Oval2 Long Times

Chris Rackauckas

July 5, 2020

```
using StochasticDiffEq, DiffEqProblemLibrary, Random
Random.seed!(200)
using DiffEqProblemLibrary.SDEProblemLibrary: importsdeproblems; importsdeproblems()
prob =
DiffEqProblemLibrary.SDEProblemLibrary.oval2ModelExample(largeFluctuations=true,useBigs=false)

SDEProblem with uType Array{Float64,1} and tType Float64. In-place: true
timespan: (0.0, 500.0)
u0: [0.128483, 1.256853, 0.0030203, 0.0027977, 0.0101511, 0.0422942, 0.2391
346, 0.0008014, 0.0001464, 2.67e-5, 4.8e-6, 9.0e-7, 0.0619917, 1.2444292, 0
.0486676, 199.9383546, 137.4267984, 1.5180203, 1.5180203]

Random.seed!(200)
@time for i in 1:10
    sol = solve(prob,SRIW1(),dt=(1/2)^(18),qmax=1.125,
        saveat=0.1,maxiters=1e7,abstol=1e-5,reltol=1e-3)
end

95.087560 seconds (337.57 M allocations: 56.860 GiB, 7.55% gc time)

Random.seed!(200)
@time for i in 1:10
    @show i
    sol = solve(prob,ImplicitEM(),dt=1/60000)
end

i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
i = 10
  0.270168 seconds (1.21 M allocations: 89.069 MiB, 6.73% gc time)

Random.seed!(200)
@time for i in 1:10
    @show i
    sol = solve(prob,ImplicitRKMil(),dt=1/50000)
end
```

```
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
i = 10
  1.494594 seconds (6.78 M allocations: 602.419 MiB, 5.88% gc time)
```

```
Random.seed!(200)
@time for i in 1:10
    sol = solve(prob,SOSRI(),dt=(1/2)^(18),qmax=1.125,
          saveat=0.1,maxiters=1e7,abstol=1e-4,reltol=1e-2)
end
```

```
40.372694 seconds (118.55 M allocations: 20.732 GiB, 4.07% gc time)
```

```
Random.seed!(200)
@time for i in 1:10
    sol = solve(prob,SOSRI2(),dt=(1/2)^(18),qmax=1.125,
          saveat=0.1,maxiters=1e7,abstol=1e-4,reltol=1e-4)
end
```

```
137.070111 seconds (403.18 M allocations: 70.490 GiB, 3.38% gc time)
```

```
Random.seed!(200)
@time for i in 1:10
    sol = solve(prob,SOSRI2(),dt=(1/2)^(18),qmax=1.125,
          saveat=0.1,maxiters=1e7,abstol=1e-5,reltol=1e-3)
end
```

```
138.298853 seconds (404.03 M allocations: 70.675 GiB, 3.46% gc time)
```

```
Random.seed!(200)
@time for i in 1:10
    sol = solve(prob,SOSRI(),dt=(1/2)^(18),qmax=1.125,
          saveat=0.1,maxiters=1e7,abstol=1e-3,reltol=1e-2)
end
```

```
22.367574 seconds (65.20 M allocations: 11.399 GiB, 3.71% gc time)
```

```
Random.seed!(200)
@time for i in 1:10
    sol = solve(prob,SOSRI(),dt=(1/2)^(18),qmax=1.125,
          saveat=0.1,maxiters=1e7,abstol=1e-4,reltol=1e-4)
end
```

```
139.375987 seconds (402.36 M allocations: 70.411 GiB, 3.65% gc time)
```

```
Random.seed!(200)
@time for i in 1:10
    sol = solve(prob,SOSRI(),dt=(1/2)^(18),qmax=1.125,
          saveat=0.1,maxiters=1e7,abstol=1e-2,reltol=1e-2)
end
```

```
13.210645 seconds (38.40 M allocations: 6.703 GiB, 3.97% gc time)
```

```
Random.seed!(200)
@time for i in 1:10
    sol = solve(prob,SOSRI(),dt=(1/2)^(18),qmax=1.125,
          saveat=0.1,maxiters=1e7,abstol=1e-5,reltol=1e-3)
end
```

```
141.348569 seconds (404.77 M allocations: 70.792 GiB, 3.83% gc time)
```

```
Random.seed!(200)
@time for i in 1:10
    sol = solve(prob,SOSRI(),dt=(1/2)^(18),qmax=1.125,
          saveat=0.1,maxiters=1e7,abstol=1e-2,reltol=1e-1)
end
```

```
13.969114 seconds (39.82 M allocations: 6.958 GiB, 4.02% gc time)
```

```
Random.seed!(200)
@time for i in 1:10
    sol = solve(prob,SOSRI2(),dt=(1/2)^(18),qmax=1.125,
          saveat=0.1,maxiters=1e7,abstol=1e-4,reltol=1e-1)
end
```

```
8.404336 seconds (24.10 M allocations: 4.158 GiB, 4.07% gc time)
```

```
Random.seed!(200)
@time for i in 1:10
    @show i
    sol = solve(prob,ImplicitEM(),dt=1/50000)
end
```

```
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
i = 10
  0.186945 seconds (935.58 k allocations: 67.902 MiB)
```

```
Random.seed!(200)
@time for i in 1:10
    @show i
    sol = solve(prob,ImplicitRKMil(),dt=1/40000)
end
```

```
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
i = 10
  0.737950 seconds (3.69 M allocations: 328.117 MiB, 2.78% gc time)
```

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

0.1 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("StiffSDE","Oval2LongTimes.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/StiffSDE/Project.toml`
[f3b72e0c-5b89-59e1-b016-84e28bfd966d] DiffEqDevTools 2.22.0
[77a26b50-5914-5dd7-bc55-306e6241c503] DiffEqNoiseProcess 5.0.2
[a077e3f3-b75c-5d7f-a0c6-6bc4c8ec64a9] DiffEqProblemLibrary 4.8.0
[91a5bcdd-55d7-5caf-9e0b-520d859cae80] Plots 1.5.3
[789caeaf-c7a9-5a7d-9973-96adeb23e2a0] StochasticDiffEq 6.24.0
[37e2e46d-f89d-539d-b4ee-838fcccc9c8e] LinearAlgebra
[9a3f8284-a2c9-5f02-9a11-845980a1fd5c] Random
[10745b16-79ce-11e8-11f9-7d13ad32a3b2] Statistics
```