SDE SIR

Simon Frost

April 27, 2020

```julia
using DifferentialEquations
using SimpleDiffEq
using Distributions
using Random
using Plots
using BenchmarkTools

function sir_sde(du,u,p,t)
    (S,I,R) = u
    (β,γ,δt) = p
    N = S+I+R
    ifrac = β*I/N*S*δt
    rfrac = γ*I*δt
    ifrac_noise = sqrt(ifrac)*rand(Normal(0,1))
    rfrac_noise = sqrt(rfrac)*rand(Normal(0,1))
    @inbounds begin
        du[1] = S-(ifrac+ifrac_noise)
        du[2] = I+(ifrac+ifrac_noise) - (rfrac + rfrac_noise)
        du[3] = R+(rfrac+rfrac_noise)
    end
    for i in 1:3
        if du[i] < 0 du[i]=0 end
    end
    nothing
end
```

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

```julia
δt = 0.01
nsteps = 5000
tf = nsteps*δt
tspan = (0.0,nsteps)
```

```
(0.0, 5000)
```

```julia
u0 = [990.0,10.0,0.0]
p = [0.5,0.25,0.01]
```

```
3-element Array{Float64,1}:
 0.5
 0.25
 0.01
```

```julia
Random.seed!(1234)
```

Random.MersenneTwister(UInt32[0x000004d2], Random.DSFMT.DSFMT_state(Int32[-
1393240018, 1073611148, 45497681, 1072875908, 436273599, 1073674613, -20437
16458, 1073445557, -254908435, 1072827086 ...   -599655111, 1073144102, 36765
5457, 1072985259, -1278750689, 1018350124, -597141475, 249849711, 382, 0]),
 [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ...   0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0], UInt128[0x00000000000000000000000000000000,
0x00000000000000000000000000000000, 0x00000000000000000000000000000000, 0x0
0000000000000000000000000000000, 0x00000000000000000000000000000000, 0x0000
0000000000000000000000000000, 0x00000000000000000000000000000000, 0x0000000
0000000000000000000000000, 0x00000000000000000000000000000000, 0x0000000000
00000000000000000000000 ...   0x00000000000000000000000000000000, 0x0000000000
000000000000000000000000, 0x00000000000000000000000000000000, 0x0000000000000
00000000000000000000, 0x00000000000000000000000000000000, 0x0000000000000000
0000000000000000, 0x00000000000000000000000000000000, 0x0000000000000000000
0000000000000, 0x00000000000000000000000000000000, 0x00000000000000000000000
0000000000], 1002, 0)

```
prob_sir_sde = DiscreteProblem((du,u,p,t)->sir_sde(du,u,p,t),u0,tspan,p)
sol_sir_sde = solve(prob_sir_sde,solver=FunctionMap)
```

```
retcode: Success
Interpolation: left-endpoint piecewise constant
t: 5001-element Array{Float64,1}:
    0.0
    1.0
    2.0
    3.0
    4.0
    5.0
    6.0
    7.0
    8.0
    9.0
    ⋮
 4992.0
 4993.0
 4994.0
 4995.0
 4996.0
 4997.0
 4998.0
 4999.0
 5000.0
u: 5001-element Array{Array{Float64,1},1}:
 [990.0, 10.0, 0.0]
 [989.7575274297931, 10.360050786410875, 0.0]
 [989.8182210764063, 10.418767672664574, 0.0]
 [989.5704084759752, 10.283557162266963, 0.383023110828716]
 [989.3993669198878, 10.47245985961835, 0.36516196956474867]
 [989.2332481519006, 10.696048478585872, 0.3076921185844038]
 [989.3092616894122, 10.596449490521206, 0.3312775691374906]
 [989.2275420377456, 10.35011732186858, 0.6593293894567847]
 [989.363627745111, 10.170446479489094, 0.7029145244708828]
 [989.3696608412107, 10.080034261645968, 0.7872936462142908]
 ⋮
 [241.03773979754035, 4.480316516535241, 754.8787792103499]
 [241.10087333645683, 4.370316992692285, 754.9256451952764]
 [241.10586361375567, 4.517953156259023, 754.7730187544108]
```

```
[241.06025486079685, 4.399702476740089, 754.9368781868886]
[241.09504471067217, 4.316392938786903, 754.9853978749665]
[241.13547651048168, 4.359867906779514, 754.9014911071644]
[241.15081631338433, 4.370166741338697, 754.8758524697025]
[241.0761466822617, 4.313364127885237, 755.0073247142786]
[240.93476885531518, 4.4336179072290385, 755.0284487618814]
```

`plot(sol_sir_sde)`
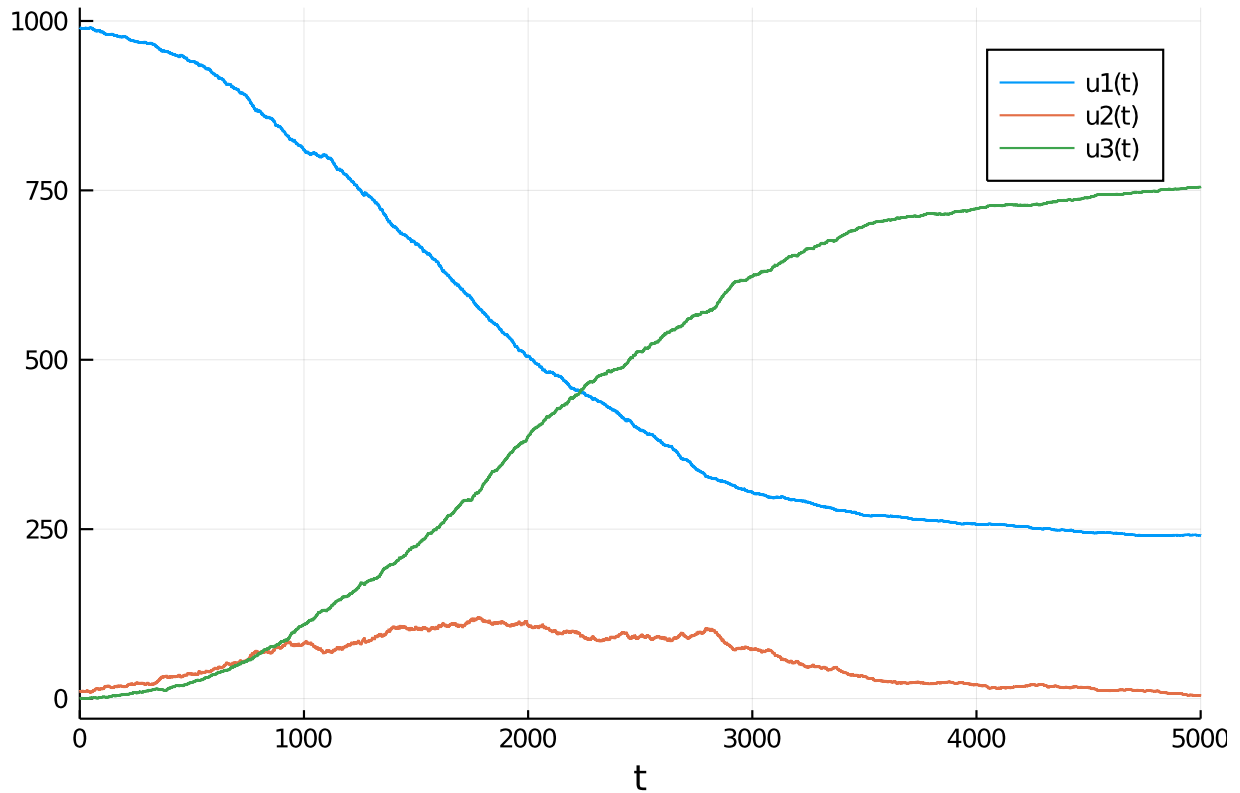
`@benchmark solve(prob_sir_sde,solver=FunctionMap)`

```
BenchmarkTools.Trial:
  memory estimate:  670.05 KiB
  allocs estimate:  5079
  --------------
  minimum time:     976.900 μs (0.00% GC)
  median time:      4.284 ms (0.00% GC)
  mean time:        4.359 ms (2.28% GC)
  maximum time:     49.525 ms (0.00% GC)
  --------------
  samples:          1139
  evals/sample:     1
```