

Quadratic Stiffness Benchmarks

Chris Rackauckas

July 4, 2020

1 Quadratic Stiffness

In this notebook we will explore the quadratic stiffness problem. References:

The composite Euler method for stiff stochastic differential equations

Kevin Burrage, Tianhai Tian

And

S-ROCK: CHEBYSHEV METHODS FOR STIFF STOCHASTIC DIFFERENTIAL EQUATIONS

ASSYR ABDULLE AND STEPHANE CIRILLI

This is a scalar SDE with two arguments. The first controls the deterministic stiffness and the later controls the diffusion stiffness.

```
using DiffEqProblemLibrary, StochasticDiffEq, DiffEqDevTools
```

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

```
using DiffEqProblemLibrary.SDEProblemLibrary: importsdeproblems; importsdeproblems()
```

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

```
import DiffEqProblemLibrary.SDEProblemLibrary: prob_sde_stiffquadito
```

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

```
using Plots; gr()
```

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

```
const N = 10
```

10

```
prob = remake(prob_sde_stiffquadito,p=(50.0,1.0))
```

Error: UndefVarError: remake not defined

```
sol = solve(prob,SRIW1())
```

Error: UndefVarError: SRIW1 not defined

```
plot(sol)
```

Error: UndefVarError: plot not defined

```
prob = remake(prob_sde_stiffquadito,p=(500.0,1.0))
```

Error: UndefVarError: remake not defined

```
sol = solve(prob,SRIW1())
```

Error: UndefVarError: SRIW1 not defined

```
plot(sol)
```

Error: UndefVarError: plot not defined

1.1 Top dts

Let's first determine the maximum dts which are allowed. Anything higher is mostly unstable.

1.1.1 Deterministic Stiffness Mild

```
prob = remake(prob_sde_stiffquadito,p=(50.0,1.0))
```

Error: UndefVarError: remake not defined

```
@time sol = solve(prob,SRIW1())
```

Error: UndefVarError: SRIW1 not defined

```
@time sol = solve(prob,SRIW1(),adaptive=false,dt=0.01)
```

Error: UndefVarError: SRIW1 not defined

```
@time sol = solve(prob,ImplicitRKMil(),dt=0.005)
```

Error: UndefVarError: ImplicitRKMil not defined

```
@time sol = solve(prob,EM(),dt=0.01);
```

Error: UndefVarError: EM not defined

1.1.2 Deterministic Stiffness High

```
prob = remake(prob_sde_stiffquadito,p=(500.0,1.0))
```

Error: UndefVarError: remake not defined

```
@time sol = solve(prob,SRIW1())
```

Error: UndefVarError: SRIW1 not defined

```
@time sol = solve(prob,SRIW1(),adaptive=false,dt=0.002)
```

Error: UndefVarError: SRIW1 not defined

```
@time sol = solve(prob,ImplicitRKMil(),dt=0.001)
```

Error: UndefVarError: ImplicitRKMil not defined

```
@time sol = solve(prob,EM(),dt=0.002);
```

Error: UndefVarError: EM not defined

1.1.3 Mixed Stiffness

```
prob = remake(prob_sde_stiffquadito,p=(5000.0,70.0))
```

Error: UndefVarError: remake not defined

```
@time sol = solve(prob,SRIW1(),dt=0.0001)
```

Error: UndefVarError: SRIW1 not defined

```
@time sol = solve(prob,SRIW1(),adaptive=false,dt=0.00001)
```

Error: UndefVarError: SRIW1 not defined

```
@time sol = solve(prob,ImplicitRKMil(),dt=0.00001)
```

Error: UndefVarError: ImplicitRKMil not defined

```
@time sol = solve(prob,EM(),dt=0.00001);
```

Error: UndefVarError: EM not defined

Notice that in this problem, the stiffness in the noise term still prevents the semi-implicit integrator to do well. In that case, the advantage of implicitness does not take effect, and thus explicit methods do well. When we don't care about the error, Euler-Maruyama is fastest. When there's mixed stiffness, the adaptive algorithm is unstable.

1.2 Work-Precision Diagrams

```
prob = remake(prob_sde_stiffquadito,p=(50.0,1.0))
```

Error: UndefVarError: remake not defined

```
reltols = 1.0 ./ 10.0 .^ (3:5)
```

```
abstols = reltols#[0.0 for i in eachindex(reltols)]
```

```
setups = [Dict(:alg=>SRIW1()),
```

```
          Dict(:alg=>EM(),:dts=>1.0./8.0.^((1:length(reltols)) .+ 1)),
```

```
          Dict(:alg=>SRIW1(),:dts=>1.0./8.0.^((1:length(reltols)) .+ 1),:adaptive=>false)
```

```
          #Dict(:alg=>RKMil(),:dts=>1.0./8.0.^((1:length(reltols)) .+
```

```
          1),:adaptive=>false),
```

```
        ]
```

```
Error: UndefVarError: SRIW1 not defined
```

```
names = ["SRIW1", "EM", "SRIW1 Fixed"] # "RKMil",
wp =
WorkPrecisionSet(prob, abstols, reltols, setups; numruns=N, names=names, error_estimate=:l2)
```

```
Error: UndefVarError: WorkPrecisionSet not defined
```

```
plot(wp)
```

```
Error: UndefVarError: plot not defined
```

```
prob = remake(prob_sde_stiffquadito, p=(500.0, 1.0))
```

```
Error: UndefVarError: remake not defined
```

```
reltols = 1.0 ./ 10.0 .^ (3:5)
abstols = reltols # [0.0 for i in eachindex(reltols)]
setups = [Dict{:alg=>SRIW1()},
           Dict{:alg=>EM(), :dts=>1.0./8.0.^((1:length(reltols)) .+ 2)},
           Dict{:alg=>SRIW1(), :dts=>1.0./8.0.^((1:length(reltols)) .+ 2), :adaptive=>false}
           #Dict{:alg=>RKMil(), :dts=>1.0./8.0.^((1:length(reltols)) .+
           2), :adaptive=>false},
          ]
```

```
Error: UndefVarError: SRIW1 not defined
```

```
names = ["SRIW1", "EM", "SRIW1 Fixed"] # "RKMil",
wp =
WorkPrecisionSet(prob, abstols, reltols, setups; numruns=N, names=names, error_estimate=:l2, print_names=true)
```

```
Error: UndefVarError: WorkPrecisionSet not defined
```

```
plot(wp)
```

```
Error: UndefVarError: plot not defined
```

1.3 Conclusion

Noise stiffness is tough. Right now the best solution is to run an explicit integrator with a low enough dt. Adaptivity does have a cost in this case, likely due to memory management.

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

1.4 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("StiffSDE", "QuadraticStiffness.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 = 4
```

Package Information:

```
Status: `~/builds/JuliaGPU/DiffEqBenchmarks.jl/Project.toml`
[7073ff75-c697-5162-941a-fcdaad2a7d2a] IJulia 1.21.2
[44d3d7a6-8a23-5bf8-98c5-b353f8df5ec9] Weave 0.10.2
[b77e0a4c-d291-57a0-90e8-8db25a27a240] InteractiveUtils
[d6f4376e-aef5-505a-96c1-9c027394607a] Markdown
[44cfe95a-1eb2-52ea-b672-e2afdf69b78f] Pkg
```