

OREGO Work-Precision Diagrams

Chris Rackauckas

July 6, 2020

```
using OrdinaryDiffEq, DiffEqDevTools, ParameterizedFunctions, Plots, ODE,  
ODEInterfaceDiffEq, LSODA, Sundials
```

```
Error: Failed to precompile OrdinaryDiffEq [1dea7af3-3e70-54e6-95c3-0bf5283  
fa5ed] to /builds/JuliaGPU/DiffEqBenchmarks.jl/.julia/compiled/v1.4/Ordinar  
yDiffEq/DlSvy_YAMOL.ji.
```

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

```
Error: UndefVarError: gr not defined
```

```
using LinearAlgebra  
LinearAlgebra.BLAS.set_num_threads(1)
```

```
f = @code_def Orego begin  
    dy1 = p1*(y2+y1*(1-p2*y1-y2))  
    dy2 = (y3-(1+y1)*y2)/p1  
    dy3 = p3*(y1-y3)  
end p1 p2 p3
```

```
Error: LoadError: UndefVarError: @code_def not defined  
in expression starting at none:2
```

```
p = [77.27,8.375e-6,0.161]  
prob = ODEProblem(f,[1.0,2.0,3.0],[0.0,30.0],p)
```

```
Error: UndefVarError: ODEProblem not defined
```

```
sol = solve(prob,Rodas5(), abstol=1/1014, reltol=1/1014)
```

```
Error: UndefVarError: Rodas5 not defined
```

```
test_sol = TestSolution(sol)
```

```
Error: UndefVarError: TestSolution not defined
```

```
abstols = 1.0 ./ 10.0 .^ (4:11)  
reltols = 1.0 ./ 10.0 .^ (1:8);
```

```
8-element Array{Float64,1}:
```

```
0.1  
0.01  
0.001  
0.0001  
1.0e-5  
1.0e-6  
1.0e-7  
1.0e-8
```

```
plot_prob = ODEProblem(f,[1.0,2.0,3.0] ,(0.0,400.0),p)
```

Error: UndefVarError: ODEProblem not defined

```
sol = solve(plot_prob,CVODE_BDF())
```

Error: UndefVarError: CVODE_BDF not defined

```
plot(sol,yscale=:log10)
```

Error: UndefVarError: plot not defined

0.1 Omissions and Tweaking

The following were omitted from the tests due to convergence failures. ODE.jl's adaptivity is not able to stabilize its algorithms, while GeometricIntegratorsDiffEq has not upgraded to Julia 1.0. GeometricIntegrators.jl's methods used to be either fail to converge at comparable dts (or on some computers errors due to type conversions).

```
#sol = solve(prob,ode23s()); println("Total ODE.jl steps: $(length(sol))")
#using GeometricIntegratorsDiffEq
#try
#     sol = solve(prob,GIRadIIA3(),dt=1/10)
#catch e
#     println(e)
#end
```

```
sol = solve(prob,ARKODE(), abstol=1e-5, reltol=1e-1);
```

Error: UndefVarError: ARKODE not defined

```
sol = solve(prob,ARKODE(nonlinear_convergence_coefficient =
1e-3), abstol=1e-5, reltol=1e-1);
```

Error: UndefVarError: ARKODE not defined

```
sol = solve(prob,ARKODE(order=3), abstol=1e-5, reltol=1e-1);
```

Error: UndefVarError: ARKODE not defined

```
sol = solve(prob,ARKODE(order=3, nonlinear_convergence_coefficient =
1e-5), abstol=1e-5, reltol=1e-1);
```

Error: UndefVarError: ARKODE not defined

```
sol = solve(prob,ARKODE(order=5), abstol=1e-5, reltol=1e-1);
```

Error: UndefVarError: ARKODE not defined

The stabilized explicit methods are not stable enough to handle this problem well. While they don't diverge, they are really slow.

```
setups = [
    #Dict(:alg=>ROCK2())      #Unstable
    #Dict(:alg=>ROCK4())      #needs more iterations
    #Dict(:alg=>ESERK5()),
]
```

0-element Array{Any,1}

The EPIRK and exponential methods also fail:

```
sol = solve(prob,EXPRB53s3(),dt=2.0^(-8));
```

Error: UndefVarError: EXPRB53s3 not defined

```
sol = solve(prob,EPIRK4s3B(),dt=2.0^(-8));
```

Error: UndefVarError: EPIRK4s3B not defined

```
sol = solve(prob,EPIRK5P2(),dt=2.0^(-8));
```

Error: UndefVarError: EPIRK5P2 not defined

PDIRK44 also fails

```
sol = solve(prob,PDIRK44(),dt=2.0^(-8));
```

Error: UndefVarError: PDIRK44 not defined

0.2 High Tolerances

This is the speed when you just want the answer.

```
abstols = 1.0 ./ 10.0 .^ (5:8)
reltols = 1.0 ./ 10.0 .^ (1:4);
setups = [Dict(:alg=>Rosenbrock23()),
          Dict(:alg=>Rodas3()),
          Dict(:alg=>TRBDF2()),
          Dict(:alg=>CVODE_BDF()),
          Dict(:alg=>rodas()),
          Dict(:alg=>radau()),
          Dict(:alg=>RadauIIA5()),
          Dict(:alg=>ROS34PW1a()),
          Dict(:alg=>lsoda()),
          ]
```

Error: UndefVarError: Rosenbrock23 not defined

```
wp = WorkPrecisionSet(prob,abstols,reltols,setups;
                      save_everystep=false,appxsol=test_sol,maxiters=Int(1e5),numruns=10)
```

Error: UndefVarError: test_sol not defined

```
plot(wp)
```

Error: UndefVarError: plot not defined

```
wp = WorkPrecisionSet(prob,abstols,reltols,setups;dense = false,verbose=false,
                      appxsol=test_sol,maxiters=Int(1e5),error_estimate=:l2,numruns=10)
```

Error: UndefVarError: test_sol not defined

```
plot(wp)
```

Error: UndefVarError: plot not defined

```
wp = WorkPrecisionSet(prob, abstols, reltols, setups;
    appxsol=test_sol, maxiters=Int(1e5), error_estimate=:L2, numruns=10)
```

Error: UndefVarError: test_sol not defined

```
plot(wp)
```

Error: UndefVarError: plot not defined

```
setups = [Dict(:alg=>Rosenbrock23()),
    Dict(:alg=>Kvaerno3()),
    Dict(:alg=>CVODE_BDF()),
    Dict(:alg=>KenCarp4()),
    Dict(:alg=>TRBDF2()),
    Dict(:alg=>KenCarp3()),
    # Dict(:alg=>SDIRK2()), # Removed because it's bad
    Dict(:alg=>radau())]
```

Error: UndefVarError: Rosenbrock23 not defined

```
wp = WorkPrecisionSet(prob, abstols, reltols, setups;
    save_everystep=false, appxsol=test_sol, maxiters=Int(1e5), numruns=10)
```

Error: UndefVarError: test_sol not defined

```
plot(wp)
```

Error: UndefVarError: plot not defined

```
wp = WorkPrecisionSet(prob, abstols, reltols, setups; dense = false, verbose = false,
    appxsol=test_sol, maxiters=Int(1e5), error_estimate=:l2, numruns=10)
```

Error: UndefVarError: test_sol not defined

```
plot(wp)
```

Error: UndefVarError: plot not defined

```
wp = WorkPrecisionSet(prob, abstols, reltols, setups;
    appxsol=test_sol, maxiters=Int(1e5), error_estimate=:L2, numruns=10)
```

Error: UndefVarError: test_sol not defined

```
plot(wp)
```

Error: UndefVarError: plot not defined

```
setups = [Dict(:alg=>Rosenbrock23()),
    Dict(:alg=>KenCarp5()),
    Dict(:alg=>KenCarp4()),
    Dict(:alg=>KenCarp3()),
    Dict(:alg=>ARKODE(order=5)),
    Dict(:alg=>ARKODE(nonlinear_convergence_coefficient = 1e-6)),
    Dict(:alg=>ARKODE(nonlinear_convergence_coefficient = 1e-5, order=3))
]
```

Error: UndefVarError: Rosenbrock23 not defined

```
names = ["Rosenbrock23" "KenCarp5" "KenCarp4" "KenCarp3" "ARKODE5" "ARKODE4" "ARKODE3"]
wp = WorkPrecisionSet(prob, abstols, reltols, setups;
    names=names,
    save_everystep=false, appxsol=test_sol, maxiters=Int(1e5), numruns=10)
```

Error: UndefVarError: test_sol not defined

```
plot(wp)
```

Error: UndefVarError: plot not defined

```
setups = [Dict(:alg=>Rosenbrock23()),
    Dict(:alg=>TRBDF2()),
    Dict(:alg=>ImplicitEulerExtrapolation()),
    #Dict(:alg=>ImplicitDeufhardExtrapolation()), # Diverges
    #Dict(:alg=>ImplicitHairerWannerExtrapolation()), # Diverges
    Dict(:alg=>ABDF2()),
    Dict(:alg=>QNDF()),
    Dict(:alg=>Exprb43()),
    Dict(:alg=>Exprb32()),
]
```

Error: UndefVarError: Rosenbrock23 not defined

```
wp = WorkPrecisionSet(prob, abstols, reltols, setups;
    save_everystep=false, appxsol=test_sol, maxiters=Int(1e5), numruns=10)
```

Error: UndefVarError: test_sol not defined

```
plot(wp)
```

Error: UndefVarError: plot not defined

0.2.1 Low Tolerances

This is the speed at lower tolerances, measuring what's good when accuracy is needed.

```
abstols = 1.0 ./ 10.0 .^ (7:13)
reltols = 1.0 ./ 10.0 .^ (4:10)
```

```
setups = [Dict(:alg=>GRK4A()),
    Dict(:alg=>Rodas4P()),
    Dict(:alg=>CVODE_BDF()),
    Dict(:alg=>ddebdf()),
    Dict(:alg=>Rodas4()),
    Dict(:alg=>rodas()),
    Dict(:alg=>radau()),
    Dict(:alg=>RadauIIA5()),
    Dict(:alg=>lsoda()),
]
```

Error: UndefVarError: GRK4A not defined

```
wp = WorkPrecisionSet(prob, abstols, reltols, setups;
    save_everystep=false, appxsol=test_sol, maxiters=Int(1e5), numruns=10)
```

Error: UndefVarError: test_sol not defined

```
plot(wp)
```

Error: UndefinedVarError: plot not defined

```
wp = WorkPrecisionSet(prob, abstols, reltols, setups; verbose=false,  
dense=false, appxsol=test_sol, maxiters=Int(1e5), error_estimate=:l2, numruns=10)
```

Error: UndefinedVarError: test_sol not defined

```
plot(wp)
```

Error: UndefinedVarError: plot not defined

```
wp = WorkPrecisionSet(prob, abstols, reltols, setups;  
appxsol=test_sol, maxiters=Int(1e5), error_estimate=:L2, numruns=10)
```

Error: UndefinedVarError: test_sol not defined

```
plot(wp)
```

Error: UndefinedVarError: plot not defined

```
setups = [  
    Dict(:alg=>Rodas5()),  
    Dict(:alg=>Kvaerno5()),  
    Dict(:alg=>CVODE_BDF()),  
    Dict(:alg=>KenCarp4()),  
    Dict(:alg=>KenCarp5()),  
    Dict(:alg=>Rodas4()),  
    Dict(:alg=>radau())]
```

Error: UndefinedVarError: Rodas5 not defined

```
wp = WorkPrecisionSet(prob, abstols, reltols, setups;  
save_everystep=false, appxsol=test_sol, maxiters=Int(1e5), numruns=10)
```

Error: UndefinedVarError: test_sol not defined

```
plot(wp)
```

Error: UndefinedVarError: plot not defined

```
wp = WorkPrecisionSet(prob, abstols, reltols, setups; verbose=false,  
dense=false, appxsol=test_sol, maxiters=Int(1e5), error_estimate=:l2, numruns=10)
```

Error: UndefinedVarError: test_sol not defined

```
plot(wp)
```

Error: UndefinedVarError: plot not defined

```
wp = WorkPrecisionSet(prob, abstols, reltols, setups;  
appxsol=test_sol, maxiters=Int(1e5), error_estimate=:L2, numruns=10)
```

Error: UndefinedVarError: test_sol not defined

```
plot(wp)
```

Error: UndefinedVarError: plot not defined

The following algorithms were removed since they failed.

```
#setups = [Dict(:alg=>Hairer4()),
            #Dict(:alg=>Hairer42()),
            #Dict(:alg=>Rodas3()),
            #Dict(:alg=>Kvaerno4()),
            #Dict(:alg=>Cash4())
#]
#wp = WorkPrecisionSet(prob, abstols, reltols, setups;
#
#save_everystep=false, appxsol=test_sol, maxiters=Int(1e5), numruns=10)
#plot(wp)
```

0.2.2 Conclusion

At high tolerances, `Rosenbrock23` hits the the error estimates and is fast. At lower tolerances and normal user tolerances, `Rodas4` and `Rodas5` are extremely fast. When you get down to `reltol=1e-9` `radau` begins to become as efficient as `Rodas4`, and it continues to do well below that.

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

0.3 Appendix

These benchmarks are a part of the `DiffEqBenchmarks.jl` repository, found at: <https://github.com/JuliaDiffEq/DiffEqBenchmarks.jl>

To locally run this tutorial, do the following commands:

```
using DiffEqBenchmarks
DiffEqBenchmarks.weave_file("StiffODE", "Orego.jmd")
```

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/StiffODE/Project.toml`  
[eb300fae-53e8-50a0-950c-e21f52c2b7e0] DiffEqBiological 4.3.0  
[f3b72e0c-5b89-59e1-b016-84e28bfd966d] DiffEqDevTools 2.22.0  
[5a33fad7-5ce4-5983-9f5d-5f26ceab5c96] GeometricIntegratorsDiffEq 0.1.0  
[7f56f5a3-f504-529b-bc02-0b1fe5e64312] LSODA 0.6.1  
[c030b06c-0b6d-57c2-b091-7029874bd033] ODE 2.5.0  
[09606e27-ecf5-54fc-bb29-004bd9f985bf] ODEInterfaceDiffEq 3.7.0  
[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  
[b4db0fb7-de2a-5028-82bf-5021f5cfa881] ReactionNetworkImporters 0.1.5  
[c3572dad-4567-51f8-b174-8c6c989267f4] Sundials 4.2.5  
[a759f4b9-e2f1-59dc-863e-4aeb61b1ea8f] TimerOutputs 0.5.6  
[37e2e46d-f89d-539d-b4ee-838fcccc9c8e] LinearAlgebra
```