Adaptive Efficiency Tests

Chris Rackauckas

July 5, 2020

```
using Distributed
addprocs(2)

p1 = Vector{Any}(undef,3)
p2 = Vector{Any}(undef,3)
p3 = Vector{Any}(undef,3)

@everywhere begin
  using DiffEqMonteCarlo, StochasticDiffEq, DiffEqProblemLibrary, DiffEqNoiseProcess,
Plots, ParallelDataTransfer
  using DiffEqProblemLibrary.SDEProblemLibrary: importsdeproblems; importsdeproblems()
  import DiffEqProblemLibrary.SDEProblemLibrary: prob_sde_additive,
              prob_sde_linear, prob_sde_wave
end
```

```
Error: On worker 2:
ArgumentError: Package Plots [91a5bcdd-55d7-5caf-9e0b-520d859cae80] is requ
ired but does not seem to be installed:
 - Run `Pkg.instantiate()` to install all recorded dependencies.

_require at ./loading.jl:998
require at ./loading.jl:927
#1 at /buildworker/worker/package_linux64/build/usr/share/julia/stdlib/v1.4
/Distributed/src/Distributed.jl:78
#101 at /buildworker/worker/package_linux64/build/usr/share/julia/stdlib/v1
.4/Distributed/src/process_messages.jl:290
run_work_thunk at /buildworker/worker/package_linux64/build/usr/share/julia
/stdlib/v1.4/Distributed/src/process_messages.jl:79
run_work_thunk at /buildworker/worker/package_linux64/build/usr/share/julia
/stdlib/v1.4/Distributed/src/process_messages.jl:88
#94 at ./task.jl:358

...and 3 more exception(s).
```

```
using DiffEqMonteCarlo, StochasticDiffEq, DiffEqProblemLibrary, DiffEqNoiseProcess,
Plots, ParallelDataTransfer
```

```
Error: On worker 2:
ArgumentError: Package ParallelDataTransfer [2dcacdae-9679-587a-88bb-8b444f
b7085b] is required but does not seem to be installed:
 - Run `Pkg.instantiate()` to install all recorded dependencies.

_require at ./loading.jl:998
require at ./loading.jl:927
#1 at /buildworker/worker/package_linux64/build/usr/share/julia/stdlib/v1.4
```

```
/Distributed/src/Distributed.jl:78
#101 at /buildworker/worker/package_linux64/build/usr/share/julia/stdlib/v1
.4/Distributed/src/process_messages.jl:290
run_work_thunk at /buildworker/worker/package_linux64/build/usr/share/julia
/stdlib/v1.4/Distributed/src/process_messages.jl:79
run_work_thunk at /buildworker/worker/package_linux64/build/usr/share/julia
/stdlib/v1.4/Distributed/src/process_messages.jl:88
#94 at ./task.jl:358

...and 3 more exception(s).

using DiffEqProblemLibrary.SDEProblemLibrary: importsdeproblems; importsdeproblems()

Error: LoadError: On worker 2:
ArgumentError: Package DiffEqBiological [eb300fae-53e8-50a0-950c-e21f52c2b7
e0] is required but does not seem to be installed:
 - Run `Pkg.instantiate()` to install all recorded dependencies.

_require at ./loading.jl:998
require at ./loading.jl:927
#1 at /buildworker/worker/package_linux64/build/usr/share/julia/stdlib/v1.4
/Distributed/src/Distributed.jl:78
#101 at /buildworker/worker/package_linux64/build/usr/share/julia/stdlib/v1
.4/Distributed/src/process_messages.jl:290
run_work_thunk at /buildworker/worker/package_linux64/build/usr/share/julia
/stdlib/v1.4/Distributed/src/process_messages.jl:79
run_work_thunk at /buildworker/worker/package_linux64/build/usr/share/julia
/stdlib/v1.4/Distributed/src/process_messages.jl:88
#94 at ./task.jl:358

...and 3 more exception(s).

in expression starting at /builds/JuliaGPU/DiffEqBenchmarks.jl/.julia/packa
ges/DiffEqProblemLibrary/bJJs6/src/sde_premade_problems.jl:1

import DiffEqProblemLibrary.SDEProblemLibrary: prob_sde_additive,
            prob_sde_linear, prob_sde_wave

probs = Matrix{SDEProblem}(undef,3,3)
## Problem 1
prob = prob_sde_linear

Error: UndefVarError: prob_sde_linear not defined

probs[1,1] =
SDEProblem(prob.f,prob.g,prob.u0,prob.tspan,prob.p,noise=WienerProcess(0.0,0.0,0.0,rswm=RSWM(adaptivea

Error: UndefVarError: prob not defined

probs[1,2] =
SDEProblem(prob.f,prob.g,prob.u0,prob.tspan,prob.p,noise=WienerProcess(0.0,0.0,0.0,rswm=RSWM(adaptivea

Error: UndefVarError: prob not defined

probs[1,3] =
SDEProblem(prob.f,prob.g,prob.u0,prob.tspan,prob.p,noise=WienerProcess(0.0,0.0,0.0,rswm=RSWM(adaptivea

Error: UndefVarError: prob not defined
```

```
## Problem 2
prob = prob_sde_wave
```

```
Error: UndefVarError: prob_sde_wave not defined
```

```
probs[2,1] =
SDEProblem(prob.f,prob.g,prob.u0,prob.tspan,prob.p,noise=WienerProcess(0.0,0.0,0.0,rswm=RSWM(adaptivea
```

```
Error: UndefVarError: prob not defined
```

```
probs[2,2] =
SDEProblem(prob.f,prob.g,prob.u0,prob.tspan,prob.p,noise=WienerProcess(0.0,0.0,0.0,rswm=RSWM(adaptivea
```

```
Error: UndefVarError: prob not defined
```

```
probs[2,3] =
SDEProblem(prob.f,prob.g,prob.u0,prob.tspan,prob.p,noise=WienerProcess(0.0,0.0,0.0,rswm=RSWM(adaptivea
```

```
Error: UndefVarError: prob not defined
```

```
## Problem 3
prob = prob_sde_additive
```

```
Error: UndefVarError: prob_sde_additive not defined
```

```
probs[3,1] =
SDEProblem(prob.f,prob.g,prob.u0,prob.tspan,prob.p,noise=WienerProcess(0.0,0.0,0.0,rswm=RSWM(adaptivea
```

```
Error: UndefVarError: prob not defined
```

```
probs[3,2] =
SDEProblem(prob.f,prob.g,prob.u0,prob.tspan,prob.p,noise=WienerProcess(0.0,0.0,0.0,rswm=RSWM(adaptivea
```

```
Error: UndefVarError: prob not defined
```

```
probs[3,3] =
SDEProblem(prob.f,prob.g,prob.u0,prob.tspan,prob.p,noise=WienerProcess(0.0,0.0,0.0,rswm=RSWM(adaptivea
```

```
Error: UndefVarError: prob not defined
```

```
fullMeans = Vector{Array}(undef,3)
fullMedians = Vector{Array}(undef,3)
fullElapsed = Vector{Array}(undef,3)
fullTols = Vector{Array}(undef,3)
offset = 0
```

```
Ns = [17,23,
17]
```

```
3-element Array{Int64,1}:
 17
 23
 17
```

Timings are only valid if no workers die. Workers die if you run out of memory.

```
for k in 1:size(probs,1)
  global probs, Ns, fullMeans, fullMedians, fullElapsed, fullTols
  println("Problem $k")
  ## Setup
  N = Ns[k]

  msims = Vector{Any}(undef,N)
  elapsed = Array{Float64}(undef,N,3)
  medians = Array{Float64}(undef,N,3)
  means   = Array{Float64}(undef,N,3)
  tols    = Array{Float64}(undef,N,3)

  #Compile
  prob = probs[k,1]
  ParallelDataTransfer.sendto(workers(), prob=prob)
  monte_prob = MonteCarloProblem(prob)

solve(monte_prob,SRIW1(),dt=1/2^(4),adaptive=true,num_monte=1000,abstol=2.0^(-1),reltol=0)

  println("RSwM1")
  for i=1+offset:N+offset
    tols[i-offset,1] = 2.0^(-i-1)
    msims[i-offset] = DiffEqBase.calculate_monte_errors(solve(monte_prob,SRIW1(),
                                          num_monte=1000,abstol=2.0^(-i-1),
                                          reltol=0,force_dtmin=true))
    elapsed[i-offset,1] = msims[i-offset].elapsedTime
    medians[i-offset,1] = msims[i-offset].error_medians[:final]
    means[i-offset,1]   = msims[i-offset].error_means[:final]
  end

  println("RSwM2")
  prob = probs[k,2]

  ParallelDataTransfer.sendto(workers(), prob=prob)
  monte_prob = MonteCarloProblem(prob)

solve(monte_prob,SRIW1(),dt=1/2^(4),adaptive=true,num_monte=1000,abstol=2.0^(-1),reltol=0)

  for i=1+offset:N+offset
    tols[i-offset,2] = 2.0^(-i-1)
    msims[i-offset] = DiffEqBase.calculate_monte_errors(solve(monte_prob,SRIW1(),
                                          num_monte=1000,abstol=2.0^(-i-1),
                                          reltol=0,force_dtmin=true))
    elapsed[i-offset,2] = msims[i-offset].elapsedTime
    medians[i-offset,2] = msims[i-offset].error_medians[:final]
    means[i-offset,2]   = msims[i-offset].error_means[:final]
  end

  println("RSwM3")
  prob = probs[k,3]
  ParallelDataTransfer.sendto(workers(), prob=prob)
  monte_prob = MonteCarloProblem(prob)

solve(monte_prob,SRIW1(),dt=1/2^(4),adaptive=true,num_monte=1000,abstol=2.0^(-1),reltol=0)

  for i=1+offset:N+offset
    tols[i-offset,3] = 2.0^(-i-1)
        msims[i-offset] = DiffEqBase.calculate_monte_errors(solve(monte_prob,SRIW1(),
                                  adaptive=true,num_monte=1000,abstol=2.0^(-i-1),
```

```
                                    reltol=0,force_dtmin=true))
    elapsed[i-offset,3] = msims[i-offset].elapsedTime
    medians[i-offset,3] = msims[i-offset].error_medians[:final]
    means[i-offset,3]   = msims[i-offset].error_means[:final]
  end

  fullMeans[k] = means
  fullMedians[k] =medians
  fullElapsed[k] = elapsed
  fullTols[k] = tols
end
```

```
Problem 1
Error: UndefRefError: access to undefined reference
```

```
gr(fmt=:svg)
lw=3
leg=String["RSwM1","RSwM2","RSwM3"]

titleFontSize = 16
guideFontSize = 14
legendFontSize= 14
tickFontSize  = 12

for k in 1:size(probs,1)
  global probs, Ns, fullMeans, fullMedians, fullElapsed, fullTols
  p1[k] = Plots.plot(fullTols[k],fullMeans[k],xscale=:log10,yscale=:log10,
xguide="Absolute Tolerance",yguide="Mean Final Error",title="Example $k"
,linewidth=lw,grid=false,lab=leg,titlefont=font(titleFontSize),legendfont=font(legendFontSize),tickfon
  p2[k] =
Plots.plot(fullTols[k],fullMedians[k],xscale=:log10,yscale=:log10,xguide="Absolute
Tolerance",yguide="Median Final Error",title="Example
$k",linewidth=lw,grid=false,lab=leg,titlefont=font(titleFontSize),legendfont=font(legendFontSize),tick
  p3[k] =
Plots.plot(fullTols[k],fullElapsed[k],xscale=:log10,yscale=:log10,xguide="Absolute
Tolerance",yguide="Elapsed Time",title="Example $k"
,linewidth=lw,grid=false,lab=leg,titlefont=font(titleFontSize),legendfont=font(legendFontSize),tickfon
end
```

```
Error: UndefRefError: access to undefined reference
```

```
Plots.plot!(p1[1])
```

```
Error: UndefRefError: access to undefined reference
```

```
Plots.plot(p1[1],p1[2],p1[3],layout=(3,1),size=(1000,800))
```

```
Error: UndefRefError: access to undefined reference
```

```
#savefig("meanvstol.png")
#savefig("meanvstol.pdf")
```

```
plot(p3[1],p3[2],p3[3],layout=(3,1),size=(1000,800))
```

```
Error: UndefRefError: access to undefined reference
```

```
#savefig("timevstol.png")
#savefig("timevstol.pdf")
```

```
plot(p1[1],p3[1],p1[2],p3[2],p1[3],p3[3],layout=(3,2),size=(1000,800))
```

```
Error: UndefRefError: access to undefined reference
```

```
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("AdaptiveSDE","AdaptiveEfficiencyTests.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/AdaptiveSDE/Project.toml`
[78ddff82-25fc-5f2b-89aa-309469cbf16f] DiffEqMonteCarlo 0.15.1
[77a26b50-5914-5dd7-bc55-306e6241c503] DiffEqNoiseProcess 5.0.2
[a077e3f3-b75c-5d7f-a0c6-6bc4c8ec64a9] DiffEqProblemLibrary 4.8.0
[2dcacdae-9679-587a-88bb-8b444fb7085b] ParallelDataTransfer 0.5.0
[91a5bcdd-55d7-5caf-9e0b-520d859cae80] Plots 1.5.3
[789caeaf-c7a9-5a7d-9973-96adeb23e2a0] StochasticDiffEq 6.24.0
[8ba89e20-285c-5b6f-9357-94700520ee1b] Distributed
[9a3f8284-a2c9-5f02-9a11-845980a1fd5c] Random
```