => :NotSpecified,:PopulationSize => 50,:MaxTime => 0.0,:SearchR
```

```
ange => (-1.0, 1.0),:Method => :adaptive_de_rand_1_bin_radiuslimited,:MaxNu
mStepsWithoutFuncEvals => 100,:RngSeed => 1234,:MaxFuncEvals => 0,:SaveTrac
e => false...)], 4094, BlackBoxOptim.ScalarFitnessScheme{true}(), BlackBoxOpt
im.TopListArchiveOutput{Float64,Array{Float64,1}}(0.4911285915223069, [0.71
46183683806315, 0.7937072523334917, 0.08062926486024631, 0.5094527247920518
]), BlackBoxOptim.PopulationOptimizerOutput{BlackBoxOptim.FitPopulation{Flo
at64}}(BlackBoxOptim.FitPopulation{Float64}([2.260638808090285 0.3506658851
666085 ... 1.0899906471958247 0.33737302166693817; 2.2607776704308136 0.31970
10497900014 ... 0.9500180075318758 0.3204083710690755; 0.28160002034368753 0.
06317267310446827 ... 0.11517670404008962 0.0629432591602614; 0.6409380870537
328 0.5609394154040026 ... 0.7015763049852316 0.5556268491769338], NaN, [907.
4953605202339, 369.8717201616244, 7.89985607556545, 9.83700974696869, 2.501
012920155386, 0.9939864993403027, 1.0740071798295765, 15.350508360417926, 8
.378050359897614, 2.501012920155386  ...  768.3307439051587, 154.404604058138
23, 855.3727849224631, 821.3314140875556, 476.9662049891442, 870.8451292809
387, 1077.092437280615, 801.9117006321264, 129.9431689991378, 365.168034910
65446], 0, BlackBoxOptim.Candidate{Float64}[BlackBoxOptim.Candidate{Float64
}([0.33737302166693817, 0.3204083710690755, 0.0629432591602614, 0.555626849
1769338], 50, 365.16803491065446, BlackBoxOptim.AdaptiveDiffEvoRandBin{3}(B
lackBoxOptim.AdaptiveDiffEvoParameters(BlackBoxOptim.BimodalCauchy(Distribu
tions.Cauchy{Float64}(μ=0.65, σ=0.1), Distributions.Cauchy{Float64}(μ=1.0,
σ=0.1), 0.5, false, true), BlackBoxOptim.BimodalCauchy(Distributions.Cauchy
{Float64}(μ=0.1, σ=0.1), Distributions.Cauchy{Float64}(μ=0.95, σ=0.1), 0.5,
 false, true), [0.6986395568757303, 0.9513699935450842, 0.7559978844950942,
 0.6936028543695156, 0.9899775556317325, 1.0, 0.7438054226127516, 1.0, 0.69
26204950099375, 0.5748171111321352  ...  0.27480908867446063, 0.7962045296294
875, 0.7701042996207188, 0.8667747996131769, 1.0, 0.679415372180648, 0.7561
879616318882, 1.0, 0.5068973459442596, 0.7051354626409724], [1.0, 0.1571281
9018004565, 0.9113407253551775, 0.18983574926785254, 0.16497867132251653, 0
.11973069134059226, 0.9518973880031999, 0.15369940016729836, 1.0, 0.7627769
58648434  ...  1.0, 0.9152898285958466, 0.9262983074307303, 1.0, 0.1869364716
4952892, 1.0, 0.059720183534589764, 0.8319710237508402, 0.0665633577171579,
 0.07829108969559387])), 0), BlackBoxOptim.Candidate{Float64}([0.7949826054
3759, 1.1421119085519142, 0.3496902638896918, 0.4819624817328057], 50, 2688
.5438351731495, BlackBoxOptim.AdaptiveDiffEvoRandBin{3}(BlackBoxOptim.Adapt
iveDiffEvoParameters(BlackBoxOptim.BimodalCauchy(Distributions.Cauchy{Float
64}(μ=0.65, σ=0.1), Distributions.Cauchy{Float64}(μ=1.0, σ=0.1), 0.5, false
, true), BlackBoxOptim.BimodalCauchy(Distributions.Cauchy{Float64}(μ=0.1, σ
=0.1), Distributions.Cauchy{Float64}(μ=0.95, σ=0.1), 0.5, false, true), [0.
6986395568757303, 0.9513699935450842, 0.7559978844950942, 0.693602854369515
6, 0.9899775556317325, 1.0, 0.7438054226127516, 1.0, 0.6926204950099375, 0.
5748171111321352  ...  0.27480908867446063, 0.7962045296294875, 0.77010429962
07188, 0.8667747996131769, 1.0, 0.679415372180648, 0.7561879616318882, 1.0,
 0.5068973459442596, 0.7051354626409724], [1.0, 0.15712819018004565, 0.9113
407253551775, 0.18983574926785254, 0.16497867132251653, 0.11973069134059226
, 0.9518973880031999, 0.15369940016729836, 1.0, 0.762776958648434  ...  1.0,
0.9152898285958466, 0.9262983074307303, 1.0, 0.18693647164952892, 1.0, 0.05
9720183534589764, 0.8319710237508402, 0.0665633577171579, 0.078291089695593
87])), 0)]))
```

```
opt = Opt(:GN_ORIG_DIRECT_L, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[5.0,5.0,5.0,5.0])
min_objective!(opt, obj.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,glo_init)
```

```
16.116845 seconds (25.18 M allocations: 4.548 GiB, 2.00% gc time)
(81.060918547418, [1.111111111112095, 1.1111111111081604, 0.100594421579125
```

```
43, 0.576131687239848], :XTOL_REACHED)

opt = Opt(:GN_CRS2_LM, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[5.0,5.0,5.0,5.0])
min_objective!(opt, obj.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 20000)
@time (minf,minx,ret) = NLopt.optimize(opt,glo_init)

15.247817 seconds (23.79 M allocations: 4.297 GiB, 2.03% gc time)
(7.74998161395459e-19, [0.6999999999944185, 0.8000000000009041, 0.080000000
00028791, 0.4999999999977666], :XTOL_REACHED)

opt = Opt(:GN_ISRES, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[5.0,5.0,5.0,5.0])
min_objective!(opt, obj.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 50000)
@time (minf,minx,ret) = NLopt.optimize(opt,glo_init)

98.891629 seconds (154.60 M allocations: 27.925 GiB, 2.01% gc time)
(4.2223613315409814e-15, [0.6999999982262703, 0.7999999984962752, 0.0799999
9982409692, 0.49999999927360217], :MAXEVAL_REACHED)

opt = Opt(:GN_ESCH, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[5.0,5.0,5.0,5.0])
min_objective!(opt, obj.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 20000)
@time (minf,minx,ret) = NLopt.optimize(opt,glo_init)

39.632564 seconds (61.84 M allocations: 11.170 GiB, 2.00% gc time)
(165.7478383336907, [1.2218262160551567, 1.4612665220208805, 0.101987593254
05608, 0.4734881708923628], :MAXEVAL_REACHED)

opt = Opt(:LN_BOBYQA, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[1.0,1.0,1.0,1.0])
min_objective!(opt, obj.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,loc_init)

1.010591 seconds (1.58 M allocations: 291.669 MiB, 1.93% gc time)
(7.706166061229668e-19, [0.6999999999964565, 0.8000000000026627, 0.08000000
000038654, 0.4999999999981912], :XTOL_REACHED)

opt = Opt(:LN_NELDERMEAD, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[1.0,1.0,1.0,1.0])
min_objective!(opt, obj.cost_function2)
xtol_rel!(opt,1e-9)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,loc_init)

1.029350 seconds (1.61 M allocations: 297.388 MiB, 1.91% gc time)
(3160.4055222829093, [1.0, 1.0, 1.0, 0.865698720522405], :XTOL_REACHED)
```

```
opt = Opt(:LD_SLSQP, 4)
lower_bounds!(opt,[0.0,0.0,0.0,0.0])
upper_bounds!(opt,[1.0,1.0,1.0,1.0])
min_objective!(opt, obj.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,loc_init)
```

```
0.381379 seconds (590.54 k allocations: 109.224 MiB, 2.59% gc time)
(3160.7697975362676, [0.999947943152157, 0.9999639245452518, 0.999924712179
197, 0.8655666816582644], :XTOL_REACHED)
```

As expected from other problems the longer sample proves to be extremely challenging for some of the global optimizers. A few give the accurate values, while others seem to struggle with accuracy a lot.

```
obj_short =
build_loss_objective(prob_short,Tsit5(),L2Loss(t_short,data_short),tstops=t_short)
lower = [0,0,0,0]
upper = [1,1,1,1]
splits = ([0,0.3,0.7],[0,0.3,0.7],[0,0.3,0.7],[0,0.3,0.7])
@time root, x0 = analyze(obj_short,splits,lower,upper)
```

```
Error: UndefVarError: analyze not defined
```

```
minimum(root)
```

```
Error: UndefVarError: root not defined
```

```
obj = build_loss_objective(prob,Vern9(),L2Loss(t,data),tstops=t,reltol=1e-9,abstol=1e-9)
lower = [0,0,0,0]
upper = [5,5,5,5]
splits = ([0,0.5,1],[0,0.5,1],[0,0.5,1],[0,0.5,1])
@time root, x0 = analyze(obj_short,splits,lower,upper)
```

```
Error: UndefVarError: analyze not defined
```

```
minimum(root)
```

```
Error: UndefVarError: root not defined
```

2 Conclusion

It is observed that lower tolerance lead to higher accuracy but too low tolerance could affect the convergance time drastically. Also fitting a shorter timespan seems to be easier in comparision (quite intutively). NLOpt methods seem to give great accuracy in the shorter problem with a lot of the algorithms giving 0 fitness, BBO performs very well on it with marginal change with tol values. In case of global optimization of the longer problem there is some difference in the perfomance amongst the algorithms with :LN*BOBYQA giving accurate results for the local optimization and :GN*ISRES :GN*CRS2*LM in case of the global give the

highest accuracy. BBO also fails to perform too well in the case of the longer problem. QuadDIRECT performs well in case of the shorter problem but fails to give good results in the longer version.

```
using DiffEqBenchmarks
DiffEqBenchmarks.bench_footer(WEAVE_ARGS[:folder],WEAVE_ARGS[:file])
```

2.1 Appendix

These benchmarks are a part of the DiffEqBenchmarks.jl repository, found at: https://github.com/JuliaDi

To locally run this tutorial, do the following commands:

```
using DiffEqBenchmarks
DiffEqBenchmarks.weave_file("ParameterEstimation","FitzHughNagumoParameterEstimation.jm
```

Computer Information:

```
Julia Version 1.4.2
Commit 44fa15b150* (2020-05-23 18:35 UTC)
Platform Info:
  OS: Linux (x86_64-pc-linux-gnu)
  CPU: Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz
  WORD_SIZE: 64
  LIBM: libopenlibm
  LLVM: libLLVM-8.0.1 (ORCJIT, skylake)
Environment:
  JULIA_DEPOT_PATH = /builds/JuliaGPU/DiffEqBenchmarks.jl/.julia
  JULIA_CUDA_MEMORY_LIMIT = 2147483648
  JULIA_PROJECT = @.
  JULIA_NUM_THREADS = 8
```

Package Information:

```
Status: `/builds/JuliaGPU/DiffEqBenchmarks.jl/benchmarks/ParameterEstimation/Project.to
[6e4b80f9-dd63-53aa-95a3-0cdb28fa8baf] BenchmarkTools 0.5.0
[a134a8b2-14d6-55f6-9291-3336d3ab0209] BlackBoxOptim 0.5.0
[593b3428-ca2f-500c-ae53-031589ec8ddd] CmdStan 6.0.6
[ebbdde9d-f333-5424-9be2-dbf1e9acfb5e] DiffEqBayes 2.16.0
[1130ab10-4a5a-5621-a13d-e4788d82bd4c] DiffEqParamEstim 1.15.0
[ef61062a-5684-51dc-bb67-a0fcdec5c97d] DiffEqUncertainty 1.4.1
[31c24e10-a181-5473-b8eb-7969acd0382f] Distributions 0.23.4
[bbc10e6e-7c05-544b-b16e-64fede858acb] DynamicHMC 2.1.5
```

```
[76087f3c-5699-56af-9a33-bf431cd00edd] NLopt 0.6.0
[1dea7af3-3e70-54e6-95c3-0bf5283fa5ed] OrdinaryDiffEq 5.41.0
[65888b18-ceab-5e60-b2b9-181511a3b968] ParameterizedFunctions 5.3.0
[91a5bcdd-55d7-5caf-9e0b-520d859cae80] Plots 1.5.3
[731186ca-8d62-57ce-b412-fbd966d074cd] RecursiveArrayTools 2.5.0
```